Comply
Contents

Welcome to Puppet Comply................................................................. 5
  Comply terminology ...................................................................... 5
  Comply overview ........................................................................... 6
  Supported CIS Benchmarks ............................................................... 6
  CIS-CAT Pro Assessor history .............................................................. 7

Comply release notes ........................................................................ 9
  Comply known issues ..................................................................... 17

Beginner’s guide to Comply ............................................................... 19

Puppet Application Manager .......................................................... 20
  Welcome to Puppet Application Manager (PAM) .................................. 21
  Release notes .................................................................................... 23
    PAM release notes ........................................................................ 23
    Known issues ................................................................................ 28
  Architecture overview ...................................................................... 29
  PAM system requirements ................................................................ 32
  Component versions in PAM releases .................................................. 45
  Install PAM ...................................................................................... 46
    Install Puppet applications using PAM on a customer-supported Kubernetes cluster ........................................ 47
    PAM HA online installation ............................................................ 49
    PAM HA offline installation ............................................................ 53
    PAM standalone online installation ................................................ 56
    PAM standalone offline installation ................................................ 59
    Automate PAM and Puppet application online installations .................. 61
    Automate PAM and Puppet application offline installations ................ 63
    Uninstall PAM ................................................................................ 67
  Working with Puppet applications ..................................................... 67
    Install applications via the PAM UI ................................................ 68
    Update a license for online installations ......................................... 69
    Update a license for offline installations ......................................... 69
    Upgrade an automated online application installation ....................... 69
    Upgrade an automated offline application installation ....................... 70
  Maintenance and tuning .................................................................. 71
  Upgrading PAM on a Puppet-supported cluster .................................... 72
  Upgrading PAM on a customer-supported cluster .................................. 76
  Backing up PAM using snapshots ...................................................... 77
  Migrating PAM data to a new system .................................................. 80
  Disaster recovery with PAM .............................................................. 86
  Troubleshooting PAM ...................................................................... 87

Installing .......................................................................................... 92
  System requirements ...................................................................... 92
  Set up Comply ................................................................................ 93
Configure Comply (online environment) ................................................................. 94
Configure Comply in an offline environment .......................................................... 95
Configure Comply TLS certificates ......................................................................... 96
Configure Comply for a custom NGINX ingress (online environment) ................... 97
Configure Comply for a custom NGINX ingress (offline environment) ................... 98
Configure Comply TLS certificates for a custom NGINX ingress ............................. 99
Install the Comply module ..................................................................................... 101
Classify nodes ....................................................................................................... 102
Deploy Comply ..................................................................................................... 103
Add PE credentials ............................................................................................... 103
Uninstall Comply and remove PAM ................................................................. 104
Uninstall Comply without removing PAM .......................................................... 104
Remove the CIS-CAT Pro Assessor from a node .................................................. 105

Upgrading .............................................................................................................. 105

Desired compliance .............................................................................................. 108

Custom profiles ..................................................................................................... 109

Exceptions ............................................................................................................ 111

CIS scans .............................................................................................................. 112
  Guidelines for running scans at scale ................................................................. 113
  Run an ad hoc scan ........................................................................................... 114
  Scheduled scans ............................................................................................... 115
    View details about a scheduled scan ............................................................. 115
    Pause and resume a scheduled scan ............................................................ 116
    Edit a scheduled scan ................................................................................... 116
    Delete a scheduled scan .............................................................................. 116
    Create a one-off scheduled scan ................................................................. 116
    Create a repeating scheduled scan .............................................................. 117
  CIS scan reports ............................................................................................... 117
    CIS scan report details .................................................................................. 118

Scan results .......................................................................................................... 118

Enforce CIS benchmarks ...................................................................................... 121

Troubleshooting ................................................................................................. 121

Introducing the Compliance Enforcement Modules .............................................. 125

CEM for Linux ..................................................................................................... 125
  Release notes ................................................................................................. 126
Known issues and limitations...............................................................................................................130
Installing CEM.................................................................................................................................................. 132
  Prepare to install the module............................................................................................................... 132
  Install the module and classify nodes.................................................................................................. 133
  Uninstall the module.............................................................................................................................133
Upgrading CEM................................................................................................................................................ 134
Configuring CEM............................................................................................................................................. 134
  Overview of configuration options.......................................................................................................134
  How to configure the module: Examples and guidelines.................................................................... 138
Auditing and querying issues identified during scans.....................................................................................143
Reference: Benchmarks and controls...............................................................................................................143

CEM for Windows..............................................................................................................................................143
  Release notes.....................................................................................................................................................143
  Known issues and limitations...............................................................................................................148
  Installing CEM................................................................................................................................................ 149
  Prepare to install the module............................................................................................................... 149
  Install the module and classify nodes.................................................................................................. 150
  Uninstall the module.............................................................................................................................150
Upgrading CEM................................................................................................................................................ 151
Configuring CEM............................................................................................................................................. 151
  Overview of configuration options.......................................................................................................151
  How to configure the module: Examples and guidelines.................................................................... 153
Reference: Benchmarks and controls...............................................................................................................157
Welcome to Puppet Comply

Puppet Comply is a tool that assesses the infrastructure you manage with Puppet Enterprise against CIS Benchmarks — the best practices from the Center for Internet Security (CIS) for securely configuring systems.

Using Comply, you can:

- Run scans to check the compliance of your infrastructure against CIS Benchmarks on up to 5000 nodes.
- Set your desired compliance — a default benchmark and profile that you want your scans to be measured against.
- Customize profiles to specify which rules you want visible in scan reports.
- Identify the cause and source of compliance failures, and determine what configuration changes must be made to which systems.

Comply uses Puppet Enterprise (PE) to retrieve node and fact information. After you install Comply, you must configure it to integrate with PE.

If this is your first time using Comply, try out our Beginner’s guide to Comply.

### Puppet Comply docs links

<table>
<thead>
<tr>
<th>Learn the basics:</th>
<th>Other useful places</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comply overview</td>
<td>Comply videos:</td>
</tr>
<tr>
<td>Comply terminology</td>
<td>Comply introduction and demo</td>
</tr>
<tr>
<td>Beginner’s guide to Comply</td>
<td>Related Puppet products:</td>
</tr>
<tr>
<td>Release notes</td>
<td>Puppet Enterprise</td>
</tr>
<tr>
<td>Install and configure Comply:</td>
<td>Puppet Forge</td>
</tr>
<tr>
<td>System requirements</td>
<td>Get help:</td>
</tr>
<tr>
<td>Install Puppet Application Manager</td>
<td>Troubleshooting</td>
</tr>
<tr>
<td>Set up Comply</td>
<td>Support portal</td>
</tr>
</tbody>
</table>

### Comply terminology

Learn the key terms that are associated with Puppet Comply.

**CIS Benchmarks**

Developed by the Center for Internet Security (CIS), *CIS Benchmarks* are internationally recognized standards and best practices for securely configuring systems. For more information, see [CIS Benchmarks](#).

**CIS assessor**

Comply integrates with the *CIS assessor* (CIS-CAT PRO), the scanner tool that assesses CIS benchmarks. As part of the Comply configuration process, Puppet Enterprise (PE) installs the CIS assessor on your target nodes. For more information on the assessor, see [CIS-CAT Pro](#).
Profiles

CIS Benchmarks include different levels of security settings, called profiles. The Level 1 profiles are the base recommendation for every system, and the Level 2 profiles are intended for environments requiring greater security. Comply can scan for either profile.

Rules

Each profile contains multiple rules that define specific elements of system configuration.

Custom profiles

A custom profile is a benchmark profile that you customize to fit your organization's internally defined standards, by specifying which rules you want visible in scan reports. Once you create a custom profile, it appears as an option in Comply when selecting a benchmark and profile.

Desired compliance

Desired compliance is the benchmark and profile that you assign to a node. It becomes the default scan for that node.

For a full list of Puppet terminology, see the Puppet Glossary.

Comply overview

Welcome to Puppet Comply!

This overview is intended for new users of Comply. We go over what Comply is, how it works, and show a demo of the 1.0.0 release. Before you begin, we recommend familiarizing yourself with our terminology.

What is Comply and how does it work?

Comply is a tool that expands the compliance capabilities of Puppet Enterprise (PE), by integrating with the CIS assessor to scan your infrastructure against the latest CIS Benchmarks. Comply connects to your PE environment and gathers information about your PE managed nodes, including operating system facts and classification node groups. It uses this information to suggest appropriate scans.

You can choose to run ad-hoc scans or desired compliance scans — a default CIS benchmark and profile scan that you assign to a node. Comply can automate desired compliance for you based on the information it gathers about your nodes from PE, or you can manually choose your desired compliance from a list of benchmarks and profiles. You can also create custom profiles to fit internally defined standards, by specifying which rules you want visible in scan reports. Most of the time, you only need to set your desired compliance once.

The scans are run as a task in PE. Scan results populate in the Comply Compliance dashboard, where you can see the number of nodes scanned and their compliance breakdown. In each node listed, there is a further breakdown of rule information which tells you why that rule is important, and steps you can take to fix the rule if it is failing the scans.

To see Comply in action, watch the demo below, or go through the steps yourself in our beginner's guide.

For a full list of features, see the release notes.

Comply demo

The following demo walks you through the key features of the Comply 1.0.0 release:

Supported CIS Benchmarks

Comply supports the following CIS operating system benchmarks.
<table>
<thead>
<tr>
<th>Operating system</th>
<th>Supported versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alma Linux</td>
<td>8</td>
</tr>
<tr>
<td>Amazon Linux</td>
<td>2, 2 STIG</td>
</tr>
<tr>
<td>Azure Compute Microsoft Windows Server 2019</td>
<td>1.0.0</td>
</tr>
<tr>
<td>CentOS</td>
<td>6, 7</td>
</tr>
<tr>
<td>Debian Linux</td>
<td>9, 10, 11 (v1.0.0)</td>
</tr>
<tr>
<td>Mac OS X</td>
<td>10.15, 11.0, 12</td>
</tr>
<tr>
<td>Oracle Linux</td>
<td>6, 7, 8</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux (RHEL)</td>
<td>6, 7, 7 STIG, 8, 8 STIG</td>
</tr>
<tr>
<td>Rocky Linux</td>
<td>8</td>
</tr>
<tr>
<td>SUSE Linux Enterprise Server (SLES)</td>
<td>12, 15</td>
</tr>
<tr>
<td>Ubuntu</td>
<td>16.04*, 18.04, 20.04, 20.04 STIG, 22.04</td>
</tr>
</tbody>
</table>

* **Ubuntu users:**
  - If you use Ubuntu 16.04, you need Curl with Mutual Transport Layer Security (MTLS) support to upgrade Comply and run scans.
  - The feature for managing Java runtime packages, manage-java, is no longer supported for Ubuntu 16.04 because the Puppet java module is not available in this environment.

** **Microsoft Windows users:** Starting with Comply 2.8.0, you can apply the CIS Microsoft Windows 10 Stand-alone Benchmark to stand-alone systems, which are not connected to a domain.

**CIS-CAT Pro Assessor history**

Comply supports the latest and previous version of the CIS-CAT Pro Assessor.

**CIS-CAT Pro Assessor v4.23.0**

**Checksum:** e8c1ae0519c89e420c27622d4497c12e772d82c3ea2c6952abad9185acafe532f

**Benchmarks:**

- CIS Ubuntu Linux 22.04 LTS Benchmark v1.0.0
- CIS Microsoft Windows Server 2019 Benchmark v1.3.0
- CIS Red Hat Enterprise Linux 8 STIG Benchmark v1.0.0
- CIS Amazon Linux 2 Benchmark v2.0.0
- CIS Microsoft Windows Server 2012 R2 Benchmark v2.6.0
- CIS Amazon Linux Benchmark v2.0.0
- CIS Apple macOS 12.0 Monterey Benchmark v1.1.0
- CIS Ubuntu Linux 16.04 LTS Benchmark v2.0.0
- CIS Debian Linux 10 Benchmark v1.0.0
- CIS SUSE Linux Enterprise 12 Benchmark v3.1.0
- CIS Microsoft Windows Server 2016 Benchmark v1.4.0
- CIS Red Hat Enterprise Linux 6 Benchmark v3.0.0
- CIS Oracle Linux 8 Benchmark v2.0.0

© 2022 Puppet, Inc., a Perforce company
• CIS Debian Linux 11 Benchmark v1.0.0
• CIS Amazon Linux 2 STIG Benchmark v2.0.0
• CIS Apple macOS 10.15 Catalina Benchmark v2.1.0
• CIS Red Hat Enterprise Linux 8 Benchmark v2.0.0
• CIS Microsoft Windows Server 2012 (non-R2) Benchmark v2.4.0
• CIS Microsoft Windows Server 2016 STIG Benchmark v1.2.0
• CIS Microsoft Windows 11 Enterprise Benchmark v1.0.0
• CIS Ubuntu Linux 20.04 LTS Benchmark v1.1.0
• CIS Microsoft Windows Server 2019 Benchmark v1.1.0
• CIS CentOS Linux 7 Benchmark v3.1.2
• CIS Oracle Linux 6 Benchmark v2.0.0
• CIS Microsoft Windows Server 2022 Benchmark v1.0.0
• CIS Red Hat Enterprise Linux 7 STIG Benchmark v2.0.0
• CIS Azure Compute Microsoft Windows Server 2019 Benchmark v1.0.0
• CIS Rocky Linux 8 Benchmark v1.0.0
• CIS Red Hat Enterprise Linux 7 Benchmark v3.1.1
• CIS Microsoft Windows 10 Stand-alone Benchmark v1.0.1
• CIS Ubuntu Linux 20.04 LTS STIG Benchmark v1.0.0
• CIS Microsoft Windows 10 Enterprise Benchmark v1.12.0
• CIS CentOS Linux 6 Benchmark v3.0.0
• CIS SUSE Linux Enterprise 15 Benchmark v1.1.1
• CIS Ubuntu Linux 18.04 LTS Benchmark v2.1.0
• CIS Oracle Linux 7 Benchmark v3.1.1
• CIS Apple macOS 11.0 Big Sur Benchmark v2.1.0
• CIS Debian Linux 9 Benchmark v1.0.1
• CIS Alma Linux OS 8 Benchmark v2.0.0

CIS-CAT Pro Assessor v4.22.0

Checksum: 9331f00f1b481fdd161512f258dcc0002ca5893d5a7c807dfc0379751071408

Benchmarks:
• CIS Ubuntu Linux 22.04 LTS Benchmark v1.0.0
• CIS Microsoft Windows Server 2019 Benchmark v1.3.0
• CIS Red Hat Enterprise Linux 8 STIG Benchmark v1.0.0
• CIS Amazon Linux 2 Benchmark v2.0.0
• CIS Microsoft Windows Server 2012 R2 Benchmark v2.6.0
• CIS Amazon Linux Benchmark v2.0.0
• CIS Apple macOS 12.0 Monterey Benchmark v1.1.0
• CIS Ubuntu Linux 16.04 LTS Benchmark v2.0.0
• CIS Debian Linux 10 Benchmark v1.0.0
• CIS SUSE Linux Enterprise 12 Benchmark v3.1.0
• CIS Microsoft Windows Server 2016 Benchmark v1.4.0
• CIS Red Hat Enterprise Linux 6 Benchmark v3.0.0
• CIS Oracle Linux 8 Benchmark v2.0.0
• CIS Debian Linux 11 Benchmark v1.0.0
• CIS Amazon Linux 2 STIG Benchmark v2.0.0
• CIS Apple macOS 10.15 Catalina Benchmark v2.1.0
• CIS Red Hat Enterprise Linux 8 Benchmark v2.0.0
• CIS Microsoft Windows Server 2012 (non-R2) Benchmark v2.4.0
• CIS Microsoft Windows Server 2016 STIG Benchmark v1.2.0

© 2022 Puppet, Inc., a Perforce company
Comply release notes

Learn about the new features, enhancements, and resolved issues for the Puppet Comply 2.x release series.

Comply 2.10.0
Released 1 December 2022

New in this release:

- **CIS-CAT Pro Assessor v4.23.0.** Comply 2.10.0 includes the CIS-CAT Pro Assessor v4.23.0.
- **Security notice:**
  - The CIS-CAT Pro Assessor v4.23.0 resolves a security vulnerability present in the embedded, third party dependency for the jackson-databind mapping functionality. This library has moved to jackson-databind-2.13.4.jar.
- **Export scan results.** You can now export the last scan results for all nodes, a subset of nodes, or a single node. All exported data is collected in a single .csv file. To export scan results, use the Export CSV button on the Node Results pane on the Compliance Dashboard. To view and download previous reports, use the Generated Reports button in the Comply navigation pane.
- **Resolve exceptions.** You can now resolve exceptions that are no longer needed. Details about resolved exceptions remain visible in Puppet Comply for reporting purposes. You can resolve an exception for all nodes or for a subset of nodes.
- **Exception details.** You can now view and edit the details of your exceptions.
- **Using old versions of the CIS-CAT Pro Assessor.** You can now upgrade to the latest version of Comply without updating the CIS-CAT Pro Assessor. As of this release the supported versions of the CIS-CAT Pro Assessor are 4.22.0 and 4.23.0. In future releases, the current version and the two previous versions will be supported. All nodes still must run the same version of the CIS-CAT Pro Assessor.
- **Security notice:**
  - Only the latest version of the CIS-CAT Pro Assessor has the latest security fixes. Customers on older versions of the CIS-CAT Pro Assessor may be vulnerable to security issues.
• **Node group filtering.** Anywhere all nodes are listed, node groups filtering now supports nodes that have been pinned to the node group in PE. Node groups are based on PE classification groups.

Resolved in this release:

• **Exceptions remain active after they are no longer applicable.** Exceptions are now removed if their custom profile is deleted or edited to remove the relevant rule.

• **A deleted exception cannot immediately be re-created.** Previously, if you created an exception for a specified rule and node and then deleted the exception, you could not immediately re-create the exception for the specified rule and node. This has been fixed.

**Comply 2.9.0**

Released 20 October 2022

New in this release:

• **CIS-CAT Pro Assessor v4.22.0.** Comply
  2.9.0 includes the CIS-CAT Pro Assessor v4.22.0 and the following associated benchmarks:
  • Debian Linux 11, v1.0.0
  • Azure Compute Microsoft Windows Server 2019, v1.0.0
  • Security notice:
    • The CIS-CAT Pro Assessor v4.22.0 resolves security vulnerabilities present in embedded, third party dependencies. For details, see [CIS-CAT Pro Dashboard and Assessor September 2022 Vulnerability Updates](#).

• **Create temporary exceptions to rules.** With Comply 2.9.0, you can create a temporary exception to a CIS Benchmark rule and apply that exception to a node, a group of nodes, or all nodes. During the period when the exception is active, the rule's compliance score is excluded from the overall compliance score for the selected nodes. Exceptions are useful in many situations. For example, if you plan to install a software patch on several nodes, but the patch requires additional testing, you can specify a temporary exception for the affected nodes while testing continues. During the next scan, the exception is applied, and the compliance score reflects the exception. When testing is completed, you can apply the software patch to the nodes, and the exception expires automatically on your specified date.

• **View and delete exceptions.** You can go to the new Exceptions page to view and delete exceptions.

Resolved in this release:

• **Scans fail to complete processing.** In some cases, when scans were run manually, the scans would remain in the started state and would fail to generate a final report.

**Comply 2.8.0**

Released 8 September 2022

New in this release:

• **CIS-CAT Pro Assessor v4.21.0.** Comply
  2.8.0 includes the CIS-CAT Pro Assessor v4.21.0 and the following associated benchmarks:
  • Microsoft Windows 11.
  • Microsoft Windows 10 (stand-alone). (A stand-alone system is not connected to a domain and cannot be managed by using Active Directory.)
  • Ubuntu 22.04.

• **Specify a refresh interval to obtain the latest inventory updates from Puppet Enterprise (PE).** By default, the Comply inventory is refreshed every 24 hours with the latest node and fact information from Puppet Enterprise. With Comply 2.8.0, you can customize the refresh interval to meet your organization’s requirements.

Resolved in this release:

• **Consistency of scan compliance scores.** To help ensure consistency of compliance scores throughout the Comply user interface, the Node detail page and the Rule detail page are updated. The donut charts and the
accompanying legends now exclude non-scoring statuses. A non-scoring status means that a CIS recommendation is not applicable or cannot be automatically validated. With this change, the charts on the Node detail page and Rule detail page now provide a more realistic view of compliance.

- **Accurate status for profiles.** The Profile column on the Scan Report page now reflects the correct status of profiles. Previously, if you hovered over the Profile column, you might have seen an invalid message that the profile was deleted.
- **Scheduled scans not running after Comply upgrade.** After upgrading Comply, scheduled scans that were created before the upgrade might not run. After upgrading to Comply v2.8.0, these scans should run as configured.

Security notice:

- This release includes a security update that helps to prevent command injection in the Comply module.

**Comply 2.7.0**

Released 27 July 2022

New in this release:

- **CIS-CAT Pro Assessor v4.19.0.** Comply 2.7.0 includes the CIS-CAT Pro Assessor v4.19.0.
- **Learn how to run Comply at scale.** You can scan up to 5000 nodes in a single batch to check the compliance of your infrastructure against Center for Internet Security (CIS) Benchmarks. The documentation is updated to help you configure and run scans at scale. See Guidelines for running scans at scale on page 113.
- **Delete a custom profile.** In previous releases, you could create a custom profile based on a CIS Benchmark. In this release, you can also delete one or more custom profiles.

Resolved in this release:

- **Warning messages during preflight checks.** An issue that caused invalid warning messages to be displayed during preflight checks is resolved in this release. The invalid message, No matching files, is no longer displayed.

**Comply 2.6.0**

Released 16 June 2022

New in this release:

- **CIS-CAT Pro Assessor v4.18.0.** Comply 2.6.0 includes the CIS-CAT Pro Assessor v4.18.0 and the following associated benchmarks:
  - Alma Linux 8 v2.0.0
  - Microsoft Windows Server 2016 v1.4.0
  - Microsoft Windows Server 2016 STIG v1.2.0
  - Microsoft Windows Server 2012 v2.4.0
  - Microsoft Windows Server 2012 R2 v2.6.0
- **Possible errors due to renamed benchmarks:** In addition to version changes, CIS renamed two benchmarks in this release. AlmaLinux was renamed to Alma Linux and Microsoft Windows Server 2016 RTM (Release_1607) was renamed to Microsoft Windows Server 2016. If you are using a benchmark that was renamed, you might see an error message indicating that the benchmark is no longer supported. If your nodes use custom profiles that are based on renamed benchmarks, you must manually update the nodes because they will not be automatically updated during the Comply upgrade process.
- **Edit a scheduled scan.** You can edit a scheduled scan to modify the type of scan, the frequency, and the start and end dates.
- **Delete a scheduled scan.** You can delete a scheduled scan to permanently remove it.
- **Take advantage of enhanced usability for scan reports.** From a scan report, you can navigate to the Node detail page, where the Scan status pane now includes a legend showing the total number of rules that were run on the node and detailed results. You can hover over the results to see percentages in the donut chart.
on the **Rule detail** page, the **Scan status** pane now shows the total number of scanned nodes and detailed results. You can hover over the results to see percentages in the donut chart. The **Rule detail** page includes a new **Environment** column so that you can determine the environment (for example, test or production) in which the scan took place. The **Node detail** page includes a new **Last passed on** column, which shows the date and time of the last successful scan for each rule.

Security notice:
- Vulnerability in the 3.14.2-alpine image. The release updates the alpine image to 3.15.4.

**Comply 2.5.1**

Released 31 May 2022

Resolved in this release:

- **Potential deployment issue for users of Comply 2.4.0 and 2.5.0.** This issue can affect users who install Comply in a Google Kubernetes Engine (GKE) environment and potentially other environments. If you are unable to start Comply after installation, you might be experiencing this issue. To diagnose the issue, review the log for the **comply-scarpy** pod. If the issue is occurring, the pod will be in an **Init:CrashLoopBackOff** state during the attempt to start Comply. Review of the pod will show that the **comply-scarpy-init** container was terminated with an out-of-memory error (**OOMKilled**). To resolve the issue, install Comply 2.5.1. If you do not detect the issue, it is not necessary to install Comply 2.5.1.

**Comply 2.5.0**

Released 5 May 2022

New in this release:

- **CIS-CAT Pro Assessor v4.16.1.** Comply 2.5.0 includes the CIS-CAT Pro Assessor v4.16.1 and the following associated benchmarks:
  - Microsoft Windows Server 2019 v1.3.0
  - Microsoft Windows Server 2019 STIG v1.1.0
  - Oracle Linux 8
  - Rocky Linux 8
  
  - The following CIS benchmarks are at end of life and are no longer supported:
    - CentOS Linux 8
    - SUSE Linux Enterprise Server 11
  - **View details about a scheduled scan.** You can select a scheduled compliance scan and view its details, including the creation date, last modification date, affected nodes, start and end times, and frequency. You can also view the scan history, including the number of runs, the date and time of the most recent run, and the date and time of the next scheduled run.
  - **Pause, resume, or end a scheduled scan.** On the **Scheduled scan details** page, you can pause, resume, or end a scheduled scan.
  - **Assign benchmarks and profiles to multiple nodes simultaneously.** On the **Inventory** page, you can select multiple nodes and assign a benchmark, a profile, and, optionally, a custom profile to all. The selected nodes must be running on the same operating system, and the latest version of the CIS-CAT Pro Assessor must be installed on each node.
  - **View a report about scan results for a single rule.** The **Scan rule report** lists the nodes on which the rule was run, the results, and the overall compliance score for the rule.
  - **View a report about scan results for a single node.** The **Scan node report** lists the rules that were run on the node, the results, and the overall compliance score for the node.
Resolved in this release:

- **Initial deployment issue on Microsoft Windows Server 2016 and Microsoft Windows Server 2019 operating systems.** In previous releases, the initial deployment of the Comply module sometimes failed with the following error message:

```
Provider wget is not functional on this host
```

**Comply 2.4.0**

Released 24 March

New in this release:

- **CIS-CAT Pro Assessor v4.15.0.** Comply 2.4.0 includes the latest version of the CIS-CAT assessor and the following supported associated benchmarks:
  - CentOS Linux 8 (final release)
  - Microsoft Windows 10 v1.12.0.
  - Microsoft Windows Server 2022 v1.0.0
  - Red Hat Enterprise Linux 8 v2.0.0
  - SUSE Linux Enterprise 11 v2.1.1 (final release)

**Note:** The Microsoft Windows 10 benchmark has upgraded from 1.11.0 CIS Microsoft Windows 10 Enterprise Release 21H1 to 1.12.0 CIS Microsoft Windows 10 Enterprise. Comply’s 1.12.0 CIS Microsoft Windows 10 Enterprise benchmark is based on Microsoft Windows 10 Enterprise Release 21H2 and is intended for all versions of the Windows 10 operating system, including older versions. If any of your nodes use custom profiles based on the 1.11.0 CIS Microsoft Windows 10 Enterprise Release 21H1 benchmark, you need to resolve these manually, as they will not automatically update during the upgrade process.

- **Profile and Custom profile.** You can view and sort two new columns on the Inventory page - Profile and Custom profile. The columns allow you to see if a node has a default profile or custom profile assigned to it.

- **Benchmark column.** The Desired compliance column has been renamed to Benchmark.

Resolved in this release:

- **Sync license.** Fixed an issue where a user was logged out of Comply after selecting Sync license on the License page.

**Comply 2.3.0**

Released 10 February 2022

New in this release:

- **Scheduled scans.** You can now schedule one-off and repeating scans, in addition to running manual ad hoc scans, in Comply.
  
  For more information, see Scheduled scans on page 115.

- **Environment information.** The Scan list page now shows the scan report environment.

- **CIS-CAT Pro Assessor v4.14.0.** Comply 2.3.0 includes the latest version of the CIS-CAT assessor and the following supported associated benchmarks:
  - SUSE Linux Enterprise 12 v3.1.0

© 2022 Puppet, Inc., a Perforce company
• SUSE Linux Enterprise 15 v1.1.1

This release of the assessor resolves security vulnerability present in embedded, third party dependencies:

• The OpenDXL Java Client library, which includes log4j, is now a derivative work of version 0.2.6 which includes log4j 2.17.1.
• The logback-core and logback-classic libraries have been moved to version 1.2.10.
• Comply now supports Kubernetes 1.19 to 1.24. Kubernetes 1.17 and 1.18 are no longer supported.

Resolved in this release:

• Rule details. Fixed an bug where the last reported time stamp on the rule detail page did not recognize the user's local timezone.
• Compliance profiles. Corrected an issue where the default compliance profile was incorrectly assigned for Windows Server versions.

Comply 2.2.2

Released 20 January 2022

New in this release:

• Debug mode. You can now choose to run in debug mode to provide easier access to assessor logs.
  For more information, see Run an ad hoc scan on page 114.
• CIS-CAT Pro Assessor v4.13.1. Comply 2.2.2 includes the latest version of the CIS-CAT assessor and the following supported associated benchmarks:
  • AlmaLinux OS 8 v1.0.0
  • Amazon Linux 2 STIG v2.0.0
  • Apple macOS 11.0 Big Sur v2.0.0
  • Microsoft Windows Server 2012 (non-R2) v2.3.0
  • Red Hat Enterprise Linux 8 STIG v1.0.0

CIS-CAT Pro Assessor v4.13.1 resolved security vulnerabilities present in the following embedded, third party dependency:

• log4j-core - This library was updated to version 2.17.0.

Comply 2.2.1

Released 20 December 2021

New in this release:

CIS-CAT Pro Assessor v4.13.0. Comply 2.2.1 includes the latest version of the CIS-CAT assessor and the following supported associated benchmarks:

• Apple macOS 10.15 Catalina v2.0.0
• Red Hat Enterprise Linux 7 STIG v2.0.0

The following benchmark is at end of life and is no longer supported:

• Mac OS 10.14

Security notice:
• **CIS-CAT Pro Assessor v4.13.0** resolved security vulnerabilities present in the following embedded, third party dependencies:
  • **log4j-core** - This library was updated to version 2.15.0.
  • **bcprov-jdk15on** - This library was updated to version 1.69.
• **Component upgrade to address CVEs.** To address various CVEs, this version includes an upgrade of Kubernetes to 1.19.15.

| Important: | Version 2.15.0 of the log4j-core library addresses the potential escalation of privilege vulnerability. We do not believe Comply is vulnerable to any of the additional risks addressed in the 2.16.0 release, but plan to release an update in the near future which includes version 2.17.0 or later. |

**Comply 2.2.0**

Released 18 November 2021.

New in this release:

• **Scan Reports improvements.** Scan reporting functionality is extended to include the ability to access a list of historical scans and view scan details. For more information, see CIS scan report details on page 118.
• **Filtering and sorting.** Filtering and sorting functionality has been implemented on all table columns in the Comply UI.

| Note: | Filter drop-downs display all available options for a given parameter. On pages where multiple filtering options are available, selecting one filter option does not affect the options presented by any other filter drop-down. |

• **CIS-CAT Pro Assessor v4.11.0.** Comply 2.2.0 includes the latest version of the CIS-CAT assessor and its associated benchmarks:
  • Microsoft Windows Server 2012 R2 v2.5.0
  • Microsoft Windows Server 2016 STIG v1.1.0
  • SUSE Linux 15 v1.1.0
• **Desired compliance.** The Comply UI has been simplified so that users are no longer required to manually accept the profiles applied by Comply based on fact information from PE.
• **Custom Comply port.** You can now specify a custom Comply port in Puppet Application Manager if you do not want to use the default port (30303). For more information, see System requirements on page 92.
• **Data retention.** The retention period for scan data can now be set on the Puppet Application Manager Config tab. For more information see, Scan results on page 118.

Resolved in this release:

• **Node Deletion.** A fix was added to ensure that nodes deleted in Puppet Enterprise are no longer listed in Comply as available for scanning.
• **License page node count.** Corrected an issue where the number of nodes displayed on the license page was not updated when a node was deleted in Puppet Enterprise.
• **Required installations page.** The required installations page that was part of the assessor install procedure was removed as it was no longer required.
• **Comply-graphql.** Fixed a known issue where the comply-graphql deployment did not become healthy after restoring Comply using Puppet Application Manager.
• **Rule ordering.** Corrected an issue where rules were not always displayed in the correct numerical order.

**Comply 2.1.0**

Released 7 October 2021.

New in this release:
• **Scan Reports.** The Comply UI has a new Scan Reports page that provides a report on rules passed/failed and node compliance from the most recent CIS scan. For more information, see CIS scan report details on page 118.

• **CIS-CAT Pro Assessor v4.9.0.** Comply 2.1.0 includes the latest version of the CIS-CAT assessor and its associated benchmark:
  - CentOS Linux 7 v3.1.2

• **Scanner upgrades.** Scanner upgrade in Comply is not forced but optional to allow better management of PE jobs.

  Note: By default in Comply 2.1.0, assessor upgrade does not happen automatically when you upgrade Comply. Assessor upgrade takes place when you instigate a Puppet Enterprise (PE) Puppet run job after Comply is upgraded. For more information, see Upgrade from Comply 2.2.2 to 2.3.0 on page 107.

Resolved in this release:

• **Desired compliance upgrades.** Fixed an issue where Windows 10 nodes lost their desired compliance after upgrade to Compliance 2.x

• **Upgrade statistics.** Resolved an issue where statistics were overwritten when multiple upgrades take place.

• **Service start up.** Updated Comply so that it now starts when IPv6 is disabled.

• **Preflight failure.** Fixed an issue where preflight checks failed during install when trailing newline returns were present in certificates.

• **Scan wizard.** The Comply scan wizard was updated to correct an issue where the environment name field did not revert to the previous saved value if the scan set up was cancelled.

Comply 2.0.0

Released August 2021.

New in this release:

• **CIS-CAT Pro Assessor v4.8.2.** Comply 2.0.0 includes the latest version of the CIS-CAT assessor and its associated benchmarks:
  - Apple macOS 10.14 v1.4.0
  - Apple macOS 10.15 v1.4.0
  - Apple macOS 11.0 v1.2.0
  - CentOS Linux 7 v3.1.1
  - CentOS Linux 8 v1.0.1
  - Debian Linux 8 v2.0.2
  - Microsoft Windows Server 2019 v1.2.1
  - Microsoft Windows Server 2019 STIG v1.0.1
  - Microsoft Windows 10 20H2 v1.10.1
  - Oracle Linux 7 v3.1.1
  - Oracle Linux 8 v1.0.1
  - Red Hat Linux 7 v3.1.1
  - Red Hat Linux 8 v1.0.1
  - Amazon Linux 2 v2.0.0
  - Microsoft Windows 10 21H1 v1.11.0
  - Microsoft Windows Server 2016 v1.3.0
  - Ubuntu Linux 20.04 LTS STIG v1.0.0

© 2022 Puppet, Inc., a Perforce company
• **Automatic upgrades of the CIS-CAT assessor.** Every time you upgrade your Comply application, the assessor automatically upgrades to the latest version. This update also includes the following changes to how you interact with Comply:
  - You can only run a desired compliance scan against nodes with the latest version of the assessor.
  - You can only run a custom scan against benchmarks with the latest version of the assessor.
  - On the node inventory screen, nodes without the latest assessor are highlighted red to indicate that they need upgrading.
  - You can no longer set a desired compliance benchmark against a node that does not have the latest version of the assessor.
  - When the assessor upgrades, custom profiles are automatically updated to use the new benchmarks and profiles, sending you a notification.

• **Assessor upgrades tab.** The Assessor upgrades tab on the Activity feed screen provides a summary of assessor upgrades, including the number of nodes that have passed or failed. Note that this only shows the status of your nodes after the upgrade, and does not update again, even if your nodes change to passing.

• **comply module Secure Sockets Layer (SSL).** This includes changes to how you install and upgrade the Comply module.

Resolved in this release:

• **Comply tries to install 7-zip on Windows.** The comply module no longer installs 7zip on Windows systems.
• **Windows Server Semi Annual Channel (SAC) builds are assigned the wrong CIS profile.** SAC builds are now assigned the correct Windows 2019 profile.

Security notice:

• **Vulnerability in 12.18.3-alpine image.** The release updates the alpine image to 15.13.0.
• **Vulnerability keycloak:15.0.0.** This release updates keycloak to version 15.0.0.
• **Vulnerability in dependencies.** This release upgrades NodeJS to version 14.17.1 and React to version 17.0.2.

For upgrade instructions, see Upgrade from Comply 2.2.2 to 2.3.0 on page 107.

---

**Comply known issues**

These are the known issues for the Puppet Comply 1.x and 2.x releases.

**Comply does not import node groups from PE when configured using rules**

Node groups are only imported for nodes explicitly pinned to the node group. If rules have been configured to match nodes to groups based on facts then Comply does not import these groups.

**Exceptions that are both resolved and expired disappear from the exceptions page**

If an exception is resolved before the expiry time is reached then the exception is removed from the Exceptions page after the expiry time.

**Reports export null data when custom profile filter is applied**

Exported reports are empty if a custom profile has been selected on the Profiles quick filter.

**Node group filtering does not work for deleted nodes**

Any nodes deleted before upgrading to 2.10 do not have node group information available. The Node Group quick filters on the Scan Report (Nodes tab) and Rule Detail pages do not apply to those nodes.
An exception might be incorrectly listed as active

If you create an exception that applies to a custom profile, but you then delete the custom profile, the exception is inactive and no longer affects scan reports. However, this update might not be immediately reflected in the Comply user interface. For example, the Exceptions page and the Rule Detail page might incorrectly indicate that the exception is still active.

Invalid information might be displayed on the Scan Report page

On the Scan Report page, when you hover over an item in the Profile column, you might see an invalid message that the profile has been deleted.

Comply UI pages not loading correctly after an upgrade

If the Comply UI pages are not loading correctly after an upgrade, delete the comply-graphql and comply-scarpy pods and wait for Comply to automatically restart.

Session timeout in Comply 2.2.0

Comply does not redirect users to the login screen on session timeout and some screens show error messages. Reloading the page in Comply fixes this issue.

Multiple filtering options

On pages where multiple filtering options are available, selecting one filter option does not affect the options presented by any other filter drop-down menu. This means filter drop-down menus display all available options for a given parameter and therefore invalid options might appear for a given filtering scenario.

GraphQL issue after Puppet Application Manager restore in Comply 2.1.0 or earlier

The Comply-GraphQL pod becomes stuck in CrashLoopBackOff after Comply is restored using Puppet Application Manager (PAM). This problem is due to an issue with the Hasura database used in Comply 2.1.0 or earlier. To resolve the issue, contact Puppet support for help or upgrade to Comply 2.2.0 or later.

Scan report metrics bar node count not matched in Scan Report page Nodes tab table in Comply 2.1.0

If an error occurs after a scan report is sent from PE to Comply (owing, for example, to the Comply module being out-of-date on the node), the number of nodes appearing in the Scan Report page Nodes tab table can differ from the node count that appears in the Scan report metrics bar.

Running scans on CentOS 7 with Comply 1.0.4

The CentOS 7 benchmark in Comply 1.0.4 has been updated to version 3.1.0. If you have already installed Comply and set desired compliance for your CentOS 7 nodes, run the following command on your comply-scarpy pod to update the benchmark version from 3.0.0 to 3.1.0:

```
kubectl exec --stdin --tty -n <namespace> $(kubectl get pods -n dio-comply | grep comply-scarpy | awk '{print $1}') -- /bin/scarp upgrade-assessor --assessor_version '4.6.0'
```

By taking this action, you help to ensure that Comply uses the latest CIS-CAT Pro Assessor Benchmark and profiles.

Running scan tasks in Puppet Enterprise (PE)

Comply uses PE tasks to run compliance scans on nodes. Although you can see the scan tasks in PE, we advise against running these tasks from PE because this practice can have unforeseen effects on both PE and Comply. Instead, run all CIS scans from Comply. You can view the scan results in both products.
Welcome to the Beginner’s guide to Comply! As a new user, you'll need to perform some initial installation and configuration tasks, and then we'll show you how to use the core features of Comply.

You're just a few steps away from enforcing compliant configurations across your infrastructure. Before you begin, we recommend familiarizing yourself with our terminology and Comply overview on page 6.

**Step 1: Install and configure Comply**

Use the main documentation to install and configure Comply. If you already completed these steps, proceed to step 2.

- Install Puppet Application Manager (PAM)
- Set up Comply

**Related concepts**

*Install PAM* on page 46

You can install Puppet-supported Puppet Application Manager on a single node or in an HA configuration. Both online and offline install packages are available. You can also install it on an existing Kubernetes cluster.

*Set up Comply* on page 93

To start using Puppet Comply, you must complete the setup process, using both Puppet Application Manager (PAM) and Puppet Enterprise (PE).

**Step 2: Set desired compliance**

Desired compliance is the benchmark and profile that you to assign to a particular node. It is what is scanned on that node by default. Most of the time, you only need to set this once for your nodes.

Based on fact information from PE, Comply automatically assigns an appropriate benchmark for each operating system, along with a Level 1 profile, to nodes that have not been set. Accepting this option is the quickest way to get up and running with desired compliance.

Alternatively, you can manually choose your own benchmark and profiles. For more information, see *Manually set desired compliance*.

**Step 3: Run a CIS scan**

You are now ready to run a scan.

This topic describes how to run an initial *ad hoc* scan.

1. In Comply, click *Scans*, and then *Run an ad hoc scan*.
2. In the *Benchmark* drop-down menu, select *Desired compliance* or a benchmark and profile of your choice.
   
   If you have not set desired compliance, follow the instructions in *Setting desired compliance*.
3. Next, select an option from the *Profile* drop-down menu. To use a custom profile for this scan, select the *Use an associated custom profile?* option and choose the relevant option from the *Custom profile* drop-down menu.
4. Click *Next* to review the PE credentials and environment you want the scan to run on.
5. Click **Next** to see the nodes selected for scanning.
   To scan only a subset of nodes, deselect any nodes that you want to exclude.

   **Debug mode:** By default, assessor logs are set to WARN level. To troubleshoot an issue, you can set the logging level to DEBUG for the scan by clicking **Run in debug mode**. The assessor logs can then be retrieved from the individual node.

   On Linux and macOS platforms the assessor log is located at:

   `/opt/puppetlabs/comply/Assessor-CLI/logs/assessor-cli.log`

   On Windows the assessor log is located at:

   `C:/ProgramData/PuppetLabs/comply/Assessor-CLI/logs/assessor-cli.log`

   Note that scanning in debug mode increases the size of the assessor log file significantly.

6. Click **Scan**. To confirm, click **Scan** again.
   You are taken to the **Activity feed**, which lists each scan. Scans are run as a task in PE. To see the details of the job, click the job ID to be taken to PE.

   **Tip:** You can also run a scan by clicking the **Scan nodes** button at the top right corner on several pages. This option uses the nodes listed on the page you are currently viewing.

7. Optionally, to review the results of your scan, navigate to the **Compliance Dashboard** page.
   See **Scan results** for a description of the scan data.

   Congratulations! You've completed the Beginner's guide to Comply. You're now familiar with the core features and know how to run CIS scans with Comply. To find out how you can enforce and automate CIS benchmarks on your failing nodes, see **Enforce CIS benchmarks** on page 121.

   **Related information**
   
   **Enforce CIS benchmarks** on page 121
   Puppet Comply provides visibility into your compliance status, but it cannot fix your failing nodes. Instead, you can use Puppet’s Compliance Enforcement Modules (CEM).

   **Custom profiles** on page 109
   A custom profile is a benchmark profile that you customize to fit your organization’s internally defined standards. You can base a custom profile on an existing benchmark and profile combination, and then specify which rules to apply.

   **Desired compliance** on page 108
   Set your desired compliance. This is the benchmark and profile that you assign to a particular node and that is scanned on that node by default. Generally, you set compliance only once for your nodes.

---

**Puppet Application Manager**

Before you can begin using Puppet Comply, you must install Puppet Application Manager. Puppet Application Manager is an administrative console that provides tools for managing Comply and other Puppet applications.

**Note:** Puppet Application Manager was previously called the platform admin console in the Comply documentation.

**What does Puppet Application Manager do?**

The Puppet Application Manager installation process sets up a managed Kubernetes cluster (or, if you prefer, adds Puppet Application Manager to your existing cluster). Comply runs on this Kubernetes cluster, and Puppet Application Manager manages the cluster for you.

© 2022 Puppet, Inc., a Perforce company
In the Puppet Application Manager UI, you can configure Comply, monitor the cluster's activity, upgrade to the latest version of the software, and back up your installation.

**How do I use Puppet Application Manager to deploy Comply?**

Once the cluster is ready, upload your Comply license and provide any needed configuration details about your installation in the Puppet Application Manager UI. You can then deploy the latest version of Comply with one click whenever you're ready.

- **Welcome to Puppet Application Manager (PAM)** on page 21
  Puppet Application Manager is an administrative console where you can install, access, and manage your Puppet applications. It is also where you can go to access upgrades to new Puppet applications releases.

- **Architecture overview** on page 29
  Puppet Application Manager (PAM) runs on Kubernetes. We provide several supported configurations for different use cases.

- **PAM system requirements** on page 32
  You can install Puppet Application Manager (PAM) on a Puppet-supported cluster or add PAM to a customer-supported cluster. Before installing PAM, ensure that your system meets these requirements.

- **Component versions in PAM releases** on page 45
  These tables show the versions of components included in recent Puppet Application Manager (PAM) releases.

- **Install PAM** on page 46
  You can install Puppet-supported Puppet Application Manager on a single node or in an HA configuration. Both online and offline install packages are available. You can also install it on an existing Kubernetes cluster.

- **Working with Puppet applications** on page 67
  You can install and upgrade Puppet applications using the Puppet Application Manager UI.

- **Maintenance and tuning** on page 71
  Follow these guidelines when you're tuning or performing maintenance on a node running Puppet Application Manager (PAM).

- **Upgrading PAM on a Puppet-supported cluster** on page 72
  Upgrade Puppet Application Manager (PAM) on a Puppet-supported cluster to take advantage of new features and bug fixes, and to upgrade your cluster to the latest version of Kubernetes when one is available.

- **Upgrading PAM on a customer-supported cluster** on page 76
  Upgrade Puppet Application Manager (PAM) on your own Kubernetes cluster to take advantage of new features and bug fixes.

- **Backing up PAM using snapshots** on page 77
  Snapshots are point-in-time backups of your Puppet Application Manager (PAM) deployment, which can be used to roll back to a previous state or restore your installation into a new cluster for disaster recovery.

- **Migrating PAM data to a new system** on page 80
  By using a snapshot, you can migrate your data to a new Puppet Application Manager (PAM) instance.

- **Disaster recovery with PAM** on page 86
  It is important to prepare your system and regularly capture full snapshots. This backs up your data and makes it easier to restore your system if disaster recovery is needed.

- **Troubleshooting PAM** on page 87
  Use this guide to troubleshoot issues with your Puppet Application Manager installation.

---

**Welcome to Puppet Application Manager (PAM)**

Puppet Application Manager is an administrative console where you can install, access, and manage your Puppet applications. It is also where you can go to access upgrades to new Puppet applications releases.

Useful links:
The Puppet Application Manager (PAM) UI provides administration functionality where you can access and manage your Puppet applications.

**PAM console menu**

Use the console menu at the top of the Puppet Application Manager UI to manage Puppet Application Manager itself. It has three tabs of interest to us:

- **Use the Dashboard** tab to:
  - Manage your applications
  - See version history
  - Set application configuration settings
  - Access support bundles for troubleshooting
  - Manage licenses
  - View files
  - Configure registry settings
- **Use the Cluster Management** tab to view current information on the nodes in your cluster. You can also use this tab to drain, and add nodes to your cluster.
- **Use the Snapshots** tab to create point-in-time backups of your deployment, which can be used to roll back to a previous state, or restore your installation into a new cluster for disaster recovery. For more information, see [Backing up PAM using snapshots on page 77](#).

You can also use the console menu to **Add a new application** and to log out of Puppet Application Manager.

**Application monitoring graphs**

When you have Prometheus installed, the **Dashboard** tab has an **Application** sub-tab that provides several simplified graphs for tracking overall health of the system.

- **Node CPU Usage (%)** shows when hosts are getting overwhelmed (high % usage).
- **Node Memory Usage (%)** shows when hosts are reaching full memory capacity that may result in processes being killed due to out-of-memory errors.
• **Node Available Storage (%)** shows when hosts are running out of storage. At 15%, pods may start to be evicted or reads/writes on databases are paused until more storage is made available.

• **Volume Available Storage (%)** shows when application persistent volumes are getting full (low %) that may lead to problems with a particular application. Note that -

  **Note:** As of the 30 June 2021 Puppet Application Manager release, the monitoring/Prometheus-Kubernetes pods limit their storage use and are expected to never fall below 10% available storage.

Puppet Application Manager HA architectures include Prometheus and Grafana. Metrics about how the system is working are sent to Prometheus, and can be displayed with Grafana. Grafana credentials are printed during install, or can be retrieved later with the following command:

```
kubectl -n monitoring get secret grafana-admin -o go-template='{{index .data "admin-user"|base64decode}}:{{index .data "admin-password"|base64decode}}'
```

**Related information**

[Backing up PAM using snapshots on page 77](#)

Snapshots are point-in-time backups of your Puppet Application Manager (PAM) deployment, which can be used to roll back to a previous state or restore your installation into a new cluster for disaster recovery.

**Release notes**

**PAM release notes**

These are the new features, enhancements, resolved issues, and deprecations for Puppet Application Manager.

**Restriction:** Because kURL does not support upgrading more than two Kubernetes versions at once, if you're upgrading from an older version of PAM, you might need to follow a specific upgrade path to avoid failures.

- If you're on PAM version 1.56.0 or earlier, you must upgrade to PAM 1.80.0 before upgrading to PAM 1.81.1 or later.

28 September 2022 (Puppet Application Manager 1.81.1)

New in this release:

- **Kubernetes version upgrade.** For standalone and HA installations, this version includes an upgrade of Kubernetes to version 1.23.9.

  **Important upgrade information:** The upgrade process takes place on all nodes, and first upgrades Kubernetes to version 1.22 before upgrading to version 1.23.9. For a three-node cluster, you can expect the upgrade process to take around an hour. Confirmations are required during the upgrade process.

  Additionally, because kURL can only be upgraded two minor versions at a time on page 92, if you're upgrading from PAM version 1.56.0 or earlier, you must upgrade to PAM 1.80.0 before upgrading to PAM 1.81.1.

  For legacy installations, Kubernetes remains on version 1.19.15. If you're not sure which installation type you're running, see [How to determine your version of Puppet Application Manager](#).

16 August 2022 (Puppet Application Manager 1.80.0)

New in this release:

- **Component upgrades to address CVEs.** To address various CVEs, this version upgrades containerd to 1.4.13, KOTS to 1.80.0, ekco to 0.19.6, and Goldpinger to 3.5.1-5.2.0.

Resolved in this release:

- Fixed an issue where legacy encryption keys didn't load properly during snapshot restores.

© 2022 Puppet, Inc., a Perforce company
2 August 2022 (Puppet Application Manager 1.76.2)

New in this release:

- **Component upgrades to address CVEs.** To address various CVEs, this version includes an upgrade of OpenEBS to version 3.2.0, an upgrade of Weave to version 2.8.1-20220720, an upgrade of Project Contour to version 1.21.1, and an upgrade of MinIO to version 2022-07-17T15-43-14Z.

20 July 2022 (Puppet Application Manager 1.76.1)

New in this release:

- **Support for Red Hat Enterprise Linux version 8.6.** Beginning with version 1.76.1, PAM can be successfully installed on systems running Red Hat Enterprise Linux version 8.6.
- **More log data is now retained.** To ensure that you and our Support team have the data you need in debugging scenarios, the size of the pod logs has been increased from 10 files of 10MiB each to 10 files of 50MiB each. This change increases the storage used in `/var/log/pods` by 400MiB.
- **Component upgrades to address CVEs.** To address various CVEs, this version includes an upgrade of Velero to version 1.9.0 and an upgrade of the Prometheus bundle to version 0.57.0-36.2.0.
- **Other component upgrades.** This version also includes an upgrade of Registry to version 2.8.1 and an upgrade of MinIO to version 2022-07-06T20-29-49Z.

Resolved in this release:

- Velero pods no longer get stuck in a pending state when creating a snapshot to be saved to internal storage on a Puppet-supported cluster.

23 June 2022 (Puppet Application Manager 1.72.1)

New in this release:

- **Component upgrades to address CVEs.** To address various CVEs, this version includes an upgrade of ekco to version 0.19.2 and an upgrade of kURL to v2022.06.17-0.

26 May 2022 (Puppet Application Manager 1.70.1)

New in this release:

- **Component upgrades to address CVEs.** To address various CVEs, this version includes an upgrade of Project Contour to version 1.21.0, an upgrade of Velero to version 1.8.1, and an upgrade of the Prometheus bundle to version 0.56.2-35.2.0.

Resolved in this release:

- Image garbage collection in Kubernetes installer-created clusters (embedded clusters) no longer removes images outside of the application's dedicated registry namespace.
- The **Deploy** button is now present in newly updated versions after the configuration is updated from the previously deployed version.
- Legends are now shown properly for the performance graphs on the dashboard.

12 April 2022 (Puppet Application Manager 1.68.0)

New in this release:

- **Install a specific version of an application.** When installing a Puppet application using the automated installation method, you now have the option to specify the application's version by passing the `--app-version-label=<version>` flag to the `kubectl kots install` command. For more information, go to the Automate PAM and Puppet application online installations on page 61.
- **Status reporting improvements.** The status reporting tools can now detect when an application is being upgraded.
• **Component upgrades to address CVEs.** To address various CVEs in Envoy, this version includes an upgrade of Project Contour to version 1.20.1.

• **Other component upgrades.** This version includes an upgrade of KOTS to version 1.68.0, which enables Kubernetes audit event logging by default and adds a 1 GB storage requirement for `/var/log/apiserver`.

Resolved in this release:

• During image garbage collection, images still in use by the cluster are no longer in danger of being deleted from the private registry in a Kubernetes installer-created cluster.

**1 March 2022 (Puppet Application Manager 1.64.0)**

Resolved in this release:

• Diffs are now shown correctly in the PAM UI.

• The OpenSSL package is no longer a prerequisite for successful installation on newer Red Hat Enterprise Linux 7 systems.

• You can now successfully install Puppet Application Manager on Red Hat Enterprise Linux 8 systems without the need to force-install the kurl-local audit-libs library.

**17 February 2022 (Puppet Application Manager 1.62.0)**

**Important:** Version 1.0.2 of the `puppetlabs/pam_firewall` module is now available. To avoid conflicts, upgrade the module **before** upgrading Puppet Application Manager to version 1.62.0.

New in this release:

• **Kubernetes version upgrade.** For standalone and HA installations, this version includes an upgrade of Kubernetes to version 1.21.8.

  **Important upgrade information:** The upgrade process takes place on all nodes, and first upgrades Kubernetes to version 1.20 before upgrading to version 1.21.8. For a three-node cluster, you can expect the upgrade process to take around an hour. Confirmations are required during the upgrade process.

For legacy installations (installed before May 2021), this version includes an upgrade of Kubernetes to version 1.19.15.

**Tip:** See How to determine your version of Puppet Application Manager if you're not sure which installation type you're running.

• **Prometheus enabled on standalone architecture.** Beginning with version 1.62.0 Prometheus is enabled by default on all new and existing standalone Puppet Application Manager installations. Prometheus requires an additional 350m CPU and 500MiB of memory, so ensure your system is properly sized before upgrading. Prometheus is an optional component; if you need to disable it to conserve resources, see Optional components on page 90.

• **Automatic certificate rotation.** By default, the self-signed certificates used by Project Contour and Envoy expire after one year. This version includes an update that auto-rotates those certificates before they expire.

• **Component upgrades to address CVEs.** To address various CVEs, this version includes an upgrade of containerd to version 1.4.12.

• **Other component upgrades.** This version includes an upgrade of KOTS to version 1.62.0.

Deprecated in this release:

• **Legacy architecture.** The legacy architecture, which was the version of Puppet Application Manager available for installation prior to May 2021, is now deprecated. (See How to determine your version of Puppet Application Manager if you need to confirm whether you're running the legacy architecture.) The legacy architecture utilizes Rook 1.0, which is incompatible with Kubernetes version 1.20 and newer versions. Kubernetes version 1.19 is no longer receiving security updates. Puppet will continue to update legacy architecture components other than Kubernetes until 30 June 2022. If security advisories against Kubernetes 1.19 arise, the remediation path is to
migrate to one of the newer architectures by following the instructions in Migrating PAM data to a new system on page 80.

Important: Before beginning the migration process from a legacy deployment you must upgrade to PAM version 1.62.0 with the `force-reapply-addons` flag included in the upgrade command. Find upgrade instructions at PAM legacy upgrades on page 74 and PAM offline legacy upgrades on page 75.

30 November 2021 (Puppet Application Manager 1.56.0)

This release includes an upgrade of KOTS to version 1.56.0, which adds the following improvements:

- **Improved support bundles:** Adds an option to upload a support bundle directly from Puppet Application Manager.
- **Improved troubleshooting:** Adds detailed information on failing pods to the Troubleshoot tab.

6 October 2021 (Puppet Application Manager 1.52.1)

New in this release:

- **Improved statuses.** More granular status levels are now available from the Application tab.
- **Component upgrades to address CVEs.** To address various CVEs, this version includes an upgrade of Kubernetes to 1.19.15.
- **Other component upgrades.** This version includes an upgrade of KOTS to version 1.52.1.

Resolved in this release:

- Generating a support bundle no longer results in unusually high memory use.
- Preflight check logs post to info level for progress messages and to error level for error messages.

25 August 2021 (Puppet Application Manager 1.49.0)

New in this release:

- **Component upgrades to address CVEs.** To address various CVEs, this version includes an upgrade of Kubernetes to 1.19.13, an upgrade of Project Contour to 1.18.0, and an upgrade of Velero to 1.6.2.
- **Goldpinger.** High availability architectures now include Goldpinger, which aids the debugging of network issues.
- **containerd upgrade.** This version includes an upgrade of containerd to version 1.4.6, and removes the need to use the `force-reapply-addons` option when upgrading.
- **Other component upgrades.** This version includes an upgrade of KOTS to version 1.49.0, an upgrade of ekco to 0.11.0, an upgrade of Prometheus to 0.49.0, and an upgrade of Rook to 1.5.12.

30 June 2021 (Puppet Application Manager 1.44.1)

New in this release:

- **Certificate auto-rotation for standalone architecture.** Certificates are now automatically rotated for the Kubernetes API and Puppet Application Manager UI in the standalone architecture. With this change, certificate auto-rotation is now supported in all Puppet Application Manager architectures.
- **Rook upgrades.** This version includes an upgrade of Rook in the high availability architecture to 1.5.11 and the version of Rook in the legacy architecture to 1.0.4-14.2.21. These upgrades address a vulnerability in Ceph components (CVE-2021-20288).
- **Prometheus upgrade.** This version includes an upgrade of Prometheus in the high availability and legacy architectures to 0.48.1. Additionally, Prometheus disk usage is now limited in order to preserve the storage space required for the usage charts on the Application tab.
- **Other component upgrades.** This version includes an upgrade of KOTS to version 1.44.1, an upgrade of Project Contour to version 1.15.1, and an upgrade of Weave to version 2.8.1.

Resolved in this release:
Snapshots can now successfully use the **Other S3-Compatible Storage** option as the storage destination.

To apply this update, add the `force-reapply-addons` option during upgrade. For example:

```
curl <url> | bash -s force-reapply-addons
```

**26 May 2021**

New in this release:

- **runC.** The version of runC has been upgraded to v1.0.0-rc95 to address CVE-2021-30465.

Known issues in this release:

- Running the KOTS installer with the `airgap` and `kurl-registry-ip` flags results in an error.
  As a workaround (if you do not have any applications already installed in the cluster), delete the registry service, recreate the registry service IP and then re-run the installation script with the `kurl-registry-ip` flag.

**10 May 2021 (Puppet Application Manager 1.40.0)**

New in this release:

- Distinct architectures for standalone and high availability deployments of the Puppet Application Manager platform. Standalone supports lower system requirements and resolves inherent flaws in using Ceph on a single node. High availability uses an updated version of Rook for faster, more reliable distributed storage.

  **Note:** It is not possible currently to upgrade to these architectures from existing installations. However, migrating applications between them is on the roadmap for a future release.

- The previous architecture is maintained as the legacy configuration. This version includes an upgrade of Kubernetes to 1.19.10; this upgrade process upgrades through Kubernetes 1.18, and happens on all nodes. It can take ~1 hour to do for a 3-node cluster, and requires confirmations during that period. It also includes an upgrade of Project Contour to version 1.14.1, adds Metrics Server 0.4.1, an upgrade of ekco to 0.10.1, and an upgrade of Prometheus to 2.26.0.

  For more information on legacy upgrades, see [PAM legacy upgrades](#) on page 74.

**15 April 2021 (Puppet Application Manager 1.38.0)**

New in this release:

- **Snapshots.** Puppet Application Manager now supports full (instance-level) snapshots, which can be used for application rollbacks and disaster recovery. For more information, see [Back up Puppet Application Manager using snapshots](#).

- **Component upgrades.** This version includes an upgrade of KOTS to version 1.38.0.

**17 February 2021 (Puppet Application Manager 1.29.3)**

New in this release:

- **Support for Ubuntu 20.04.** You can now run Puppet Application Manager on Ubuntu 20.04.

- **Component upgrades.** This version includes an upgrade of Prometheus to version 2.22.1 and Prometheus Operator to version 0.44.1, an upgrade of KOTS to version 1.29.3, an upgrade of Project Contour to version 1.12.0, and an upgrade of ekco to version 0.10.0.

**3 February 2021 (Puppet Application Manager 1.29.2)**

New in this release:

- **Component upgrades.** This version includes an upgrade of KOTS to version 1.29.2, an upgrade of Project Contour to version 1.11.0, and an upgrade of `containerd` to version 1.4.3.
Resolved in this release:

- During their initial preflight checks, new installations now pull images successfully and no longer report a `Failed to pull image` error.

7 December 2020

New in this release:

- **Support for Red Hat Enterprise Linux (RHEL) 8 and CentOS 8.** You can now run Puppet Application Manager on RHEL version 8 and CentOS version 8. To support this change, `containerd` is now used independently of Docker during the installation process.

- **Component upgrades.** This version includes an upgrade of Kubernetes to version 1.17.13.

**Related information**

- **Upgrading PAM on a Puppet-supported cluster** on page 72

  Upgrade Puppet Application Manager (PAM) on a Puppet-supported cluster to take advantage of new features and bug fixes, and to upgrade your cluster to the latest version of Kubernetes when one is available.

- **Back up PAM using snapshots** on page 77

  Snapshots are point-in-time backups of your Puppet Application Manager (PAM) deployment, which can be used to roll back to a previous state or restore your installation into a new cluster for disaster recovery.

**Known issues**

These are the known issues for Puppet Application Manager (PAM).

**PAM versions 1.72.1 and older cannot be installed on RHEL 8.6+ systems**

A known issue in kURL prevents Puppet Application Manager versions 1.72.1 and older from successfully installing on Red Hat Enterprise Linux (RHEL) version 8.6 and newer versions. To work around this issue, install or upgrade to Puppet Application Manager version 1.76.1 or a newer version, which support RHEL version 8.6.

**Velero fails if network file system (NFS) snapshot storage is misconfigured**

In Puppet Application Manager version 1.64.0 and newer versions, changes to the configuration of snapshot storage on a network file system (NFS) is appended to Velero containers, rather than replaced. This means that if NFS snapshot storage is misconfigured, attempts to fix the configuration do not correct the problem. This issue manifests as a failure of Velero to start up.

**OpenSSL package required for newer RHEL 7 systems with PAM 1.62.0**

Attempts to install Puppet Application Manager version 1.62.0 or older on newer Red Hat Enterprise Linux (RHEL) 7 systems fail unless the OpenSSL package is present on the system before installation. To work around this issue, run `yum install openssl` and then re-run the PAM installation script.

**Package updates with yum or DNF fail after upgrading PAM**

If you are unable to run `yum update` or `dnf upgrade` after a PAM upgrade, run one of the following commands to clean up a temporary module added by PAM:

```shell
yum module reset kurl.local
or
dnf module reset kurl.local
```
Architecture overview

Puppet Application Manager (PAM) runs on Kubernetes. We provide several supported configurations for different use cases.

PAM can run on Puppet-supported or customer-supported Kubernetes clusters. Due to potential variations in the architecture of customer-supported clusters, the architecture overview provided on this page assumes PAM is running on Puppet-supported clusters. For more information on installing on a customer-supported Kubernetes cluster, see Install Puppet applications using PAM on a customer-supported Kubernetes cluster on page 47.

Terminology

Throughout this documentation, we use a few terms to describe different roles nodes can take:

- **Primary** - A primary node runs core Kubernetes components (referred to as the Kubernetes control plane) as well as application workloads. At least three primaries are required to support high availability for Puppet Application Manager. These are also sometimes referred to as masters.
- **Secondary** - A secondary node runs application workloads. These are also sometimes referred to as workers.

Puppet Application Manager is built on the KOTS (Kubernetes off-the-Shelf) project, and we occasionally use its CLI tools (kubectl, kots) to manage the installation.

Standalone architecture

Standalone is optimized for limited resources, storing data directly on disk. If you need to remove optional components like Prometheus and Grafana to decrease resource utilization, see Optional components on page 90. While additional compute capacity can be added through secondary nodes, this does not provide increased resilience as data is only stored on the node where a component service runs.

For information on migrating data from standalone to HA deployments, see Migrating data between two systems with different architectures on page 85.

HA architecture

A high availability (HA) architecture provides high availability for scheduling application services during failure and uses Ceph for distributed storage in case of node failure. Individual applications may still experience some loss of availability (up to 10 minutes) if individual services do not have replicas and need to be rescheduled. For more information, see Reduce recovery time when a node fails on page 89. An HA implementation requires a cluster of three primary nodes. Additional compute capacity can be added through secondary nodes.

The HA architecture installs Prometheus and Alertmanager. These are used to provide system monitoring in the Puppet Application Manager UI. Prometheus and Alertmanager are unauthenticated on ports 30900 and 30903, and you are recommended to control access to these ports via firewall rules. For information on how to remove Prometheus and Alertmanager, see Optional components on page 90.

Puppet Application Manager architectures

The following diagram and lists outline some of the core components involved in standalone and HA architectures and how they communicate. For a detailed list of ports used by Puppet Application Manager, refer to the Cluster port requirements sections of the PAM system requirements on page 32. For firewall information, refer to Web URL and port requirements for firewalls on page 44.
Internal ports

All backplane ports may also be used for inter-process communication within a single host.

The following ports are only used within a single host for inter-process communication:
- TCP 6781, 6782, 6784 (Weave)
Standalone architecture

Puppet Application Manager
Lives on a cluster within a Linux host.
The PAM application includes the admin console, application services, and PostgreSQL.
PAM communicates out of the Linux host to fetch updates.

UI ports
The application UI communicates on 80/443 to the Linux host.
The admin console HTTPS UI communicates on 8800 to the Linux host.

Backplane and internal ports
Backplane ports include 6783 (TCP or UDP) and 10250 (TCP).
Backplane ports can also be used within a single host for inter-process communication.
These ports are only used within a single host for inter-process communication (Weave): 6781 (TCP), 6782 (TCP), and 6784 (UDP)

Additional default ports
30900: Prometheus UI
30902: Grafana UI
30903: Alertmanager UI

HA cluster architecture

Control plane (primaries)
Multiple primaries that can also run application workloads.
Structured as clusters within Linux hosts with a device or partition for Ceph.
Each primary hosts PAM and can run application services in addition to supporting either PostgreSQL or the admin console.

Workers (secondaries)
Can be added later to add capacity for running application workloads.
Structured as clusters within Linux hosts.

Network or Application Balancer
The balancer communicates out to the control plane (primaries) and workers (secondaries).
Receives admin console HTTPS UI communication over 8800.
Receives application UI communication over 80/443.
Network load balancer internal APIs communicate with primaries and secondaries over 6443.
To learn about setting up health checks for your load balancer, go to Load balancer health checks on page 71.

Backplane and internal ports
Backplane ports include 2379/2380 (TCP), 6783 (TCP or UDP), and 10250 (TCP).
Backplane ports can also be used within a single host for inter-process communication.
These ports are only used within a single host for inter-process communication (Weave): 6781 (TCP), 6782 (TCP), and 6784 (UDP)

Additional default ports
30900: Prometheus UI
30902: Grafana UI
30903: Alertmanager UI
**UNSUPPORTED: Legacy architecture**

**Note:** The legacy architecture utilizes Rook 1.0, which is incompatible with Kubernetes version 1.20 and newer versions. Kubernetes version 1.19 is no longer receiving security updates. The legacy architecture reached the end of its support lifecycle on **30 June 2022**, and Puppet no longer updates legacy architecture components.

The Puppet Application Manager legacy architecture reflects an older configuration that used Ceph 1.0 which hosted data directly on the file system. Installing the legacy architecture is no longer supported.

For information on upgrading to a newer version of the legacy architecture, see [PAM legacy upgrades](#) on page 74 and [PAM offline legacy upgrades](#) on page 75.

For information on migrating data from a legacy architecture to a standalone or HA architecture, go to our Support Knowledge Base instructions:

- Migrate to a supported PAM architecture for Continuous Delivery for PE
- Migrate to a supported PAM architecture for Comply

**Related information**

- [Reduce recovery time when a node fails](#) on page 89
- If a node running a non-replicated service like PostgreSQL fails, expect some service downtime.

- [Install Puppet applications using PAM on a customer-supported Kubernetes cluster](#) on page 47
- Use these instructions to install Puppet Application Manager and any Puppet applications on an existing Kubernetes cluster.

- [PAM legacy upgrades](#) on page 74
- The legacy architecture is no longer supported. However, if you have not yet migrated to a supported architecture, you can use this method to upgrade Puppet Application Manager (PAM).

- [PAM offline legacy upgrades](#) on page 75
- The legacy architecture is no longer supported. However, if you have not yet migrated to a supported architecture, you can use this method to upgrade Puppet Application Manager (PAM) on offline nodes.

- [Troubleshooting PAM](#) on page 87
- Use this guide to troubleshoot issues with your Puppet Application Manager installation.

**PAM system requirements**

You can install Puppet Application Manager (PAM) on a Puppet-supported cluster or add PAM to a customer-supported cluster. Before installing PAM, ensure that your system meets these requirements.

**Customer-supported cluster hardware requirements**

The following Kubernetes distributions are supported:

- Google Kubernetes Engine
- AWS Elastic Kubernetes Service

If you use a different distribution, contact [Puppet Support](#) for more information on compatibility with PAM.

**Application requirements:**

<table>
<thead>
<tr>
<th>Application</th>
<th>CPU</th>
<th>Memory</th>
<th>Storage</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Delivery for Puppet Enterprise (PE)</td>
<td>3 CPU</td>
<td>8 GB</td>
<td>280 GB</td>
<td>Ingress, NodePort 8000</td>
</tr>
</tbody>
</table>

**Note:** NodePort is configurable

© 2022 Puppet, Inc., a Perforce company
<table>
<thead>
<tr>
<th>Application</th>
<th>CPU</th>
<th>Memory</th>
<th>Storage</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>Puppet Comply</td>
<td>7 CPU</td>
<td>7 GB</td>
<td>35 GB</td>
<td>Ingress, NodePort 30303</td>
</tr>
</tbody>
</table>

**Note:** NodePort is configurable

Make sure that your Kubernetes cluster meets the minimum requirements:

- Kubernetes version 1.19-1.23.
- A default storage class that can be used for relocatable storage.
- A standard Ingress controller that supports websockets (we have tested with Project Contour and NGINX).
- We currently test and support Google Kubernetes Engine (GKE) clusters.

**Cluster ports:** In addition to the NodePorts used by your Puppet applications, make sure that TCP port 443 is open for your ingress controller.

**Puppet-supported HA cluster hardware requirements**

A high availability (HA) configuration uses multiple servers to provide availability in the event of a server failure. A majority of servers must be available to preserve service availability. Below are suggested configurations for each application.

**Continuous Delivery for Puppet Enterprise (PE)**

Three servers (referred to as primaries during installation) with the following minimum requirements:

<table>
<thead>
<tr>
<th>CPU</th>
<th>Memory</th>
<th>Storage</th>
<th>Open ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 CPU</td>
<td>10 GB</td>
<td>100 GB on an unformatted storage device.</td>
<td>TCP: 80, 443, 2379, 2380, 6443, 6783, 8000, 8800, and 10250</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 GB for /var/log/apiserver for Kubernetes audit logs.</td>
<td>UDP: 6783, 6784</td>
</tr>
<tr>
<td></td>
<td></td>
<td>An additional 100 GB for /var/lib. You can use separate filesystems if necessary, but it is not a requirement to do so. For your reference, here is how the usage is roughly divided:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2 GB for /var/lib/etcd</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 10 GB for /var/lib/rook (plus buffer)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 32 GB for /var/lib/kubelet</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 40 GB for /var/lib/containerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> The storage backend prefers the file system inhabited by /var/lib/rook to remain below 70% utilization.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSDs (or similarly low-latency storage) are recommended for /var/lib/etcd and /var/lib/rook.</td>
<td></td>
</tr>
</tbody>
</table>
Puppet Comply

Three servers (referred to as primaries during installation) with the following minimum requirements:

<table>
<thead>
<tr>
<th>CPU</th>
<th>Memory</th>
<th>Storage</th>
<th>Open ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 CPU</td>
<td>10 GB</td>
<td>100 GB on an unformatted storage device.</td>
<td>TCP: 80, 443, 2379, 2380, 6443, 6783, 8800, 10250, and 30303</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 GB for /var/log/apiserver for Kubernetes audit logs.</td>
<td>UDP: 6783, 6784</td>
</tr>
<tr>
<td></td>
<td></td>
<td>An additional 100 GB for /var/lib. You can use separate filesystems if necessary, but it is not a requirement to do so. For your reference, here is how the usage is roughly divided:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2 GB for /var/lib/etcd</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 10 GB for /var/lib/rook (plus buffer)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 32 GB for /var/lib/kubelet</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 40 GB for /var/lib/containerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The storage backend prefers the file system inhabited by /var/lib/rook to remain below 70% utilization.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSDs (or similarly low-latency storage) are recommended for /var/lib/etcd and /var/lib/rook.</td>
<td></td>
</tr>
</tbody>
</table>

Continuous Delivery for Puppet Enterprise (PE) and Puppet Comply

Three servers (referred to as primaries during installation) with the following minimum requirements:
For a detailed example of an HA configuration running Continuous Delivery for PE and Puppet Comply, see [Example of an HA cluster that supports CDPE and Comply](#).

### Networking requirements

Gigabit Ethernet (1GbE) and a latency of less than 10 milliseconds (ms) between cluster members is sufficient for most deployments. For more information on networking for specific Puppet Application Manager components, see the documentation for [Ceph](#) and [etcd](#).

### Cluster port requirements

Puppet Application Manager (PAM) uses the following ports in an HA cluster architecture:

<table>
<thead>
<tr>
<th>Category</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Puppet application ports</td>
<td>443</td>
<td>TCP</td>
<td>Web UI, Relies on Server Name Indication to route requests to the application</td>
<td>Browser</td>
</tr>
<tr>
<td>Continuous Delivery for Puppet Enterprise (PE) ports</td>
<td>8000</td>
<td>TCP</td>
<td>Webhook service</td>
<td>Source control</td>
</tr>
<tr>
<td><strong>Category</strong></td>
<td><strong>Port</strong></td>
<td><strong>Protocol</strong></td>
<td><strong>Purpose</strong></td>
<td><strong>Source</strong></td>
</tr>
<tr>
<td>------------------------</td>
<td>----------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td><strong>Puppet Comply ports</strong></td>
<td>30303</td>
<td>TCP</td>
<td>Communication with Puppet Enterprise (PE)</td>
<td>PE instance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Platform ports</strong></td>
<td>2379, 2380</td>
<td>TCP</td>
<td>High availability (HA) communication</td>
<td>etcd on the Kubernetes host.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Only needs to be open between the cluster's primary nodes.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6443</td>
<td>TCP</td>
<td>Kubernetes API</td>
<td>Admin workstation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Might be useful to expose to workstations.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6781, 6782</td>
<td>TCP</td>
<td>Kubernetes networking - Weave</td>
<td>Kubernetes host</td>
</tr>
<tr>
<td></td>
<td>6783</td>
<td>TCP/UDP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6784</td>
<td>UDP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8800</td>
<td>TCP</td>
<td>PAM</td>
<td>Admin browser</td>
</tr>
<tr>
<td></td>
<td>9001</td>
<td>TCP</td>
<td>Internal registry in offline installs only. Requires configuring an Ingress to use this port.</td>
<td>Kubernetes host</td>
</tr>
<tr>
<td></td>
<td>10250</td>
<td>TCP</td>
<td>Kubernetes cluster management</td>
<td>Kubernetes host</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Only communicates in one direction, from a primary to other primaries and secondaries.</td>
<td></td>
</tr>
</tbody>
</table>

Additionally, these ports are configured by default: 30900 (Prometheus UI), 30902 (Grafana UI), and 30903 (Alertmanager UI)

For Kubernetes-specific information, refer to [Networking Requirements in the Kurl documentation](#).

**IP address range requirements**

*Important:* Puppet Application Manager must be installed on nodes with static IP assignments because IP addresses cannot be changed after installation.
Ensure that IP address ranges 10.96.0.0/22 and 10.32.0.0/22 are locally accessible. See Resolve IP address range conflicts for instructions.

**Note:** The minimum size for CIDR blocks used by PAM are:

- /23 for pod and service CIDRs
- Default of /22 is recommended to support future expansion

### Antivirus and antimalware considerations

Antivirus and antimalware software can impact PAM and its applications or prevent them from functioning properly.

To avoid issues, exclude the following directories from antivirus and antimalware tools that scan disk write operations:

- /var/lib/rook
- /var/lib/kubelet
- /var/lib/containerd

### Firewall modules

If you use the puppetlabs/firewall module to manage your cluster's firewall rules with Puppet, be advised that purging unknown rules from changes breaks Kubernetes communication. To avoid this, apply the puppetlabs/pam_firewall module before installing Puppet Application Manager.

If you've already installed PAM, apply the pam_firewall module and then restart the kube-proxy service to recreate its iptables rules by running the following on a primary:

```
systemctl restart kubelet
kubectl -n kube-system delete pod -l k8s-app=kube-proxy
kubectl -n kube-system delete pod -l name=weave-net
```

For more information, see the PAM firewall module.

### Supported operating systems

Puppet Application Manager and the applications it supports can be installed on these operating systems:

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Supported versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon Linux</td>
<td>2</td>
</tr>
<tr>
<td>CentOS</td>
<td>7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8.1, 8.2, 8.3, 8.4</td>
</tr>
<tr>
<td>Oracle Linux</td>
<td>7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8.0, 8.1, 8.2, 8.3, 8.4, 8.5</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux (RHEL)</td>
<td>7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 8.1, 8.2, 8.3, 8.4, 8.5, 8.6 with PAM version 1.76.1 or newer</td>
</tr>
<tr>
<td>Ubuntu (General availability kernels)</td>
<td>16.04, 18.04, 20.04</td>
</tr>
</tbody>
</table>
Puppet-supported standalone hardware requirements

Here are the suggested configurations for standalone installations.

**Continuous Delivery for Puppet Enterprise (PE)**

<table>
<thead>
<tr>
<th>CPU</th>
<th>Memory</th>
<th>Storage</th>
<th>Open ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 CPU</td>
<td>8 GB</td>
<td>130 GB for /var/lib and /var/openebsThis is primarily divided among:</td>
<td>TCP: 80, 443, 2379, 2380, 6443, 6783, 8000, 8800, and 10250 UDP: 6783, 6784</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2 GB for /var/lib/etcd</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 32 GB for /var/lib/kubelet</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 40 GB for /var/lib/containerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 50 GB for /var/openebs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP: 80, 443, 2379, 2380, 6443, 6783, 8000, 8800, and 10250 UDP: 6783, 6784</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UDP: 6783, 6784</td>
<td></td>
</tr>
</tbody>
</table>

**Puppet Comply**

<table>
<thead>
<tr>
<th>CPU</th>
<th>Memory</th>
<th>Storage</th>
<th>Open ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 CPU</td>
<td>7 GB</td>
<td>130 GB for /var/lib and /var/openebsThis is primarily divided among:</td>
<td>TCP: 80, 443, 2379, 2380, 6443, 6783, 8000, 8800, and 30303 UDP: 6783, 6784</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2 GB for /var/lib/etcd</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 32 GB for /var/lib/kubelet</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 40 GB for /var/lib/containerd</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 50 GB for /var/openebs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP: 80, 443, 2379, 2380, 6443, 6783, 8000, 8800, and 30303 UDP: 6783, 6784</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>UDP: 6783, 6784</td>
<td></td>
</tr>
</tbody>
</table>

**Cluster port requirements**

Puppet Application Manager (PAM) uses the following ports in a standalone architecture:

<table>
<thead>
<tr>
<th>Category</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Puppet application ports</td>
<td>442</td>
<td>TCP</td>
<td>Web UI</td>
<td>Browser</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Relies on Server Name Indication to route requests to the application.</td>
<td></td>
</tr>
<tr>
<td>Continuous Delivery for Puppet Enterprise (PE) ports</td>
<td>8000</td>
<td>TCP</td>
<td>Webhook service</td>
<td>Source control</td>
</tr>
<tr>
<td>Puppet Comply ports</td>
<td>30303</td>
<td>TCP</td>
<td>Communication with Puppet Enterprise</td>
<td>PE instance</td>
</tr>
<tr>
<td>Category</td>
<td>Port</td>
<td>Protocol</td>
<td>Purpose</td>
<td>Source</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>Platform ports</td>
<td>6443</td>
<td>TCP</td>
<td>Kubernetes API</td>
<td>Admin workstation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Might be useful to expose to workstations.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6781, 6782</td>
<td>TCP</td>
<td>Kubernetes networking - Weave</td>
<td>Kubernetes host</td>
</tr>
<tr>
<td></td>
<td>6783</td>
<td>TCP/UDP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6784</td>
<td>UDP</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>8800</td>
<td>TCP</td>
<td>PAM</td>
<td>Admin browser</td>
</tr>
<tr>
<td></td>
<td>9001</td>
<td>TCP</td>
<td>Internal registry in offline installs only.</td>
<td>Kubernetes host</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Requires configuring an Ingress to use this port.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10250</td>
<td>TCP</td>
<td>Kubernetes cluster management</td>
<td>Kubernetes host</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Only communicates in one direction, from a primary to other primaries and secondaries.</td>
<td></td>
</tr>
</tbody>
</table>

Additionally, these ports are configured by default: 30900 (Prometheus UI), 30902 (Grafana UI), and 30903 (Alertmanager UI)

For Kubernetes-specific information, refer to Networking Requirements in the Kurl documentation.

**IP address range requirements**

**Important:** Puppet Application Manager must be installed on nodes with static IP assignments because IP addresses cannot be changed after installation.

Ensure that IP address ranges 10.96.0.0/22 and 10.32.0.0/22 are locally accessible. See Resolve IP address range conflicts for instructions.

**Note:** The minimum size for CIDR blocks used by PAM are:

- /24 for pod and service CIDRs
- Default of /22 is recommended to support future expansion

**Antivirus and antimalware considerations**

Antivirus and antimalware software can impact PAM and its applications or prevent them from functioning properly.

To avoid issues, exclude the following directories from antivirus and antimalware tools that scan disk write operations:
• `/var/openebs`
• `/var/lib/kubelet`
• `/var/lib/containerd`

**Firewall modules**

If you use the `puppetlabs/firewall` module to manage your cluster's firewall rules with Puppet, be advised that purging unknown rules from changes breaks Kubernetes communication. To avoid this, apply the `puppetlabs/pam_firewall` module before installing Puppet Application Manager.

If you've already installed PAM, apply the `pam_firewall` module and then restart the `kube-proxy` service to recreate its iptables rules by running the following on a primary:

```
systemctl restart kubelet
kubectl -n kube-system delete pod -l k8s-app=kube-proxy
kubectl -n kube-system delete pod -l name=weave-net
```

For more information, see the PAM firewall module.

**Detailed hardware requirements**

For additional compute capacity, you can horizontally scale HA and standalone architectures by adding secondary nodes. During installation, only add secondaries after setting up all primaries.

You can add secondaries to HA and standalone architectures; however in standalone architectures, secondaries do not increase availability of the application, and data storage services are pinned to the host they start on and cannot be moved.

Here are the baseline requirements to run cluster services on primaries and secondaries. Any Puppet applications require additional resources on top of these requirements.
<table>
<thead>
<tr>
<th>Node type</th>
<th>CPU</th>
<th>Memory</th>
<th>Storage</th>
<th>Open ports</th>
</tr>
</thead>
</table>
| Primary   | 4 CPU| 7 GB   | At least 50 GB on an unformatted storage device in addition to application-specific storage (below) for the Ceph storage backend. This can be satisfied by multiple devices if more storage is needed later, but should be balanced across primaries. 1 GB for `/var/log/apiserver` for Kubernetes audit logs. An additional 100 GB for `/var/lib`. You can use separate filesystems if necessary, but it is not a requirement to do so. For your reference, here is how the usage is roughly divided:  
  • 2 GB for `/var/lib/etcd`  
  • 10 GB for `/var/lib/rook` (plus buffer)  
  • 32 GB for `/var/lib/kubelet`  
  • 40 GB for `/var/lib/containerd` | TCP: 80, 443, 2379, 2380, 6443, 6783, 8800, and 10250  
  UDP: 6783, 6784 |
| Secondary | 1 CPU| 1.5 GB | 1 GB for `/var/log/apiserver` for Kubernetes audit logs.  
80 GB for `/var/lib`. You can use separate filesystems if necessary, but it is not a requirement to do so. For your reference, here is how the usage is roughly divided:  
  • 32 GB for `/var/lib/kubelet`  
  • 40 GB for `/var/lib/containerd` | |

Applications are composed of multiple smaller services, so you can divide CPU and memory requirements across multiple servers. The listed ports can be accessed from all primaries and secondaries, but only need to be exposed on nodes you include in your load balancer. Apply application-specific storage to all primary nodes.
Application-specific requirements:

<table>
<thead>
<tr>
<th>Application</th>
<th>CPU</th>
<th>Memory</th>
<th>Storage</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Delivery for Puppet Enterprise (PE)</td>
<td>3 CPU</td>
<td>8 GB</td>
<td>50 GB</td>
<td>80, 443, 8000</td>
</tr>
<tr>
<td>Puppet Comply</td>
<td>7 CPU</td>
<td>7 GB</td>
<td>50 GB</td>
<td>80, 443, 30303</td>
</tr>
</tbody>
</table>

The minimum recommended size for a secondary node is 4 CPU and 8 GB of memory to allow some scheduling flexibility for individual services.

**Example of an HA cluster capable of running Continuous Delivery for PE and Comply**

An HA cluster capable of running both Continuous Delivery for Puppet Enterprise (PE) and Puppet Comply requires 10 CPU and 15 GB of application-specific memory in addition to per-node baselines. You can create a cluster from 4 CPU, 8 GB nodes. Each primary uses all CPU and 7 GB of memory for cluster services, providing 0 CPU and 1 GB of memory for application workloads; each secondary uses 1 CPU and 1.5 GB of memory for cluster services, providing 3 CPU and 6.5 GB of memory for application workloads. Create the cluster as follows:

- Three primaries provide an excess of 3 GB of memory for application workloads. Each primary must have 150 GB of storage in an unformatted, unpartitioned storage device for Ceph and 100 GB of storage for `/var/lib`.
- Three secondaries provide an excess of 9 CPU and 19.5 GB of memory for application workloads. Each secondary must have 80 GB of storage for `/var/lib`. 
This diagram illustrates the suggested three-node configuration for a cluster capable of running Continuous Delivery for Puppet Enterprise (PE) and Puppet
### Web URL and port requirements for firewalls

Puppet Application Manager interacts with external web URLs for a variety of installation, configuration, upgrade, and deployment tasks. Puppet Application Manager uses the following web URLs for internal and outbound network traffic.

<table>
<thead>
<tr>
<th>Category</th>
<th>URLs</th>
</tr>
</thead>
</table>
| Puppet Application Manager and platform | • get.replicated.com  
• registry.replicated.com  
• proxy.replicated.com  
• api.replicated.com  
• k8s.kurl.sh  
• kurl-sh.s3.amazonaws.com  
• replicated.app  
• registry-data.replicated.com |
| Container registries                    | • gcr.io  
• docker.io  
• index.docker.io  
• registry-1.docker.io  
• auth.docker.io  
• production.cloudflare.docker.com  
• quay.io |
| Puppet Enterprise                       | • pup.pt  
• forgeapi.puppet.com  
• pm.puppetlabs.com  
• amazonaws.com  
• s3.amazonaws.com  
• rubygems.org |

For information about containers and firewalls, refer to the [Networking Requirements in the Kurl documentation](#).
Firewall modules

If you use the puppetlabs/firewall module to manage your cluster's firewall rules with Puppet, be advised that purging unknown rules from changes breaks Kubernetes communication. To avoid this, apply the puppetlabs/pam_firewall module before installing Puppet Application Manager.

If you've already installed PAM, apply the pam_firewall module and then restart the kube-proxy service to recreate its iptables rules by running the following on a primary:

```bash
systemctl restart kubelet
kubectl -n kube-system delete pod -l k8s-app=kube-proxy
kubectl -n kube-system delete pod -l name=weave-net
```

For more information, see the PAM firewall module.

Supported browsers

The following browsers are supported for use with the Puppet Application Manager UI:

<table>
<thead>
<tr>
<th>Browser</th>
<th>Supported versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Chrome</td>
<td>Current version as of release</td>
</tr>
<tr>
<td>Mozilla Firefox</td>
<td>Current version as of release</td>
</tr>
<tr>
<td>Microsoft Edge</td>
<td>Current version as of release</td>
</tr>
<tr>
<td>Apple Safari</td>
<td>Current version as of release</td>
</tr>
</tbody>
</table>

Component versions in PAM releases

These tables show the versions of components included in recent Puppet Application Manager (PAM) releases.

<table>
<thead>
<tr>
<th>Component</th>
<th>PAM 1.68.0</th>
<th>PAM 1.70.1</th>
<th>PAM 1.72.1</th>
<th>PAM 1.76.1</th>
<th>PAM 1.76.2</th>
<th>PAM 1.80.0</th>
<th>PAM 1.81.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kubernetes</td>
<td>1.21.8</td>
<td>1.21.8</td>
<td>1.21.8</td>
<td>1.21.8</td>
<td>1.21.8</td>
<td>1.21.8</td>
<td>1.23.9</td>
</tr>
<tr>
<td>KOTS</td>
<td>1.68.0</td>
<td>1.70.1</td>
<td>1.72.1</td>
<td>1.76.1</td>
<td>1.76.1</td>
<td>1.80.0</td>
<td>1.81.1</td>
</tr>
<tr>
<td>kURL</td>
<td>v2022.04.08-0v2022.05.19-0v2022.06.17-0v2022.07.15-2v2022.07.20-0v2022.08.08-0v2022.08.22-0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Contour</td>
<td>1.20.1</td>
<td>1.21.0</td>
<td>1.21.0</td>
<td>1.21.0</td>
<td>1.21.0</td>
<td>1.21.1</td>
<td>1.22.0</td>
</tr>
<tr>
<td>Registry</td>
<td>2.7.1</td>
<td>2.7.1</td>
<td>2.7.1</td>
<td>2.8.1</td>
<td>2.8.1</td>
<td>2.8.1</td>
<td>2.8.1</td>
</tr>
<tr>
<td>Prometheus bundle</td>
<td>0.49.0-17.1</td>
<td>0.56.2-35.2.0</td>
<td>0.56.2-35.2.0</td>
<td>0.57.0-36.2.0</td>
<td>0.57.0-36.2.0</td>
<td>0.57.0-36.2.0</td>
<td>0.58.0-39.4.0</td>
</tr>
<tr>
<td>containerd</td>
<td>1.4.12</td>
<td>1.4.12</td>
<td>1.4.12</td>
<td>1.4.12</td>
<td>1.4.12</td>
<td>1.4.13</td>
<td>1.4.13</td>
</tr>
<tr>
<td>Velero</td>
<td>1.6.2</td>
<td>1.8.1</td>
<td>1.8.1</td>
<td>1.9.0</td>
<td>1.9.0</td>
<td>1.9.0</td>
<td>1.9.1</td>
</tr>
<tr>
<td>ekco</td>
<td>0.16.0</td>
<td>0.16.0</td>
<td>0.19.2</td>
<td>0.19.2</td>
<td>0.19.2</td>
<td>0.19.6</td>
<td>0.20.0</td>
</tr>
<tr>
<td>Kubernetes Metrics Server</td>
<td>0.4.1</td>
<td>0.4.1</td>
<td>0.4.1</td>
<td>0.4.1</td>
<td>0.4.1</td>
<td>0.4.1</td>
<td>0.4.1</td>
</tr>
<tr>
<td>Rook (HA only)</td>
<td>1.5.12</td>
<td>1.5.12</td>
<td>1.5.12</td>
<td>1.5.12</td>
<td>1.5.12</td>
<td>1.5.12</td>
<td>1.5.12</td>
</tr>
</tbody>
</table>
Looking up component versions

To view a list of the component versions included in your current version of PAM, run the following:

```
kubectl get installer -o jsonpath="{.items[].spec}" | jq
```

You can find more information about the current PAM version's component versions by navigating to the website appropriate to your installation type:

- **HA installations**: https://kurl.sh/puppet-application-manager
- **Standalone installations**: https://kurl.sh/puppet-application-manager-standalone

Install PAM

You can install Puppet-supported Puppet Application Manager on a single node or in an HA configuration. Both online and offline install packages are available. You can also install it on an existing Kubernetes cluster.

Refer to the Architecture overview on page 29 for guidance on choosing which Puppet-supported Kubernetes cluster configuration is most appropriate for your needs.

**Important:** The Puppet-supported Puppet Application Manager cluster brings its own container runtime as part of the kURL installation.

- Do not install a container runtime from your operating system (OS) vendor or third-party.
- Do not install PAM on a node that has previously hosted a container runtime from your OS vendor or third-party.

Installing a different container runtime on a node, even if you installed and removed the packages before you installed PAM, causes failures that persist even after you've uninstalled the runtime.

For information on installing Puppet Application Manager on an existing Kubernetes cluster, see Install Puppet applications using PAM on a customer-supported Kubernetes cluster on page 47.

- **Install Puppet applications using PAM on a customer-supported Kubernetes cluster** on page 47
  Use these instructions to install Puppet Application Manager and any Puppet applications on an existing Kubernetes cluster.

- **PAM HA online installation** on page 49
  The Puppet Application Manager (PAM) installation process creates a Kubernetes cluster for you and walks you through installing your Puppet application on the cluster.

- **PAM HA offline installation** on page 53
  Use these instructions to install Puppet Application Manager (PAM) in an air-gapped or offline environment where the Puppet Application Manager host server does not have direct access to the internet.

- **PAM standalone online installation** on page 56
  The Puppet Application Manager (PAM) installation process sets up the application manager (with a simple Kubernetes installation for container orchestration) for you and installs the application on the single-node cluster.

- **PAM standalone offline installation** on page 59
  Use these instructions to install Puppet Application Manager (PAM) in an offline environment where the Puppet Application Manager host server does not have direct access to the internet.
During a fresh online installation of Puppet Application Manager (PAM) and a Puppet application, you have the option to configure the software automatically rather than completing the installation script interview.

During a fresh offline installation of Puppet Application Manager (PAM) and a Puppet application, you have the option to configure the software automatically rather than completing the installation script interview.

Different uninstall procedures are required for Puppet-supported and customer-supported clusters

**Install Puppet applications using PAM on a customer-supported Kubernetes cluster**

Use these instructions to install Puppet Application Manager and any Puppet applications on an existing Kubernetes cluster.

**Before you begin**

1. If you haven’t already done so, install kubectl.

2. Puppet Application Manager is expected to work on any certified Kubernetes distribution that meets the following requirements. We validated and support:
   - Google Kubernetes Engine
   - AWS Elastic Kubernetes Service
   - If you use a different distribution, contact Puppet Support for more information on compatibility with PAM.

3. Make sure your Kubernetes cluster meets the minimum requirements:
   - Kubernetes version 1.19-1.23.
   - A default storage class that can be used for relocatable storage.
   - A standard Ingress controller that supports websockets (we have tested with Project Contour and NGINX).
   - We currently test and support Google Kubernetes Engine (GKE) clusters.

   **Note:** If you’re using self-signed certificates on your Ingress controller, you must ensure that your job hardware nodes trust the certificates. Additionally, all nodes that use Continuous Delivery for PE webhooks must trust the certificates, or SSL checking must be disabled on these nodes.

   **Important:** If you are installing Puppet Comply on Puppet Application Manager, the ingress controller must be configured to allow request payloads of up to 32 MB. Ingress controllers used by Amazon EKS commonly default to a 1 MB maximum — this causes all report submissions to fail.

   The ingress must have a generous limit for total connection time. Setting the connection timeout to infinity in conjunction with an idle timeout is recommended.

4. If you are setting up Puppet Application Manager behind a proxy server, the installer supports proxies configured via HTTP_PROXY/HTTPS_PROXY/NO_PROXY environment variables.

   **Restriction:** Using a proxy to connect to external version control systems is currently not supported.

Installation takes several (mostly hands-off) minutes to complete.

1. Install the KOTS (Kubernetes off-the-shelf software) plugin on a workstation that has kubectl access to the cluster. Your kubectl configuration must have sufficient privileges to create cluster-level roles and permissions:

   ```bash
curl https://kots.io/install | bash
   ```
2. If you are performing an offline install, ensure the required images are available in a local registry.
   a) Download the release assets matching the CLI version using the following command:

   ```
curl -LO https://github.com/replicatedhq/kots/releases/download/v$(kubectl kots version | head -n1 | cut -d' ' -f3)/kotsadm.tar.gz
```

   b) Extract the images and push them into a private registry. Registry credentials provided in this step must have push access. These credentials are not stored anywhere or reused later.

   ```
kubectl kots admin-console push-images ./kotsadm.tar.gz
   --private.registry.host=/puppet-application-manager
   --registry-username <rw-username>
   --registry-password <rw-password>
```

   c) Install Puppet Application Manager using images pushed in the previous step. Registry credentials provided in this step only need to have read access, and they are stored in a Kubernetes secret in the current namespace. These credentials are used to pull the images.

   ```
kubectl kots install puppet-application-manager
   --kotsadm-namespace puppet-application-manager
   --kotsadm-registry <private.registry.host>
   --registry-username <ro-username>
   --registry-password <ro-password>
```

   **Note:** If you are setting up Puppet Application Manager behind a proxy server, add the `--copy-proxy-env` flag to this command to copy the proxy-related environment values from your environment.

   d) You can use similar commands to upload images from the application bundle to your registry to continue to use read-only access when pulling images. Use the same registry namespace (`puppet-application-manager`) to pull application images.

   ```
kubectl kots admin-console push-images ./<application-release>.airgap
   --kotsadm-namespace puppet-application-manager
   --kotsadm-registry <private.registry.host>
   --registry-username <rw-username>
   --registry-password <rw-password>
```

3. To perform an online install of Puppet Application Manager on your cluster, run the following commands from a workstation that has kubectl access to the cluster.

   ```
kubectl kots install puppet-application-manager --namespace <target namespace>
```

   This installs Puppet Application Manager on the cluster and sets up a port forward on the ClusterIP.

4. Navigate to `http://localhost:8800` and follow the prompts to be guided through the process of uploading a license for the application, configuring a local registry (for offline installs), checking to make sure your infrastructure meets system requirements, and configuring the application.

   **Note:** If you are performing an offline install, download the application bundle and provide it when prompted.

   **Tip:** Clusters like GKE often restrict ports to 30000-32767. The webhook for Continuous Delivery for PE defaults to port 8000. To update this port to something in the allowed range, when configuring the application, use the following steps:

   a. On the Puppet Application Manager Dashboard page, under **Config > Optional configuration**, select **View options for using a proxy or external load balancer**.
   b. Enter a new value for **Webhook service port**.

© 2022 Puppet, Inc., a Perforce company
5. To configure your installation further, click Config. On this tab, you can configure a public hostname, root user, and other settings. These are written as Kubernetes secrets in the deployment manifests. For information on how to configure your application, see the documentation for that application:
   - Configure Continuous Delivery for PE in an online environment
   - Configure Comply in an online environment

6. To use cert-manager, in the Customize endpoints section, select I have cert manager and in the annotations section, add yours. For example:

   ```
kubernetes.io/ingress.class: nginx
cert-manager.io/cluster-issuer: letsencrypt-prod
   ```

7. When you are happy with your configuration, click Save config to deploy the application.

Follow the instructions for configuring and deploying your Puppet applications on Puppet Application Manager. For general information, go to Install applications via the PAM UI on page 68.

For more information on installing Continuous Delivery for PE online, see Install Continuous Delivery for PE.

For more information on installing Comply online, see Install Comply online.

Related information

Upgrade PAM on a customer-supported online cluster on page 76
Upgrading Puppet Application Manager (PAM) on a customer-supported online Kubernetes cluster can be done with a single command.

Upgrade PAM on a customer-supported offline cluster on page 76
Upgrading Puppet Application Manager (PAM) on a customer-supported offline Kubernetes cluster requires a few simple kubectl commands.

PAM HA online installation

The Puppet Application Manager (PAM) installation process creates a Kubernetes cluster for you and walks you through installing your Puppet application on the cluster.

Before you begin

1. Review the Puppet Application Manager system requirements.
2. Note that Swap is not supported for use with this version of Puppet Application Manager (PAM). The installation script attempts to disable Swap if it is enabled.
3. (Optional) If necessary, prepare additional steps related to SELinux and Firewalld:

   The PAM installation script disables SELinux and Firewalld by default. If you want to keep SELinux enabled, append the `--s preserve-selinux-config` switch to the PAM install command. This may require additional configuration to adapt SELinux policy to the installation.

   If you want to keep Firewalld enabled:

   a. Make sure Firewalld is installed on your system.

   b. To prevent the installation from disabling Firewalld, provide a patch file to the PAM install command using `--s installer-spec-file=patch.yaml`, where `patch.yaml` is the name of your patch file. For reference, here's an example patch file that enables Firewalld during installation, starts the service if it isn't running, and adds rules to open relevant ports:

   ```
   apiVersion: cluster.kurl.sh/v1beta1
   kind: Installer
   metadata:
     name: patch
   spec:
     firewalldConfig:
       firewalld: enabled
       command: ["/bin/bash", "-c"]
   ```
4. Ensure that IP address ranges 10.96.0.0/22 and 10.32.0.0/22 are locally accessible. See Resolve IP address range conflicts on page 88 for instructions.

   **Note:** The minimum size for CIDR blocks used by Puppet Application Manager are:
   - **Standalone** - /24 for pod and service CIDRs
   - **HA** - /23 for pod and service CIDRs
   - Default of /22 is recommended to support future expansion

5. If you are setting up Puppet Application Manager behind a proxy server, the installer supports proxies configured via HTTP_PROXY/HTTPS_PROXY/NO_PROXY environment variables.

   **Restriction:** Using a proxy to connect to external version control systems is currently not supported.

6. Set all nodes used in your HA implementation to the UTC timezone.

7. If you use the puppetlabs/firewall module to manage your cluster's firewall rules with Puppet, be advised that purging unknown rules from changes breaks Kubernetes communication. To avoid this, apply the puppetlabs/pam_firewall module before installing Puppet Application Manager.

This installation process results in a Puppet Application Manager instance that is configured for high availability. Installation takes several minutes (mostly hands-off) to complete.

For more context about HA components and structure, refer to the **HA architecture** section of the Architecture overview on page 29.
1. Install and configure a load balancer (or two if you want to segment internal and external traffic - for more information, see Architecture overview on page 29). Round-robin load balancing is sufficient. For an HA cluster, the following is required:

   • A network (L4, TCP) load balancer for port 6443 across primary nodes. This is required for Kubernetes components to continue operating in the event that a node fails. The port is only accessed by the Kubernetes nodes and any admins using kubectl.
   
   • A network (L4, TCP) or application (L7, HTTP/S) load balancer for ports 80, and 443 across all primaries and secondaries. This maintains access to applications in event of a node failure. Include 8800 if you want external access to the Puppet Application Manager UI.

   **Note:** Include port 8000 for webhook callbacks if you are installing Continuous Delivery for PE.
2. From the command line of your first primary node, run the installation script:

```bash
curl -sSL https://k8s.kurl.sh/puppet-application-manager | sudo bash
```

**Tip:** If you're installing Puppet Application Manager behind a proxy server, using `sudo` might cause the installation to fail. Try running the command as root and replace `sudo bash` with `bash`.

**Note:** An unformatted, unpartitioned storage device is required.

By default this installation automatically uses devices (under `/dev`) matching the pattern `vd[b-z]`. Attach a device to each host. Only devices that match the pattern, and are unformatted, are used.

If necessary, you can override this pattern by providing a patch during installation; append `-s installer-spec-file=patch.yaml` to the installation command.

```yaml
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
metadata:
  name: patch
spec:
  rook:
    blockDeviceFilter: "sd[b-z]"
```

a) When prompted for a load balancer address, enter the address of the DNS entry for your load balancer.

b) The installation script prints the address and password (only shown once, so make careful note of it) for Puppet Application Manager:

```text
---
Kotsadm: http://<PUPPET APPLICATION MANAGER ADDRESS>:8800
Login with password (will not be shown again): <PASSWORD>
---
```

**Note:** If you lose this password or wish to change it, see Reset the PAM password on page 88 for instructions.

c) When the installation script is complete, run `bash -l` to reload the shell.

**Tip:** If the installation script fails, run the following and upload the results to the Puppet Support team:

```bash
kubectl support-bundle https://kots.io
```

If you're installing as the root user, run the command directly:

```bash
/usr/local/bin/kubectl-support_bundle https://kots.io
```

3. Add two additional primary nodes to the installation by following the instructions in the install script:

To add MASTER nodes to this installation, run the following script on your other nodes:

```bash
curl -sSL https://k8s.kurl.sh/puppet-application-manager-unstable/join.sh | sudo bash -s kubernetes-master-address=...
```

If you want to keep SELinux enabled, append the `-s preserve-selinux-config` switch to the install command.

4. Add the two new nodes to your load balancer.
5. Navigate to the Puppet Application Manager UI using the address provided by the installation script (http://<PUPPET APPLICATION MANAGER ADDRESS>:8800) and follow the prompts.

The Puppet Application Manager UI is where you manage Puppet applications. You’ll be guided through the process of setting up SSL certificates, uploading a license, and checking to make sure your infrastructure meets application system requirements.

Follow the instructions for configuring and deploying your Puppet applications on Puppet Application Manager. For more information, see Install applications via the PAM UI on page 68.

For more information on installing Continuous Delivery for PE online, see Install Continuous Delivery for PE.

For more information on installing Comply online, see Install Comply online.

Related information

Reset the PAM password on page 88
As part of the installation process, Puppet Application Manager (PAM) generates a password for you. You can update this password to one of your choosing after installation.

PAM system requirements on page 32
You can install Puppet Application Manager (PAM) on a Puppet-supported cluster or add PAM to a customer-supported cluster. Before installing PAM, ensure that your system meets these requirements.

Resolve IP address range conflicts on page 88
When installing Puppet Application Manager, IP address ranges 10.96.0.0/22 and 10.32.0.0/22 must not be used by other nodes on the local network.

Architecture overview on page 29
Puppet Application Manager (PAM) runs on Kubernetes. We provide several supported configurations for different use cases.

Using sudo behind a proxy server on page 92
Many of the commands you run to install or configure Puppet Application Manager (PAM) require root access. In the PAM documentation, commands that require root access use `sudo` to elevate privileges. If you're running PAM behind a proxy, `sudo` might not work correctly. If you're having trouble running commands with `sudo`, and you're behind a proxy, try switching to the `root` user and running the command without `sudo`.

PAM HA offline installation

Use these instructions to install Puppet Application Manager (PAM) in an air-gapped or offline environment where the Puppet Application Manager host server does not have direct access to the internet.

Before you begin

1. Review the Puppet Application Manager system requirements.
2. Note that Swap is not supported for use with this version of Puppet Application Manager (PAM). The installation script attempts to disable Swap if it is enabled.
3. (Optional) If necessary, prepare additional steps related to SELinux and Firewalld:

   The PAM installation script disables SELinux and Firewalld by default. If you want to keep SELinux enabled, append the `--preserve-selinux-config` switch to the PAM install command. This may require additional configuration to adapt SELinux policy to the installation.

   If you want to keep Firewalld enabled:
   a. Make sure Firewalld is installed on your system.
   b. To prevent the installation from disabling Firewalld, provide a patch file to the PAM install command using `--installer-specified-file=patch.yaml`, where `patch.yaml` is the name of your patch file. For reference, here's an example patch file that enables Firewalld during installation, starts the service if it isn't running, and adds rules to open relevant ports:

```
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
```
metadata:
  name: patch
spec:
  firewalldConfig:
    firewalld: enabled
    command: ["/bin/bash", "-c"]
    args: ["echo 'net.ipv4.ip_forward = 1' | tee -a /etc/sysctl.conf && sysctl -p"]
  firewalldCmds:
  - ["--permanent", "--zone=trusted", "--add-interface=weave"]
  - ["--zone=external", "--add-masquerade"]
  # SSH port
  - ["--permanent", "--zone=public", "--add-port=22/tcp"]
  # HTTPS port
  - ["--permanent", "--zone=public", "--add-port=443/tcp"]
  # Kubernetes etcd port
  - ["--permanent", "--zone=public", "--add-port=2379-2830/tcp"]
  # Kubernetes API port
  - ["--permanent", "--zone=public", "--add-port=6443/tcp"]
  # Weave Net port
  - ["--permanent", "--zone=public", "--add-port=6783/udp"]
  # Weave Net port
  - ["--permanent", "--zone=public", "--add-port=6783-6874/tcp"]
  # CD4PE Webhook callback port (uncomment line below if needed)
  # - ["--permanent", "--zone=public", "--add-port=8000/tcp"]
  # KOTS UI port
  - ["--permanent", "--zone=public", "--add-port=8800/tcp"]
  # CD4PE Local registry port (offline only, uncomment line below if needed)
  # - ["--permanent", "--zone=public", "--add-port=9001/tcp"]
  # Kubernetes component ports (kubelet, kube-scheduler, kube-controller)
  - ["--permanent", "--zone=public", "--add-port=10250-10252/tcp"]
  # Reload firewall rules
  - ["--reload"]
    bypassFirewalldWarning: true
    disableFirewalld: false
    hardFailOnFirewalld: false
    preserveConfig: false

4. Ensure that IP address ranges 10.96.0.0/22 and 10.32.0.0/22 are locally accessible. See Resolve IP address range conflicts on page 88 for instructions.

Note: The minimum size for CIDR blocks used by Puppet Application Manager are:
- **Standalone** - /24 for pod and service CIDRs
- **HA** - /23 for pod and service CIDRs
- Default of /22 is recommended to support future expansion

5. Ensure that the nodes can resolve their own hostnames, through either local host mapping or a reachable DNS server.

6. Set all nodes used in your HA implementation to the UTC timezone.

7. If you use the puppetlabs/firewall module to manage your cluster's firewall rules with Puppet, be advised that purging unknown rules from changes breaks Kubernetes communication. To avoid this, apply the puppetlabs/pam_firewall module before installing Puppet Application Manager.

8. If you're restoring a backup from a previous cluster, make sure you include the kurl-registry-ip=<YOUR_IP_ADDRESS> installation option. For more information, see Migrating PAM data to a new system on page 80.

This installation process results in a basic Puppet Application Manager instance that is configured for optional high availability. Installation takes several minutes (mostly hands-off) to complete.
For more context about HA components and structure, refer to the **HA architecture** section of the **Architecture overview** on page 29.

1. Install and configure a load balancer (or two if you want to segment internal and external traffic - for more information, see **Architecture overview** on page 29). Round-robin load balancing is sufficient. For an HA cluster, the following is required:

   - A network (L4, TCP) load balancer for port 6443 across primary nodes. This is required for Kubernetes components to continue operating in the event that a node fails. The port is only accessed by the Kubernetes nodes and any admins using `kubectl`.
   - A network (L4, TCP) or application (L7, HTTP/S) load balancer for ports 80, and 443 across all primaries and secondaries. This maintains access to applications in event of a node failure. Include 8800 if you want external access to the Puppet Application Manager UI.

   **Note:** Include port 8000 for webhook callbacks if you are installing Continuous Delivery for PE.

2. From a workstation with internet access, download the cluster installation bundle (note that this bundle is ~4GB):

   ```
   https://k8s.kurl.sh/bundle/puppet-application-manager.tar.gz
   ```

3. Copy the installation bundle to your primary and secondary nodes and unpack it:

   ```
   tar xzf puppet-application-manager.tar.gz
   ```

4. Run the installation command:

   ```
   cat install.sh | sudo bash -s airgap
   ```

   **Note:** An unformatted, unpartitioned storage device is required.

   By default this installation automatically uses devices (under `/dev`) matching the pattern `vd[b-z]`. Attach a device to each host. Only devices that match the pattern, and are unformatted, are used.

   If necessary, you can override this pattern by providing a patch during installation; append `-s installer-spec-file=patch.yaml` to the installation command.

   ```
   apiVersion: cluster.kurl.sh/v1beta1
   kind: Installer
   metadata:
     name: patch
   spec:
     rook:
       blockDeviceFilter: "sd[b-z]"
   ```

   a) When prompted for a load balancer address, enter the address of the DNS entry for your load balancer.

   b) The installation script prints the address and password (only shown once, so make careful note of it) for Puppet Application Manager:

   ```
   ---
   Kotsadm: http://<PUPPET APPLICATION MANAGER ADDRESS>:8800
   Login with password (will not be shown again): <PASSWORD>
   ---
   ```

   **Note:** If you lose this password or wish to change it, see Reset the PAM password on page 88 for instructions.
5. Add two additional primary nodes to your offline installation using the instructions provided in the install script:

```
To add MASTER nodes to this installation, copy and unpack this bundle on
your other nodes, and run the following:
cat ./join.sh | sudo bash -s airgap
kubernetes-master-address=...
```

6. Add the two new nodes to your load balancer.

7. Navigate to the Puppet Application Manager UI using the address provided by the installation script (http://<PUPPET APPLICATION MANAGER ADDRESS>:8800) and follow the prompts.

The Puppet Application Manager UI is where you manage Puppet applications. You’ll be guided through the process of setting up SSL certificates, uploading a license, and checking to make sure your infrastructure meets application system requirements.

Follow the instructions for installing your Puppet applications on Puppet Application Manager. For more information, see Install applications via the PAM UI on page 68.

For more information on installing Continuous Delivery for PE offline, see Install Continuous Delivery for PE in an offline environment.

For more information on installing Comply offline, see Install Comply offline.

**Related information**

- Reset the PAM password on page 88
  As part of the installation process, Puppet Application Manager (PAM) generates a password for you. You can update this password to one of your choosing after installation.

- PAM system requirements on page 32
  You can install Puppet Application Manager (PAM) on a Puppet-supported cluster or add PAM to a customer-supported cluster. Before installing PAM, ensure that your system meets these requirements.

- Resolve IP address range conflicts on page 88
  When installing Puppet Application Manager, IP address ranges 10.96.0.0/22 and 10.32.0.0/22 must not be used by other nodes on the local network.

- Architecture overview on page 29
  Puppet Application Manager (PAM) runs on Kubernetes. We provide several supported configurations for different use cases.

  - Using sudo behind a proxy server on page 92
  Many of the commands you run to install or configure Puppet Application Manager (PAM) require root access. In the PAM documentation, commands that require root access use `sudo` to elevate privileges. If you're running PAM behind a proxy, `sudo` might not work correctly. If you're having trouble running commands with `sudo`, and you're behind a proxy, try switching to the root user and running the command without `sudo`.

**PAM standalone online installation**

The Puppet Application Manager (PAM) installation process sets up the application manager (with a simple Kubernetes installation for container orchestration) for you and installs the application on the single-node cluster.

**Before you begin**

1. Review the Puppet Application Manager system requirements.
2. Note that Swap is not supported for use with this version of Puppet Application Manager (PAM). The installation script attempts to disable Swap if it is enabled.
3. (Optional) If necessary, prepare additional steps related to SELinux and Firewalld:

The PAM installation script disables SELinux and Firewalld by default. If you want to keep SELinux enabled, append the \texttt{-s preserve-selinux-config} switch to the PAM install command. This may require additional configuration to adapt SELinux policy to the installation.

If you want to keep Firewalld enabled:

\textbf{a.} Make sure Firewalld is installed on your system.

\textbf{b.} To prevent the installation from disabling Firewalld, provide a patch file to the PAM install command using \texttt{-s installer-spec-file=patch.yaml}, where \texttt{patch.yaml} is the name of your patch file. For reference, here's an example patch file that enables Firewalld during installation, starts the service if it isn't running, and adds rules to open relevant ports:

```yaml
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
metadata:
  name: patch
spec:
  firewalldConfig:
    firewalld: enabled
    command: ["/bin/bash", "-c"]
    args: ["echo 'net.ipv4.ip_forward = 1' | tee -a /etc/sysctl.conf && sysctl -p"]
  firewalldCmds:
    - ["--permanent", "--zone=trusted", "--add-interface=weave"]
    - ["--zone=external", "--add-masquerade"]
    # SSH port
    - ["--permanent", "--zone=public", "--add-port=22/tcp"]
    # HTTPS port
    - ["--permanent", "--zone=public", "--add-port=443/tcp"]
    # Kubernetes etcd port
    - ["--permanent", "--zone=public", "--add-port=2379-2830/tcp"]
    # Kubernetes API port
    - ["--permanent", "--zone=public", "--add-port=6443/tcp"]
    # Weave Net port
    - ["--permanent", "--zone=public", "--add-port=6783/udp"]
    # Weave Net port
    - ["--permanent", "--zone=public", "--add-port=6783-6874/tcp"]
    # CD4PE Webhook callback port (uncomment line below if needed)
    # - ["--permanent", "--zone=public", "--add-port=8000/tcp"]
    # KOTS UI port
    - ["--permanent", "--zone=public", "--add-port=8800/tcp"]
    # CD4PE Local registry port (offline only, uncomment line below if needed)
    # - ["--permanent", "--zone=public", "--add-port=9001/tcp"]
    # Kubernetes component ports (kubelet, kube-scheduler, kube-controller)
    - ["--permanent", "--zone=public", "--add-port=10250-10252/tcp"]
    # Reload firewall rules
    - ["--reload"]
    bypassFirewalldWarning: true
    disableFirewalld: false
    hardFailOnFirewalld: false
    preserveConfig: false
```

4. Ensure that IP address ranges 10.96.0.0/22 and 10.32.0.0/22 are locally accessible. See \texttt{Resolve IP address range conflicts} on page 88 for instructions.

5. If you use the \texttt{puppetlabs/firewall} module to manage your cluster's firewall rules with Puppet, be advised that purging unknown rules from changes breaks Kubernetes communication. To avoid this, apply the \texttt{puppetlabs/pam_firewall} module before installing Puppet Application Manager.
This installation process results in a basic Puppet Application Manager instance. Installation takes several (mostly hands-off) minutes to complete.

1. From the command line of your node, run the installation script:

```
curl -sSL https://k8s.kurl.sh/puppet-application-manager-standalone | sudo bash
```

**Tip:** If you're installing Puppet Application Manager behind a proxy server, using `sudo` might cause the installation to fail. Try running the command as root (use command `sudo su -`) and replace `sudo bash` with `bash`.

a) When the installation script prints the Puppet Application Manager address and password, make a careful note of these credentials:

```
---
Kotsadm: http://<PUPPET APPLICATION MANAGER ADDRESS>:8800
Login with password (will not be shown again): <PASSWORD>
---
```

**Note:** If you lose this password or wish to change it, see Reset the PAM password on page 88 for instructions.

b) When the installation script is complete, run `bash -l` to reload the shell.

**Tip:** If the installation script fails, run the following and upload the results to the Puppet Support team:

```
kubectl support-bundle https://kots.io
```

If you're installing as the root user, run the command directly:

```
/usr/local/bin/kubectl-support_bundle https://kots.io
```

2. Navigate to the Puppet Application Manager UI using the address provided by the installation script (http://<PUPPET APPLICATION MANAGER ADDRESS>:8800) and follow the prompts.

The Puppet Application Manager UI is where you manage Puppet applications. You’ll be guided through the process of setting up SSL certificates, uploading a license, and checking to make sure your infrastructure meets application system requirements.

Follow the instructions for configuring and deploying your Puppet applications on Puppet Application Manager. For more information, see Install applications via the PAM UI on page 68.

For more information on installing Continuous Delivery for PE online, see Install Continuous Delivery for PE.

For more information on installing Comply online, see Install Comply online.

**Related information**

Reset the PAM password on page 88

As part of the installation process, Puppet Application Manager (PAM) generates a password for you. You can update this password to one of your choosing after installation.

PAM system requirements on page 32

You can install Puppet Application Manager (PAM) on a Puppet-supported cluster or add PAM to a customer-supported cluster. Before installing PAM, ensure that your system meets these requirements.

Resolve IP address range conflicts on page 88

When installing Puppet Application Manager, IP address ranges 10.96.0.0/22 and 10.32.0.0/22 must not be used by other nodes on the local network.

Architecture overview on page 29
Puppet Application Manager (PAM) runs on Kubernetes. We provide several supported configurations for different use cases.

**Using sudo behind a proxy server** on page 92

Many of the commands you run to install or configure Puppet Application Manager (PAM) require root access. In the PAM documentation, commands that require root access use `sudo` to elevate privileges. If you're running PAM behind a proxy, `sudo` might not work correctly. If you're having trouble running commands with `sudo`, and you're behind a proxy, try switching to the root user and running the command without `sudo`.

**PAM standalone offline installation**

Use these instructions to install Puppet Application Manager (PAM) in an offline environment where the Puppet Application Manager host server does not have direct access to the internet.

**Before you begin**

1. Review the [Puppet Application Manager system requirements](#).
2. Note that Swap is not supported for use with this version of Puppet Application Manager (PAM). The installation script attempts to disable Swap if it is enabled.
3. (Optional) If necessary, prepare additional steps related to SELinux and Firewalld:

The PAM installation script disables SELinux and Firewalld by default. If you want to keep SELinux enabled, append the `-s preserve-selinux-config` switch to the PAM install command. This may require additional configuration to adapt SELinux policy to the installation.

If you want to keep Firewalld enabled:

a. Make sure Firewalld is installed on your system.

b. To prevent the installation from disabling Firewalld, provide a patch file to the PAM install command using `-s installer-spec-file=patch.yaml`, where `patch.yaml` is the name of your patch file. For reference, here's an example patch file that enables Firewalld during installation, starts the service if it isn't running, and adds rules to open relevant ports:

```yaml
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
metadata:
  name: patch
spec:
  firewalldConfig:
    firewalld: enabled
    command: ["/bin/bash", "-c"]
    args: ["echo 'net.ipv4.ip_forward = 1' | tee -a /etc/sysctl.conf && sysctl -p"]
  firewalldCmds:
    - ["--permanent", "--zone=trusted", "--add-interface=weave"]
    - ["--zone=external", "--add-masquerade"]
    # SSH port
    - ["--permanent", "--zone=public", "--add-port=22/tcp"]
    # HTTPS port
    - ["--permanent", "--zone=public", "--add-port=443/tcp"]
    # Kubernetes etcd port
    - ["--permanent", "--zone=public", "--add-port=2379-2830/tcp"]
    # Kubernetes API port
    - ["--permanent", "--zone=public", "--add-port=6443/tcp"]
    # Weave Net port
    - ["--permanent", "--zone=public", "--add-port=6783/udp"]
    # Weave Net port
    - ["--permanent", "--zone=public", "--add-port=6783-6874/tcp"]
    # CD4PE Webhook callback port (uncomment line below if needed)
    # - ["--permanent", "--zone=public", "--add-port=8000/tcp"]
    # KOTS UI port
    - ["--permanent", "--zone=public", "--add-port=8800/tcp"]
```
This installation process results in a basic Puppet Application Manager instance. Installation takes several (mostly hands-off) minutes to complete.

1. From a workstation with internet access, download the cluster installation bundle (note that this bundle is ~4GB):

   https://k8s.kurl.sh/bundle/puppet-application-manager-standalone.tar.gz

2. Copy the installation bundle to the host node and unpack it:

   tar xzf puppet-application-manager-standalone.tar.gz

3. Run the installation command:

   cat install.sh | sudo bash -s airgap

   a) The installation script prints the address and password (only shown once, so make careful note of it) for Puppet Application Manager:

   ---
   Kotsadm: http://<PUPPET APPLICATION MANAGER ADDRESS>:8800
   Login with password (will not be shown again): <PASSWORD>
   ---

   **Note:** If you lose this password or wish to change it, see Reset the PAM password on page 88 for instructions.

4. Navigate to the Puppet Application Manager UI using the address provided by the installation script (http://<PUPPET APPLICATION MANAGER ADDRESS>:8800) and follow the prompts.

   The Puppet Application Manager UI is where you manage Puppet applications. You’ll be guided through the process of setting up SSL certificates, uploading a license, and checking to make sure your infrastructure meets application system requirements.

   Follow the instructions for configuring and deploying your Puppet applications on Puppet Application Manager. For more information, see Install applications via the PAM UI on page 68.
For more information on installing Continuous Delivery for PE offline, see Install Continuous Delivery for PE in an offline environment.

For more information on installing Comply offline, see Install Comply offline.

**Related information**

*Reset the PAM password* on page 88
As part of the installation process, Puppet Application Manager (PAM) generates a password for you. You can update this password to one of your choosing after installation.

*PAM system requirements* on page 32
You can install Puppet Application Manager (PAM) on a Puppet-supported cluster or add PAM to a customer-supported cluster. Before installing PAM, ensure that your system meets these requirements.

*Resolve IP address range conflicts* on page 88
When installing Puppet Application Manager, IP address ranges 10.96.0.0/22 and 10.32.0.0/22 must not be used by other nodes on the local network.

*Architecture overview* on page 29
Puppet Application Manager (PAM) runs on Kubernetes. We provide several supported configurations for different use cases.

*Using sudo behind a proxy server* on page 92
Many of the commands you run to install or configure Puppet Application Manager (PAM) require root access. In the PAM documentation, commands that require root access use `sudo` to elevate privileges. If you're running PAM behind a proxy, `sudo` might not work correctly. If you're having trouble running commands with `sudo`, and you're behind a proxy, try switching to the `root` user and running the command without `sudo`.

**Automate PAM and Puppet application online installations**

During a fresh online installation of Puppet Application Manager (PAM) and a Puppet application, you have the option to configure the software automatically rather than completing the installation script interview.

**Before you begin**

Ensure that your system meets the [PAM system requirements](#) on page 32.

1. Install Puppet Application Manager. For detailed instructions, see [PAM HA online installation](#) on page 49.
2. Define the configuration values for your Puppet application installation, using Kubernetes YAML format.

```yaml
apiVersion: kots.io/v1beta1
kind: ConfigValues
metadata:
  name: app-config
spec:
  values:
    accept_eula:
      value: has_accepted_eula
    annotations:
      value: "ingress.kubernetes.io/force-ssl-redirect: 'false'"
    hostname:
      value: "<HOSTNAME>"
    root_password:
      value: "<ROOT ACCOUNT PASSWORD>"
```

**Tip:** View the keyword names for all settings by clicking View files > upstream > config.yaml in Puppet Application Manager.

Replace the values indicated:

- Replace `<HOSTNAME>` with a hostname you want to use to configure an Ingress and to tell job hardware agents and web hooks how to connect to it. You might need to configure your DNS to resolve the hostname to your Kubernetes hosts.
- Replace `<ROOT ACCOUNT PASSWORD>` your chosen password for the application root account. The root account is used to administer your application and has full access to all resources and application-wide settings. This account must NOT be used for testing and deploying control repositories or modules.
- **Optional.** These configuration values disable HTTP-to-HTTPS redirection, so that SSL can be terminated at the load balancer. If you want to run the application over SSL only, change the `force-ssl-redirect` annotation to `true`.
- **Optional.** If your load balancer requires HTTP health checks, you can now enable Ingress settings that do not require Server Name Indication (SNI) for `/status`. To enable this setting, add the following to the config values statement:

```yaml
enable_lb_healthcheck:
  value: "1"
```

**Note:** The automated installation automatically accepts the Puppet application end user license agreement (EULA). Unless Puppet has otherwise agreed in writing, all software is subject to the terms and conditions of the Puppet Master License Agreement located at https://puppet.com/legal.

3. Write your license file and the configuration values generated in step 1 to the following locations:

- Write your license file to `./replicated_license.yaml`
- Write your configuration values to `./replicated_config.yaml`

4. Add the Puppet application definition to Puppet Application Manager with the license file and configuration values, passing in the Puppet Application Manager password you set in step 4:

```
kubectl kots install <APPLICATION NAME> --namespace default --shared-password <YOUR CHOSEN PASSWORD> --port-forward=false --license-file ./replicated_license.yaml --config-values ./replicated_config.yaml
```

**Note:** If you want to install a specific version of the application, include the `--app-version-label=<VERSION>` flag in the install command.

5. Wait five minutes to allow the software time to process the change.

Your configuration values are applied, and if preflight checks have passed, the application is deployed and in the process of starting up.

The application's status on the Application tab is shown as Missing for several minutes while deployment is underway. To monitor the deployment's progress, run kubectl get pods --watch.

When the deployment is complete, the application status changes to Ready.

7. Update your DNS or /etc/hosts file to include the hostname you chose during configuration.

8. Installation is now complete! Navigate to https://<HOSTNAME> and sign into Puppet application.

Related information
PAM HA online installation on page 49
The Puppet Application Manager (PAM) installation process creates a Kubernetes cluster for you and walks you through installing your Puppet application on the cluster.

Upgrade an automated online application installation on page 69
If you installed a Puppet application following the automated online installation instructions, run a script to upgrade to the latest version.

Automate PAM and Puppet application offline installations

During a fresh offline installation of Puppet Application Manager (PAM) and a Puppet application, you have the option to configure the software automatically rather than completing the installation script interview.

Before you begin
Ensure that your system meets the PAM system requirements on page 32.

Related information
PAM HA offline installation on page 53
Use these instructions to install Puppet Application Manager (PAM) in an air-gapped or offline environment where the Puppet Application Manager host server does not have direct access to the internet.

Upgrade an automated offline application installation on page 70
If you installed a Puppet application following the automated offline installation instructions, run a script to upgrade to the latest version.

Automate PAM and Puppet application offline installations on Puppet-supported clusters

1. Install Puppet Application Manager. For detailed instructions, see PAM HA offline installation on page 53.
2. Define the configuration values for your Puppet application installation, using Kubernetes YAML format.

```yaml
apiVersion: kots.io/v1beta1
kind: ConfigValues
metadata:
  name: app-config
spec:
  values:
    accept_eula:
      value: has_accepted_eula
  annotations:
    value: "ingress.kubernetes.io/force-ssl-redirect: 'false'"
hostname:
  value: "<HOSTNAME>"
root_password:
  value: "<ROOT ACCOUNT PASSWORD>"

**Tip:** View the keyword names for all settings by clicking View files > upstream > config.yaml in Puppet Application Manager.

Replace the values indicated:

- Replace `<HOSTNAME>` with a hostname you want to use to configure an Ingress and to tell job hardware agents and web hooks how to connect to it. You might need to configure your DNS to resolve the hostname to your Kubernetes hosts.
- Replace `<ROOT ACCOUNT PASSWORD>` your chosen password for the application root account. The root account is used to administer your application and has full access to all resources and application-wide settings. This account must NOT be used for testing and deploying control repositories or modules.
- **Optional.** These configuration values disable HTTP-to-HTTPS redirection, so that SSL can be terminated at the load balancer. If you want to run the application over SSL only, change the `force-ssl-redirect` annotation to `true`.
- **Optional.** If your load balancer requires HTTP health checks, you can now enable Ingress settings that do not require Server Name Indication (SNI) for `/status`. To enable this setting, add the following to the config values statement:

```
enable_lb_healthcheck:
  value: "1"
```

**Note:** The automated installation automatically accepts the Puppet application end user license agreement (EULA). Unless Puppet has otherwise agreed in writing, all software is subject to the terms and conditions of the Puppet Master License Agreement located at https://puppet.com/legal.

3. Write your license file and the configuration values generated in the previous step to the following locations:

- Write your license file to `./replicated_license.yaml`
- Write your configuration values to `./replicated_config.yaml`

4. Download the application bundle:

```
curl -L <APPLICATION BUNDLE URL> -o <APPLICATION BUNDLE FILE>
```

5. Copy the application bundle to your primary and secondary nodes and unpack it:

```
tar xzf ./<APPLICATION BUNDLE FILE>
```
6. Run the application install command on your primary node. Replace the `<YOUR CHOSEN PASSWORD>`, `<APPLICATION NAME>`, `<APPLICATION BUNDLE FILE>` values in the example below with your own values:

```
kubectl kots install <APPLICATION NAME> --namespace default --shared-password $KOTS_PASSWORD --license-file ./license.yaml --config-values ./config.yaml --airgap-bundle ./<APPLICATION BUNDLE FILE> --port-forward=false
```

# wait several minutes for the application to deploy; if it doesn’t show up, preflights or another error might have occurred

**Note:** If you want to install a specific version of the application, include the `--app-version-label=<VERSION>` flag in the install command.

### Automate PAM and Puppet application offline installations on customer-supported clusters

#### Before you begin

1. If you haven’t already done so, install `kubectl`.
2. Puppet Application Manager is expected to work on any certified Kubernetes distribution that meets the following requirements. We validated and support:
   - Google Kubernetes Engine
   - AWS Elastic Kubernetes Service
   - If you use a different distribution, contact Puppet Support for more information on compatibility with PAM.
3. Make sure your Kubernetes cluster meets the minimum requirements:
   - Kubernetes version 1.19-1.23.
   - A default storage class that can be used for relocatable storage.
   - A standard Ingress controller that supports websockets (we have tested with Project Contour and NGINX).
   - We currently test and support Google Kubernetes Engine (GKE) clusters.

   **Note:** If you’re using self-signed certificates on your Ingress controller, you must ensure that your job hardware nodes trust the certificates. Additionally, all nodes that use Continuous Delivery for PE webhooks must trust the certificates, or SSL checking must be disabled on these nodes.

#### Important: If you are installing Puppet Comply on Puppet Application Manager, the ingress controller must be configured to allow request payloads of up to 32 MB. Ingress controllers used by Amazon EKS commonly default to a 1 MB maximum — this causes all report submissions to fail.

The ingress must have a generous limit for total connection time. Setting the connection timeout to `infinity` in conjunction with an idle timeout is recommended.

4. If you are setting up Puppet Application Manager behind a proxy server, the installer supports proxies configured via `HTTP_PROXY/HTTPS_PROXY/NO_PROXY` environment variables.

**Restriction:** Using a proxy to connect to external version control systems is currently not supported.
1. Define the configuration values for your Puppet application installation, using Kubernetes YAML format.

```yaml
apiVersion: kots.io/v1beta1
kind: ConfigValues
metadata:
  name: app-config
spec:
  values:
    accept_eula:
      value: has_accepted_eula
  annotations:
    value: "ingress.kubernetes.io/force-ssl-redirect: 'false'"
  hostname:
    value: "<HOSTNAME>"
  root_password:
    value: "<ROOT ACCOUNT PASSWORD>"
```

**Tip:** View the keyword names for all settings by clicking **View files > upstream > config.yaml** in Puppet Application Manager.

Replace the values indicated:

- Replace `<HOSTNAME>` with a hostname you want to use to configure an Ingress and to tell job hardware agents and web hooks how to connect to it. You might need to configure your DNS to resolve the hostname to your Kubernetes hosts.
- Replace `<ROOT ACCOUNT PASSWORD>` your chosen password for the application root account. The root account is used to administer your application and has full access to all resources and application-wide settings. This account must NOT be used for testing and deploying control repositories or modules.
- **Optional.** These configuration values disable HTTP-to-HTTPS redirection, so that SSL can be terminated at the load balancer. If you want to run the application over SSL only, change the `force-ssl-redirect` annotation to true.
- **Optional.** If your load balancer requires HTTP health checks, you can now enable Ingress settings that do not require Server Name Indication (SNI) for `/status`. To enable this setting, add the following to the config values statement:

  ```yaml
  enable_lb_healthcheck:
    value: "1"
  ```

**Note:** The automated installation automatically accepts the Puppet application end user license agreement (EULA). Unless Puppet has otherwise agreed in writing, all software is subject to the terms and conditions of the Puppet Master License Agreement located at [https://puppet.com/legal](https://puppet.com/legal).

2. Write your license file and the configuration values generated in the previous step to the following locations:

- Write your license file to `./replicated_license.yaml`
- Write your configuration values to `./replicated_config.yaml`

3. Download the application bundle:

```bash
curl -L <APPLICATION BUNDLE URL> -o <APPLICATION BUNDLE FILE>
```
4. Create and run the following script, supplying values specific to your installation for the variables:

```bash
#!/bin/bash
REGISTRY=<YOUR_CONTAINER_REGISTRY>
APP_K8S_NAMESPACE=<DESIRED_NAMESPACE_IN_TARGET_CLUSTER>
APP_BUNDLE=<PATH_TO_AIRGAP_BUNDLE_FROM_STEP_3>
PAM_PASSWORD=<DESIRED_PAM_CONSOLE_PASSWORD>
LICENSE_FILE=<PATH_TO_LICENSE_FILE_FROM_STEP_1>
CONFIG_FILE=<PATH_TO_CONFIG_FILE_FROM_STEP_2>
curl https://kots.io/install | bash
curl -LO https://github.com/replicatedhq/kots/releases/download/v$(kubectl kots version | head -n1 | cut -d' ' -f3)/kotsadm.tar.gz
kubectl kots admin-console push-images ./kotsadm.tar.gz ${REGISTRY}
kubectl kots admin-console push-images ${APP_BUNDLE} ${REGISTRY}
kubectl kots install puppet-application-manager --namespace ${APP_K8S_NAMESPACE} --shared-password ${PAM_PASSWORD} --license-file ${LICENSE_FILE} --config-values ${CONFIG_FILE} --airgap-bundle ${APP_BUNDLE} --disable-image-push --kotsadm-registry ${REGISTRY} --port-forward=false --skip-preflights
```

**Tip:** If the script fails, it might be because:
- The `push-images` commands require that the local machine where the script is running has push access to the registry.
- The `install` command requires read access to the registry from the target cluster.
- Offline HA installs of GKE can't run preflights; therefore `--skip-preflights` must be included.

### Uninstall PAM

Different uninstall procedures are required for Puppet-supported and customer-supported clusters.

At this time it's not possible to cleanly uninstall PAM from Puppet-supported clusters.

If you need to start with a fresh PAM install, you'll need to provision a new host.

#### Uninstall PAM on customer-supported clusters

To uninstall Puppet Application Manager from customer-supported clusters, use:

```bash
kubectl delete namespace <pam-namespace>
kubectl delete clusterrolebinding kotsadm-rolebinding
kubectl delete clusterrole kotsadm-role
```

### Related information

**Using sudo behind a proxy server** on page 92

Many of the commands you run to install or configure Puppet Application Manager (PAM) require root access. In the PAM documentation, commands that require root access use `sudo` to elevate privileges. If you're running PAM behind a proxy, `sudo` might not work correctly. If you're having trouble running commands with `sudo`, and you're behind a proxy, try switching to the `root` user and running the command without `sudo`.

### Working with Puppet applications

You can install and upgrade Puppet applications using the Puppet Application Manager UI.

© 2022 Puppet, Inc., a Perforce company
• Install applications via the PAM UI on page 68
  The process of adding an application once you've installed Puppet Application Manager is simple.
• Update a license for online installations on page 69
  If you have performed online installation of an application, you can use the Puppet Application Manager UI to update your license.
• Update a license for offline installations on page 69
  If you have performed offline installation of an application, you can use the Puppet Application Manager UI to update your license.
• Upgrade an automated online application installation on page 69
  If you installed a Puppet application following the automated online installation instructions, run a script to upgrade to the latest version.
• Upgrade an automated offline application installation on page 70
  If you installed a Puppet application following the automated offline installation instructions, run a script to upgrade to the latest version.

Install applications via the PAM UI

The process of adding an application once you've installed Puppet Application Manager is simple.

<table>
<thead>
<tr>
<th>Application</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Delivery for Puppet Enterprise</td>
<td>4.6.0 or later</td>
</tr>
<tr>
<td>Comply</td>
<td>1.0.4 or later</td>
</tr>
</tbody>
</table>

For information on installing Puppet applications via the command line, see Automate PAM and Puppet application online installations on page 61 and Automate PAM and Puppet application offline installations on page 63.

To install a Puppet application using the Puppet Application Manager UI:

1. Log into the Puppet Application Manager UI, and click **Add a new application**.
   - If you have not added a Puppet application before you are prompted to upload a license.
   - If you have already added a Puppet application, click **Add a new application**.
2. Upload your `replicated_license.yaml` file when requested.

**Note:** Once the license file is installed, if offline installations are enabled, you are presented with an option to proceed with an offline setup.

Add the following information to install an offline application:

• **Hostname** - the hostname you want to use to configure an Ingress and to tell job hardware agents and web hooks how to connect to it. You might need to configure your DNS to resolve the hostname to your Kubernetes hosts.

  **Important:** The hostname must be unique for each application you install.

• **Username/Password** - The username and password for the application root account. The root account is used to administer your application and has full access to all resources and application-wide settings. This account must NOT be used for testing and deploying control repositories or modules.

• **Registry namespace** - the registry namespace for the application, e.g. **CD4PE** or **Comply**.

• **Airgap bundle** - upload the relevant application bundle tarball. Click **Continue**.
3. Add any additional required information that is presented on the **Config** page. Configure any other settings on the page relevant to your installation, such as external databases, customized endpoints, a load balancer, or TLS certificates. Click **Save Config** when you are done.

   Saving your new configuration settings prompts the creation of a new application version.

4. Click **Go to new version**, which redirects you to the **Version history** tab. The newly created version is shown in the **All versions** section of the page.

5. Monitor the new version’s preflight checks. The **Running Checks** indicator is shown on the screen while your system is checked to make sure your cluster meets minimum system requirements. When the preflight check is complete:

   • If the status is **Checks Failed**, click **View preflights**. Correct the issues and click **Re-run**. Repeat this step as needed.

   | **Important:** Do not move on until all preflight checks pass. |

   • If the status is **Ready to Deploy**, move on to the next step.

6. Once the version is ready to deploy, click **Deploy**. On the **Application** tab, monitor the application for readiness.

   The application’s status is shown as **Missing** for several minutes while deployment is underway. To monitor the deployment’s progress, run `kubectl get pods --watch`.

   When the deployment is complete, the application status changes to **Ready**.

7. Navigate to `https://<HOSTNAME>` (using the hostname you entered on the **Config** screen) and sign into your application.

**Update a license for online installations**

If you have performed online installation of an application, you can use the Puppet Application Manager UI to update your license.

To update the license for an online application:

1. Log in to Puppet Application Manager, click the **License** tab, and then **Sync License**.
2. On the **Version history** tab, click **Deploy**.

   Puppet Application Manager adds “License Change” as the deployment cause on the **Version history** tab.

**Update a license for offline installations**

If you have performed offline installation of an application, you can use the Puppet Application Manager UI to update your license.

To update the license for an offline application:

1. Ask your Puppet sales representative to email you an updated license file.
2. Log in to Puppet Application Manager, click the **License** tab.
3. Drag and drop or upload the updated license file provided by your Puppet sales representative.
4. On the **Version history** tab, click **Deploy**.

   Puppet Application Manager adds “License Change” as the deployment cause on the **Version history** tab.

**Upgrade an automated online application installation**

If you installed a Puppet application following the automated online installation instructions, run a script to upgrade to the latest version.

| **Important:** Ensure that you are following an approved upgrade path for the application you want to upgrade. For more information, check the relevant application documentation. |
1. From the command line of your primary (control plane) node, get the application slug for the application you want to upgrade:

   ```bash
   kubectl kots --namespace <NAMESPACE> get apps
   ```

   Replace `<NAMESPACE>` with the name of the namespace in which you installed PAM (usually `default`).

2. Run the upgrade script:

   ```bash
   kubectl kots upstream upgrade <APPLICATION SLUG> --namespace <NAMESPACE> --deploy
   ```

   Replace `<APPLICATION SLUG>` with the relevant application slug for the application you want to upgrade.
   Replace `<NAMESPACE>` with the name of the namespace in which you installed PAM (usually `default`).

3. Wait five minutes to allow the software time to process the change.


   If preflight checks have passed, the upgraded application is deployed and in the process of starting up. To monitor the deployment's progress, run:

   ```bash
   kubectl get pods --watch
   ```

**Related information**

- Automate PAM and Puppet application offline installations on page 63

During a fresh offline installation of Puppet Application Manager (PAM) and a Puppet application, you have the option to configure the software automatically rather than completing the installation script interview.

**Upgrade an automated offline application installation**

If you installed a Puppet application following the automated offline installation instructions, run a script to upgrade to the latest version.

**Important:** Ensure that you are following an approved upgrade path for the application you want to upgrade. For more information, check the relevant application documentation.

1. Download the application bundle you want to upgrade to. Copy to your primary node.

2. From the command line of your primary (control plane) node, get the application slug for the application you want to upgrade:

   ```bash
   kubectl kots --namespace <NAMESPACE> get apps
   ```

   Replace `<NAMESPACE>` with the name of the namespace in which you installed PAM (usually `default`).

3. Run the upgrade script:

   ```bash
   kubectl kots upstream upgrade <APPLICATION SLUG> --airgap-bundle ./<APPLICATION BUNDLE_FILE> --kotsadm-namespace <REGISTRY NAMESPACE> --namespace <NAMESPACE> --deploy
   ```

   - Replace `<APPLICATION SLUG>` with the relevant application slug for the application you want to upgrade.
   - Replace `<APPLICATION BUNDLE_FILE>` with the name of the application bundle file.
   - Replace `<REGISTRY NAMESPACE>` with your Registry namespace where images are uploaded.
   - Replace `<NAMESPACE>` with the name of the namespace in which you installed PAM (usually `default`).

4. Wait five minutes to allow the software time to process the change.
5. Navigate to `http://<NODE IP ADDRESS>:8800` and log in with the Puppet Application Manager password.

If preflight checks have passed, the upgraded application is deployed and in the process of starting up. To monitor the deployment's progress, run:

```
kubectl get pods --watch
```

**Related information**

Automate PAM and Puppet application offline installations on page 63

During a fresh offline installation of Puppet Application Manager (PAM) and a Puppet application, you have the option to configure the software automatically rather than completing the installation script interview.

---

**Maintenance and tuning**

Follow these guidelines when you're tuning or performing maintenance on a node running Puppet Application Manager (PAM).

**How to look up your Puppet Application Manager architecture**

If you're running PAM on a Puppet-supported cluster, you can use the following command to determine your PAM architecture version:

```
kubectl get installer --sort-by=.metadata.creationTimestamp -o jsonpath='{.items[-1:].metadata.name}' ; echo
```

Depending on which architecture you used when installing, the command returns one of these values:

- **HA architecture**: `puppet-application-manager`
- **Standalone architecture**: `puppet-application-manager-standalone`
- **Legacy architecture**: Any other value, for example, `puppet-application-manager-legacy`, `cd4pe`, or `comply`

**Rebooting PAM nodes**

Where possible, avoid rebooting or shutting down a PAM node. Shutting down an HA PAM node incorrectly could result in storage volume corruption and the loss of data.

For tasks such as package updates or security patches, where you must perform a reboot or shut down, follow the procedure below to gracefully shut down the node and ensure that it is drained correctly.

To reboot a node:

1. Shut down services using Ceph-backed storage:

   ```
   /opt/ekco/shutdown.sh
   ```

2. If you're using a high availability (HA) cluster, drain the node:

   ```
   kubectl drain <NODE NAME> --ignore-daemonsets --delete-local-data
   ```

3. Reboot the node.

**Load balancer health checks**

To set up health checks for the load balancer that your Puppet Application Manager (PAM) applications are running behind, set up rules for these applications and services.
<table>
<thead>
<tr>
<th>Application/service</th>
<th>URL/port</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Puppet application. For example, Continuous Delivery for Puppet Enterprise or Puppet Comply</td>
<td>https://&lt;CDPE HOSTNAME&gt;:443/status</td>
<td>Although Puppet applications might expose other ports (Continuous Delivery for PE exposes ports 443, 80, and 8000), 443 is the HTTPS endpoint, and is the best port to use for health checks.</td>
</tr>
<tr>
<td>Puppet Application Manager (PAM)</td>
<td>https://&lt;KUBERNETES PRIMARY IP&gt;:8800/healthz</td>
<td></td>
</tr>
<tr>
<td>External load balancer endpoint</td>
<td>Port 6443 or https://&lt;KUBERNETES PRIMARY IP&gt;:6443/livez</td>
<td>For information on setting up a TCP probe on an external load balancer endpoint, consult the kURL load balancer documentation.</td>
</tr>
<tr>
<td>Local container registry (for offline installations)</td>
<td>https://&lt;KUBERNETES PRIMARY IP&gt;:9001</td>
<td></td>
</tr>
</tbody>
</table>

### Upgrading PAM on a Puppet-supported cluster

Upgrade Puppet Application Manager (PAM) on a Puppet-supported cluster to take advantage of new features and bug fixes, and to upgrade your cluster to the latest version of Kubernetes when one is available.

There are four possible upgrade types for Puppet Application Manager installations:

- **Online** - For standalone or HA installations with a connection to the internet.
- **Offline** - For air-gapped standalone or HA installations without a connection to the internet.
- **Online legacy** - For standalone or HA installations created prior to April 2021 with a connection to the internet.
- **Offline legacy** - For air-gapped standalone or HA installations created prior to April 2021 without a connection to the internet.

**Restriction:** You cannot use the upgrade process to move from a legacy deployment to a non-legacy deployment, or from standalone to HA, or vice versa. If you wish to change architecture types, see Migrating PAM data to a new system on page 80.

#### How to look up your Puppet Application Manager architecture

If you're running PAM on a Puppet-supported cluster, you can use the following command to determine your PAM architecture version:

```
kubectl get installer --sort-by=.metadata.creationTimestamp -o jsonpath='{.items[-1:].metadata.name}' ; echo
```

Depending on which architecture you used when installing, the command returns one of these values:

- **HA architecture**: puppet-application-manager
- **Standalone architecture**: puppet-application-manager-standalone
- **Legacy architecture**: Any other value, for example, puppet-application-manager-legacy, cd4pe, or comply

#### Upgrade PAM online

Upgrade Puppet Application Manager (PAM) to take advantage of new features and bug fixes, and to upgrade your cluster to the latest version of Kubernetes when one is available.
Before you begin
Make sure you have captured an up-to-date snapshot of your PAM installation, which you can use to fall back the current version if there is an issue with the upgrade process. Learn more about snapshots at Backing up PAM using snapshots on page 77.

1. On your first primary node, rerun the installation script, passing in any arguments you included when installing for the first time:
   For standalone deployments, use:
   ```bash
   curl -sSL https://k8s.kurl.sh/puppet-application-manager-standalone | sudo bash -s force-reapply-addons
   ```
   For HA deployments, use:
   ```bash
   curl -sSL https://k8s.kurl.sh/puppet-application-manager | sudo bash -s force-reapply-addons
   ```

2. If a new version of Kubernetes is available, the installer notes upgrade scripts to run on other nodes in an HA cluster.
The installer also pauses before draining nodes as part of the Kubernetes upgrade. The node draining process can take several minutes to complete, during which time application workloads are stopped or migrated to other systems. This migration may cause several minutes of downtime while databases are rescheduled.

Related information
Using sudo behind a proxy server on page 92
Many of the commands you run to install or configure Puppet Application Manager (PAM) require root access. In the PAM documentation, commands that require root access use `sudo` to elevate privileges. If you're running PAM behind a proxy, `sudo` might not work correctly. If you're having trouble running commands with `sudo`, and you're behind a proxy, try switching to the `root` user and running the command without `sudo`.

Upgrade PAM offline
Users operating in environments without direct access to the internet must use the links below to upgrade to the latest version of Puppet Application Manager (PAM).

Before you begin
Make sure you have captured an up-to-date snapshot of your PAM installation, which you can use to fall back the current version if there is an issue with the upgrade process. Learn more about snapshots at Backing up PAM using snapshots on page 77.

To upgrade Puppet Application Manager:
1. From a workstation with internet access, download the latest version of the installation bundle that is relevant for your installation type:
   For standalone installations, enter the following command (note that this bundle is ~4GB):
   ```bash
   curl -LO https://k8s.kurl.sh/bundle/puppet-application-manager-standalone.tar.gz
   ```
   For HA installations, enter the following command (note that this bundle is ~4GB):
   ```bash
   curl -LO https://k8s.kurl.sh/bundle/puppet-application-manager.tar.gz
   ```
2. Copy the installation bundle to your primary and secondary nodes and unpack it:

   For standalone installations, use:
   ```
tar xzf puppet-application-manager-standalone.tar.gz
   ```

   For HA installations, use:
   ```
tar xzf puppet-application-manager.tar.gz
   ```

3. On your primary node, rerun the installation script, passing in any arguments you included when installing for the first time:

   ```
cat install.sh | sudo bash -s airgap force-reapply-addons
   ```

   **Note:** This script issues a prompt to run the `task.sh` and `upgrade.sh` scripts on your secondary nodes. Use the versions of these scripts from the downloaded bundle in step 2.

4. If a new version of Kubernetes is available, the installer systems provide upgrade scripts to run on other nodes in an HA cluster. The installer also pauses before draining nodes as part of the Kubernetes upgrade. Node draining is performed as part of a Kubernetes upgrade.

   The node draining process can take several minutes to complete, during which time application workloads are stopped or migrated to other systems. This migration may cause several minutes of downtime while databases are rescheduled.

   When the deployment is complete, sign into Puppet Application Manager - [http://<PUPPET APPLICATION MANAGER ADDRESS>:8800](http://<PUPPET APPLICATION MANAGER ADDRESS>:8800) - and verify that the new version number is displayed in the bottom left corner of the web UI.

**PAM legacy upgrades**

The legacy architecture is no longer supported. However, if you have not yet migrated to a supported architecture, you can use this method to upgrade Puppet Application Manager (PAM).

**Before you begin**

Make sure you have captured an up-to-date snapshot of your PAM installation, which you can use to fall back the current version if there is an issue with the upgrade process. Learn more about snapshots at [Backing up PAM using snapshots](#) on page 77.

**Legacy architecture is no longer supported:** The legacy architecture utilizes Rook 1.0, which is incompatible with Kubernetes version 1.20 and newer versions. Kubernetes version 1.19 is no longer receiving security updates. The legacy architecture reached the end of its support lifecycle on 30 June 2022, and Puppet no longer updates legacy architecture components. For information on migrating data from a legacy architecture to a standalone or HA architecture, go to our Support Knowledge Base instructions:

- Migrate to a supported PAM architecture for Continuous Delivery for PE
- Migrate to a supported PAM architecture for Comply

**Restriction:** It is not possible to upgrade from an online legacy install to a new offline install configuration. Similarly, upgrades from an offline legacy configuration to a new online install are not supported.

To upgrade a legacy version of Puppet Application Manager on nodes with internet access:
1. On your node (or control plane node if you have a HA deployment), rerun the installation script, passing in any arguments you included when installing for the first time:

   • For standalone installs:
     ```bash
     curl -sSL https://k8s.kurl.sh/puppet-application-manager-legacy | sudo bash -s force-reapply-addons
     ```

   • For HA installs:
     ```bash
     curl -sSL https://k8s.kurl.sh/puppet-application-manager-legacy | sudo bash -s ha force-reapply-addons
     ```

2. If a new version of Kubernetes is available, the systems provide upgrade scripts to run on each node in your cluster.

   Node draining is performed as part of a Kubernetes upgrade. The node draining process can take several minutes to complete.

   **Note:** During the Kubernetes upgrade process, nodes are not able to properly route network connections. If you have a HA deployment, make sure you have load balancers or a multi-node fail-over process in place, or schedule downtime before upgrading.

---

**PAM offline legacy upgrades**

The legacy architecture is no longer supported. However, if you have not yet migrated to a supported architecture, you can use this method to upgrade Puppet Application Manager (PAM) on offline nodes.

**Before you begin**

Make sure you have captured an up-to-date snapshot of your PAM installation, which you can use to fall back the current version if there is an issue with the upgrade process. Learn more about snapshots at [Backing up PAM using snapshots](#) on page 77.

**Legacy architecture is no longer supported:** The legacy architecture utilizes Rook 1.0, which is incompatible with Kubernetes version 1.20 and newer versions. Kubernetes version 1.19 is no longer receiving security updates. The legacy architecture reached the end of its support lifecycle on **30 June 2022**, and Puppet no longer updates legacy architecture components. For information on migrating data from a legacy architecture to a standalone or HA architecture, go to our Support Knowledge Base instructions:

   • Migrate to a supported PAM architecture for Continuous Delivery for PE
   • Migrate to a supported PAM architecture for Comply

**Restriction:** It is not possible to upgrade from an online legacy install to a new offline install configuration. Similarly, upgrades from an offline legacy configuration to a new online install are not supported.

To upgrade Puppet Application Manager on nodes without a connection to the internet:

1. From a workstation with internet access, download the latest version of the cluster installation bundle (note that this bundle is ~4GB):

   ```bash
   https://k8s.kurl.sh/bundle/puppet-application-manager-legacy.tar.gz
   ```

2. Copy the installation bundle to your primary and secondary Puppet Application Manager nodes and unpack it:

   ```bash
   tar xzf puppet-application-manager-legacy.tar.gz
   ```
3. Rerun the installation script. Don't forget to pass in any additional arguments you included when installing for the first time you installed the product:

For standalone installs use:

```
cat install.sh | sudo bash -s airgap force-reapply-addons
```

For HA installs use:

```
cat install.sh | sudo bash -s airgap ha force-reapply-addons
```

| Note: During the upgrade process, follow any prompts to run commands on your other cluster nodes. |

When the deployment is complete, sign into Puppet Application Manager and verify that the new version number is displayed in the bottom center of the web UI.

---

**Upgrading PAM on a customer-supported cluster**

Upgrade Puppet Application Manager (PAM) on your own Kubernetes cluster to take advantage of new features and bug fixes.

There are two possible upgrade types for customer-supported Puppet Application Manager deployments:

- **Online** - For installations with a connection to the internet.
- **Offline** - For air-gapped installations without a connection to the internet.

---

**Upgrade PAM on a customer-supported online cluster**

Upgrading Puppet Application Manager (PAM) on a customer-supported online Kubernetes cluster can be done with a single command.

**Before you begin**

Make sure you have captured an up-to-date snapshot of your PAM installation, which you can use to fall back the current version if there is an issue with the upgrade process. Learn more about snapshots at Backing up PAM using snapshots on page 77.

To upgrade Puppet Application Manager on a customer-supported online cluster:

1. Upgrade kubectl KOTS:

```
curl https://kots.io/install | bash
```

2. Issue the following KOTS command:

```
kubectl kots admin-console upgrade --namespace <target namespace>
```

| Tip: Run the `kubectl kots admin-console upgrade -h` command for more usage information. |

---

**Upgrade PAM on a customer-supported offline cluster**

Upgrading Puppet Application Manager (PAM) on a customer-supported offline Kubernetes cluster requires a few simple kubectl commands.

**Before you begin**

Make sure you have captured an up-to-date snapshot of your PAM installation, which you can use to fall back the current version if there is an issue with the upgrade process. Learn more about snapshots at Backing up PAM using snapshots on page 77.
To upgrade Puppet Application Manager on a customer-supported offline cluster, perform the following steps from a workstation that has kubectl access to the cluster:

1. Upgrade kubectl KOTS:
   ```bash
curl https://kots.io/install | bash
   ```

2. Ensure the required images are available in your local registry. Download the release assets matching the CLI version using the following command:
   ```bash
curl -LO https://github.com/replicatedhq/kots/releases/download/v$(kubectl kots version | head -n1 | cut -d' ' -f3)/kotsadm.tar.gz
   ```

3. Extract the images and push them to your private registry. Registry credentials provided in this step must have push access. These credentials are not stored anywhere or reused later.
   ```bash
   kubectl kots admin-console push-images ./kotsadm.tar.gz
   <private.registry.host>/puppet-application-manager \
   --registry-username <rw-username> \
   --registry-password <rw-password>
   ```

4. After you push the images to your private registry, execute the upgrade command with registry read-only credentials:
   ```bash
   kubectl kots upgrade puppet-application-manager \
   --kotsadm-namespace puppet-application-manager \
   --kotsadm-registry <private.registry.host> \
   --registry-username <ro-username> \
   --registry-password <ro-password> \
   --namespace <target namespace>
   ```

---

### Backing up PAM using snapshots

Snapshots are point-in-time backups of your Puppet Application Manager (PAM) deployment, which can be used to roll back to a previous state or restore your installation into a new cluster for disaster recovery.

**Related information**

**Using sudo behind a proxy server** on page 92

Many of the commands you run to install or configure Puppet Application Manager (PAM) require root access. In the PAM documentation, commands that require root access use `sudo` to elevate privileges. If you're running PAM behind a proxy, `sudo` might not work correctly. If you're having trouble running commands with `sudo`, and you're behind a proxy, try switching to the root user and running the command without `sudo`.

### Full and partial snapshots

There are two options available when you're creating a snapshot for your Puppet Application Manager (PAM) deployment, full snapshots (also known as instance snapshots) and partial (or application) snapshots. For full disaster recovery, make sure you've configured and scheduled regular full snapshots stored on a remote storage solution such as an S3 bucket or NFS share.

Full snapshots offer a comprehensive backup of your PAM deployment, because they include the core PAM application together with the Puppet applications you've installed in your PAM deployment. You can use a full snapshot to restore your PAM deployment and all of your installed Puppet applications to a previous backup. For example, you could use a full snapshot to revert an undesired configuration change or a failed upgrade, or to migrate your PAM deployment to another Puppet-supported cluster.

Partial snapshots are available from the PAM console, but are limited in their usefulness. To restore from a partial snapshot, you must already have an installed and functioning version of PAM. A functioning PAM installation is
needed because the option to restore a partial snapshot can only be accessed from the Snapshots section of the PAM admin console.

Partial snapshots only back up the Puppet application you specified when you configured the snapshot, for example, Continuous Delivery for Puppet Enterprise, or Puppet Comply. They do not back up the underlying PAM deployment. Partial snapshots are sometimes useful if you want to roll back to a previous version of a specific Puppet application that you've installed on your PAM deployment, but are far less versatile than full snapshots. To make sure that you have all disaster recovery options available to you, use a full snapshot wherever possible.

**Configure snapshots**

Before using snapshots, select a storage location, set a snapshot retention period, and indicate whether snapshots are created manually or on a set schedule.

**Important:** Disaster recovery requires that the store backend used for backups is accessible from the new cluster. When setting up snapshots in an offline cluster, make sure to record the registry service IP address with the following command:

```
kubectl -n kurl get svc registry -o jsonpath='{.spec.clusterIP}'
```

Be sure to record the value returned by this command as it is required when creating a new cluster to restore to as part of Disaster recovery with PAM on page 86.

1. In the upper navigation bar of the Puppet Application Manager UI, click Snapshots > Settings & Schedule.
2. The snapshots feature uses https://velero.io, an open source backup and restore tool. Click Check for Velero to determine whether Velero is present on your cluster, and to install it if needed.
3. Select a destination for your snapshot storage and provide the required configuration information. You can choose to set up snapshot storage in the PAM UI or on the command line. Supported destinations are listed below. We recommend using an external service or NFS, depending on what is available to you:

- Internal storage (default)
- Amazon S3
- Azure Blob Storage
- Google Cloud Storage
- Other S3-compatible storage
- Network file system (NFS)
- Host path

**Amazon S3 storage**

If using the PAM UI, provide the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>The name of the AWS bucket where snapshots are stored.</td>
</tr>
<tr>
<td>Region</td>
<td>The AWS region the bucket is available in.</td>
</tr>
<tr>
<td>Path</td>
<td>Optional. The path within the bucket where all snapshots are stored.</td>
</tr>
<tr>
<td>Use IAM instance role?</td>
<td>If selected, an IAM instance role is used instead of an access key ID and secret.</td>
</tr>
<tr>
<td>Access key ID</td>
<td>Required only if not using an IAM instance role. The AWS IAM access key ID that can read from and write to the bucket.</td>
</tr>
<tr>
<td>Access key secret</td>
<td>Required only if not using an IAM instance role. The AWS IAM secret access key that is associated with the access key ID.</td>
</tr>
</tbody>
</table>

If using the command line, run the appropriate command:

**Not using an IAM instance role:**

```bash
kubectl kots velero configure-aws-s3 access-key --access-key-id <string> --bucket <string> --path <string> --region <string> --secret-access-key <string>
```

**Using an IAM instance role:**

```bash
kubectl kots velero configure-aws-s3 instance-role --bucket <string> --path <string> --region <string>
```

**Azure Blob Storage**

If using the PAM UI, provide the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bucket</td>
<td>The name of the Azure Blob Storage container where snapshots are stored.</td>
</tr>
<tr>
<td>Path</td>
<td>Optional. The path within the container where all snapshots are stored.</td>
</tr>
<tr>
<td>Subscription ID</td>
<td>Required only for access via service principal or AAD Pod Identity.</td>
</tr>
<tr>
<td></td>
<td>The subscription ID associated with the target container.</td>
</tr>
<tr>
<td>Tenant ID</td>
<td>Required only for access via service principal. The tenant ID associated with the Azure account of the target container.</td>
</tr>
<tr>
<td>Client ID</td>
<td>Required only for access via service principal. The client ID of a Service Principle with access to the target container.</td>
</tr>
<tr>
<td>Client secret</td>
<td>Required only for access via service principal. The Client Secret of a Service Principle with access to the target container.</td>
</tr>
</tbody>
</table>

**Note:** Only connections via service principals are currently supported.
4. Click **Update storage settings** to save your storage destination information. Depending on your chosen storage provider, saving and configuring your storage provider might take several minutes.

5. Optional: To automatically create new snapshots on a schedule, select **Enable automatic scheduled snapshots** on the **Full snapshots (instance)** tab. (If desired, you can also set up a schedule for capturing partial (application-only) snapshots.)

   You can schedule a new snapshot creation for every hour, day, or week, or you can create a custom schedule by entering a cron expression.

6. Finally, set the retention schedule for your snapshots by selecting the time period after which old snapshots are automatically deleted. The default retention period is one month.

   **Note:** A snapshot's retention period cannot be changed once the snapshot is created. If you update the retention schedule, the new retention period applies only to snapshots created after the update is made.

7. Click **Update schedule** to save your changes.

Snapshots are automatically created according to your specified schedule and saved to the storage location you selected. You can also create an unscheduled snapshot at any time by clicking **Start a snapshot** on the **Dashboard** or on the **Snapshots** page.

**Roll back changes using a snapshot**

When necessary, you can use a snapshot to roll back to a previous version of your Puppet Application Manager set-up without changing the underlying cluster infrastructure.

To roll back changes:

1. In console menu of the Puppet Application Manager UI, click **Snapshots** > **Full Snapshots (Instance)**.

2. Select the snapshot you wish to roll back to from the list of available snapshots and click **Restore from this backup**.

3. Follow the instructions to complete either a partial restore or a full restore.

   A full restore is useful if you need to stay on an earlier version of an application and want to disable automatic version updates. Otherwise, a partial restore is the quicker option.

**Migrating PAM data to a new system**

By using a snapshot, you can migrate your data to a new Puppet Application Manager (PAM) instance.

**Data migration prerequisites**

In order to perform a data migration, your system must be configured as follows:

- On the original system, Puppet Application Manager (PAM) must be configured to support **Full Snapshots (Instance)**. For instructions on configuring snapshots, see **Backing up PAM using snapshots** on page 77.

- Velero must be configured to use an external snapshot destination accessible to both the old and new clusters, such as S3 or NFS.

- Both the old and new clusters must have the same connection status (online or offline). Migrating from offline to online clusters or vice versa is not supported.

- For offline installs, both the old and new clusters must use the same version of PAM.

- Upgrade to the latest version of PAM on both the old and new clusters before you begin.

**Migrating data between two systems with the same architecture**

To perform data migration between two systems using the same architecture (from standalone to standalone, or from HA to HA), you must create a new cluster to migrate to, then follow the process outlined below to recover your instance from a snapshot.
Before you begin

Review the requirements in Data migration prerequisites on page 80.

**Important:** If you are migrating from a legacy architecture, go to our Support Knowledge Base instructions for migrating to a supported architecture for your Puppet application:

- Migrate to a supported PAM architecture for Continuous Delivery for PE
- Migrate to a supported PAM architecture for Comply

1. On the original system, find the version of kURL your deployment is using by running the following command. Save the version for use in step 3.

```bash
kubectl get configmap -n kurl kurl-current-config -o jsonpath="{.data.kurl-version}" && echo
```
2. Get the installer `spec` section by running the command appropriate to your PAM installation type:

Tip: See How to determine your version of Puppet Application Manager if you’re not sure which installation type you’re running.

- **HA installation**: `kubectl get installers puppet-application-manager -o yaml`
- **Standalone installation**: `kubectl get installers puppet-application-manager-standalone -o yaml`
- **Legacy installation**: `kubectl get installers puppet-application-manager-legacy -o yaml`

The command's output looks similar to the following. The `spec` section is shown in bold in the example below. Save your `spec` section for use in step 3.

```
# kubectl get installers puppet-application-manager-standalone -o yaml
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"cluster.kurl.sh/v1beta1","kind":"Installer","metadata":
        {" annotations":{},"creationTimestamp":null,"name":"puppet-application-manager-standalone","namespace":"default"},"spec":{"containerd":
          {"version":"1.4.12"},"contour":{"version":"1.18.0"},"ekco":
          {"version":"0.16.0"},"kotsadm":{"applicationSlug":"puppet-application-manager","version":"1.64.0"},"kubernetes":
          {"version":"1.21.8"},"metricsServer":{"version":"0.4.1"},"minio":
          {"version":"2020-01-25T02-50-51Z"},"openebs":
          {"isLocalPVEnabled":true,"localPVStorageClassName":"default","version":"2.6.0"},"prometheus":
          {"version":"0.49.0-17.1.1"},"registry":{"version":1.6.2"},"weave":
          {"podCidrRange":"/22"},{"version":"2.8.1"}},"status":{}}
creationTimestamp: "2021-06-04T00:05:08Z"
generation: 4
labels:
  velero.io/exclude-from-backup: "true"
name: puppet-application-manager-standalone
namespace: default
resourceVersion: "102061068"
uid: 4e7f1196-5fab-4072-9399-15d18dcc5137
spec:
  containerd:
    version: 1.4.12
  contour:
    version: 1.18.0
  ekco:
    version: 0.16.0
  kotsadm:
    applicationSlug: puppet-application-manager
    version: 1.64.0
  kubernetes:
    version: 1.21.8
  metricsServer:
    version: 0.4.1
  minio:
    version: 2020-01-25T02-50-51Z
  openebs:
    isLocalPVEnabled: true
    localPVStorageClassName: default
    version: 2.6.0
  prometheus:
    version: 0.49.0-17.1.1
  registry:
    version: 2.7.1
  velero:
    version: 1.6.2

© 2022 Puppet, Inc., a Perforce company
3. On a new machine, create a file named `installer.yaml` with the following contents, replacing `<SPEC>` and `<KURL VERSION>` with the information you gathered in the previous steps.

```yaml
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
metadata:
  <SPEC>
  kurl:
    installerVersion: "<KURL VERSION>"
```

**Important:** If you are running PAM version 1.68.0 or newer, the kURL installer version might be included in the `spec` section. If this is the case, omit the `kurl:` section from the bottom of the `installer.yaml` file. There must be only one `kurl:` entry in the file.

**Tip:** Spacing is critical in YAML files. Use a YAML file linter to confirm that the format of your file is correct.

Here is an example of the contents of the `installer.yaml` file:

```yaml
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
metadata:
  spec:
    containerd:
      version: 1.4.12
    contour:
      version: 1.18.0
    ekco:
      version: 0.16.0
    kotsadm:
      applicationSlug: puppet-application-manager
      version: 1.64.0
    kubernetes:
      version: 1.21.8
    metricsServer:
      version: 0.4.1
    minio:
      version: 2020-01-25T02-50-51Z
    openebs:
      isLocalPVEEnabled: true
      localPVSStorageClassName: default
      version: 2.6.0
    prometheus:
      version: 0.49.0-17.1.1
    registry:
      version: 2.7.1
    velero:
      version: 1.6.2
    weave:
      podCidrRange: /22
      version: 2.8.1
    kurl:
      installerVersion: "v2022.03.11-0"
```

4. Build an installer using the `installer.yaml` file. Run the following command:

```bash
curl -s -X POST -H "Content-Type: text/yaml" --data-binary @installer.yaml https://kurl.sh/installer |grep -o "[^/]*$"
```

The output is a hash. Carefully save the hash for use in step 5.
5. Install a new cluster. To do so, you can either:
   a. Point your browser to https://kurl.sh/<HASH> (replacing <HASH> with the hash you generated in step 4) to see customized installation scripts and information.
   b. Follow the appropriate PAM documentation.
      - **For online installations:** Follow the steps in PAM HA online installation on page 49 or PAM standalone online installation on page 56, replacing the installation script with the following:
        ```
curl https://kurl.sh/<HASH> | sudo bash
```
      - **For offline installations:** Follow the steps in PAM HA offline installation on page 53 or PAM standalone offline installation on page 59, replacing the installation script with the following:
        ```
curl -LO https://k8s.kurl.sh/bundle/<HASH>.tar.gz
```

When setting up a new offline cluster as part of disaster recovery, add `kurl-registry-ip=<IP>` to the install options, replacing `<IP>` with the value you recorded when setting up snapshots.

**Note:** If you do not include the `kurl-registry-ip=<IP>` flag, the registry service will be assigned a new IP address that does not match the IP on the machine where the snapshot was created. You must align the registry service IP address on the new offline cluster to ensure that the restored configuration can pull images from the correct location.

**Important:** Do not install any Puppet applications after the PAM installation is complete. You'll recover your Puppet applications later in the process.

6. To recover using a snapshot saved to a host path, ensure user/group 1001 has full access on all nodes by running:
   ```
   chown -R 1001:1001 /<PATH/TO/HOSTPATH>
   ```

7. Configure the new cluster to connect to your snapshot storage location. Run the following to see the arguments needed to complete this task:
   ```
   kubectl kots -n default velero configure-{hostpath,nfs,aws-s3,other-s3,gcp} --help
   ```

8. Run `kubectl kots get backup` and wait for the list of snapshots to become available. This might take several minutes.

9. Start the restoration process by running `kubectl kots restore --from-backup <BACKUP NAME>`. The restoration process takes several minutes to complete. When the PAM UI is available, use it to monitor the status of the application.

**Note:** When restoring, wait for all restores to complete before making any changes. The following command waits for pods to finish restoring data from backup. Other pods may not be ready until updated configuration is deployed in the next step:

   ```
   kubectl get pod -o json | jq -r '.items[] | select(.metadata.annotations."backup.velero.io/backup-volumes") | .metadata.name' | xargs kubectl wait --for=condition=Ready pod --timeout=20m
   ```

This command requires the `jq` CLI tool to be installed. It is available in most Linux OS repositories.
10. After the restoration process completes, save your config and deploy:
   a) From the PAM UI, click **Config**.
   b) (Optional) If the new cluster's hostname is different from the old one, update the **Hostname**.
   c) Click **Save Config**.
   d) Deploy the application. You must save your config and deploy even if you haven't made any changes.

   **Note:** If you have installed Continuous Delivery for PE and changed the hostname, you need to update the webhooks that connect Continuous Delivery for PE with your source control provider. For information on how to do this, see **Update webhooks**.

---

**Migrating data between two systems with different architectures**

To perform data migration between two systems using different PAM architectures (from standalone to HA, or from HA to standalone), you must create a new cluster to recover to, then follow the process outlined below to recover your instance from a snapshot.

**Before you begin**

Review the requirements in **Data migration prerequisites** on page 80.

**Important:** If you are migrating from a legacy architecture, go to our Support Knowledge Base instructions for migrating to a supported architecture for your Puppet application:

- Migrate to a supported PAM architecture for Continuous Delivery for PE
- Migrate to a supported PAM architecture for Comply

---

1. On the original system, find the version of kURL your deployment is using by running the following command. Save the version for use in step 2.

   ```bash
   kubectl get configmap -n kurl kurl-current-config -o jsonpath="{.data.kurl-version}" && echo
   ```

2. Set up a new cluster to house the recovered instance, following the system requirements for your applications.

   **Important:** Do not install any Puppet applications after the PAM installation is complete. You'll recover your Puppet applications later in the process.

   - Install PAM using the version of kURL you retrieved earlier:
     - For online installs:
       ```bash
       curl -sSL https://k8s.kurl.sh/version/<VERSION STRING>/puppet-application-manager | sudo bash -s options
       ```
     - For offline installs:
       ```bash
       curl -O https://k8s.kurl.sh/bundle/version/<VERSION STRING>/puppet-application-manager.tar.gz
       ```
     - When setting up a new offline cluster as part of disaster recovery, add `kurl-registry-ip=<IP>` to the install options, replacing `<IP>` with the value you recorded when setting up snapshots.

   **Note:** If you do not include the `kurl-registry-ip=<IP>` flag, the registry service will be assigned a new IP address that does not match the IP on the machine where the snapshot was created. You must align the registry service IP address on the new offline cluster to ensure that the restored configuration can pull images from the correct location.
3. To recover using a snapshot saved to a **host path**, ensure user/group 1001 has full access on all nodes by running:

```
chown -R 1001:1001 /<PATH/TO/HOSTPATH>
```

4. Configure the new cluster to connect to your snapshot storage location. Run the following to see the arguments needed to complete this task:

```
kubectl kots -n default velero configure-{hostpath,nfs,aws-s3,other-s3,gcp} --help
```

5. Run `kubectl kots get backup` and wait for the list of snapshots to become available. This might take several minutes.

6. Start the restoration process by running `kubectl kots restore --from-backup <BACKUP NAME>`. The restoration process takes several minutes to complete. When the PAM UI is available, use it to monitor the status of the application.

   **Note:** When restoring, wait for all restores to complete before making any changes. The following command waits for pods to finish restoring data from backup. Other pods may not be ready until updated configuration is deployed in the next step:

   ```
kubectl get pod -o json | jq -r '.items[] | select(.metadata.annotations."backup.velero.io/backup-volumes") | .metadata.name' | xargs kubectl wait --for=condition=Ready pod --timeout=20m
```

   This command requires the `jq` CLI tool to be installed. It is available in most Linux OS repositories.

7. After the restoration process completes, save your config and deploy:
   a) From the PAM UI, click **Config**.
   b) (Optional) If the new cluster's hostname is different from the old one, update the **Hostname**.
   c) Click **Save Config**.
   d) Deploy the application. You must save your config and deploy even if you haven't made any changes.

   **Note:** If you have installed Continuous Delivery for PE and changed the hostname, you need to update the webhooks that connect Continuous Delivery for PE with your source control provider. For information on how to do this, see **Update webhooks**.

---

### Disaster recovery with PAM

It is important to prepare your system and regularly capture full snapshots. This backs up your data and makes it easier to restore your system if disaster recovery is needed.

#### Prepare your system to support future disaster recovery

To make sure your system is equipped to help you recover from a potential system failure, you must:

- Configure Puppet Application Manager (PAM) to support **Full Snapshots (Instance)**. For instructions on configuring snapshots, see **Backing up PAM using snapshots** on page 77.
- Configure Velero to use an external snapshot destination that is accessible to both your current cluster and future new clusters, such as S3 or NFS.
• Disaster recovery requires that the store backend used for backups is accessible from the new cluster. When setting up snapshots in an offline cluster, use the following command to record the registry service IP address:

```
kubectl -n kurl get svc registry -o jsonpath='{.spec.clusterIP}'
```

Make a record of the value returned by this command, because you'll need it to create a new cluster to restore to as part of disaster recovery.

• Run the latest version of PAM. Disaster recovery is only available on systems running PAM version 1.44.1 or newer.

Disaster recovery process

To perform a disaster recovery, you must create a new cluster to recover to and then recover your instance from a snapshot.

1. Find the version of kURL your original deployment was using.

   If you have access to the original cluster, you can use this command:

   ```
kubectl get configmap -n kurl kurl-current-config -o jsonpath="{.data.kurl-version}" && echo
```

   If you aren't able to run the command, you remember your PAM version, and you were on version 1.68.0 or later, you can look up the kURL version in the Component versions in PAM releases on page 45 table.

   If you don't remember your PAM version or you were on a version earlier than 1.68.0, you need to contact your technical account manager or Support for assistance.

2. If you have access to the original cluster, follow the steps for Migrating data between two systems with the same architecture on page 80.

   If your original cluster is completely offline and inaccessible, you'll need to contact your technical account manager or Support for assistance.

Restriction:

Your old and new clusters must have the same connection status (online or offline). Disaster recovery from an offline to an online cluster (or vice versa) is not supported.

Additionally, for offline installs, both the old and new clusters must use the same PAM version.

Troubleshooting PAM

Use this guide to troubleshoot issues with your Puppet Application Manager installation.

How to look up your Puppet Application Manager architecture

If you're running PAM on a Puppet-supported cluster, you can use the following command to determine your PAM architecture version:

```
kubectl get installer --sort-by=.metadata.creationTimestamp -o jsonpath='{.items[-1:].metadata.name}' ; echo
```

Depending on which architecture you used when installing, the command returns one of these values:

• **HA architecture**: puppet-application-manager

• **Standalone architecture**: puppet-application-manager-standalone

• **Legacy architecture**: Any other value, for example, puppet-application-manager-legacy, cd4pe, or comply
Resolve IP address range conflicts

When installing Puppet Application Manager, IP address ranges `10.96.0.0/22` and `10.32.0.0/22` must not be used by other nodes on the local network.

**Note:** The minimum size for CIDR blocks used by Puppet Application Manager are:

- **Standalone** - /24 for pod and service CIDRs
- **HA** - /23 for pod and service CIDRs
- Default of /22 is recommended to support future expansion

To resolve IP address range conflicts, create a `patch.yaml` file and add the `installer-spec-file=patch.yaml` argument when running the installation script (see below):

1. If you use IP addresses internally that overlap `10.32.0.0/22`, add the following to your `patch.yaml` file (10.40.0.0/23 used here as an example range):

```yaml
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
metadata:
  name: patch
spec:
  weave:
    podCIDR: 10.40.0.0/23
    podCidrRange: "/23"
```

2. If you use IP addresses internally that overlap `10.96.0.0/22`, add the following to your `patch.yaml` file (10.100.0.0/23 used here as an example range):

```yaml
spec:
  ...
  kubernetes:
    serviceCIDR: 10.100.0.0/23
    serviceCidrRange: "/23"
```

**CAUTION:** The `podCIDR` and `serviceCIDR` ranges must not overlap.

3. Once your `patch.yaml` file is set up, add the `installer-spec-file=patch.yaml` argument when you run the installation script:

```bash
cat install.sh | sudo bash -s airgap installer-spec-file=patch.yaml
```

**Remember:** Add the `installer-spec-file=patch.yaml` argument any time you re-run the installation script, such as when reinstalling to upgrade to a new version.

Related information

**Using sudo behind a proxy server** on page 92

Many of the commands you run to install or configure Puppet Application Manager (PAM) require root access. In the PAM documentation, commands that require root access use `sudo` to elevate privileges. If you're running PAM behind a proxy, `sudo` might not work correctly. If you're having trouble running commands with `sudo`, and you're behind a proxy, try switching to the `root` user and running the command without `sudo`.

Reset the PAM password

As part of the installation process, Puppet Application Manager (PAM) generates a password for you. You can update this password to one of your choosing after installation.
1. To reset the Puppet Application Manager password, run the following as the root user:

   ```bash
   kubectl -n default kots reset-password
   ```

   The system prompts you to enter a new password of your choosing.

2. If the command fails with an unknown command "kots" for "kubectl" error, it's because /usr/local/bin is not in the path. To address this error, either update the path to include /usr/local/bin, or run:

   ```bash
   /usr/local/bin/kubectl-kots reset-password default
   ```

**Update the PAM TLS certificate**

A self-signed TLS certificate secures the connection between your browser and Puppet Application Manager (PAM). Once the initial Puppet Application Manager setup process is complete, you can upload new certificates by enabling changes to the installation's Kubernetes secrets.

Use this process if you chose not to add a TLS certificate when installing Puppet Application Manager, or if you need to update your existing TLS certificate.

1. Enable changes to your installation's `kotsadm-tls` Kubernetes secret by running:

   ```bash
   kubectl -n default annotate secret kotsadm-tls acceptAnonymousUploads=1
   ```

2. Restart the `kurl-proxy` pod to deploy the change by running:

   ```bash
   kubectl delete pods $(kubectl get pods -A | grep kurl-proxy | awk '{print $2}')
   ```

3. Once the kurl-proxy pod restarts and is back up and running, navigate to `https://<HOSTNAME>:8800/tls` and upload your new TLS certificate.

**Reduce recovery time when a node fails**

If a node running a non-replicated service like PostgreSQL fails, expect some service downtime.

How much downtime depends on the following factors:

- Timeout for communication between Kubernetes services (at least one minute to mark the node as unreachable).
- Timeout for the ekco service to determine that pods need to be rescheduled. The default is five minutes after node is marked unreachable.
- Time to restart services (at least two minutes, possibly up to five minutes, if there are complex dependencies).

The ekco service can be configured to reschedule pods more quickly by configuring the installation with a `patch.yaml` similar to the following:

```yaml
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
metadata:
  name: patch
spec:
  ekco:
    nodeUnreachableToleration: 1m
```

Apply the patch during an install or upgrade by including `installer-spec-file=patch.yaml` as an install option.

**Important:** This patch needs to be included during all future upgrades to avoid resetting the option.
PAM components

Puppet Application Manager (PAM) uses a range of mandatory and optional components.

Support services

- Database: PostgreSQL (single instance) - https://www.postgresql.org/
- Object storage: previously MinIO - https://min.io, now Ceph - https://ceph.io
- tlser for basic TLS cert management - https://github.com/puppetlabs/tlser
- kurl_proxy for HTTPS proxying outside the Ingress (ports besides 80/443): https://github.com/replicatedhq/kots/tree/v1.36.1/kurl_proxy

Kubernetes components

- Ingress: Project Contour - https://projectcontour.io
- Kubernetes Cluster: kURL - https://kurl.sh
- Embedded kURL Cluster Operator: ekco - https://github.com/replicatedhq/ekco
- Admin Console: KOTS - https://kots.io
- Monitoring: Prometheus - https://prometheus.io
- Registry: Docker Registry - https://docs.docker.com/registry/

Optional components

Prometheus (+Grafana) and Velero (+Restic) are optional components:

- Prometheus+Grafana uses 112m/node + 600m CPU, 200MiB/node + 1750MiB RAM
- Velero+Restic uses 500m/node + 500m CPU, 512MiB/node + 128MiB RAM

If you do not need these optional components, they can be omitted from the initial install and further upgrades with a patch similar to the following:

```
apiVersion: cluster.kurl.sh/v1beta1
kind: Installer
metadata:
  name: patch
spec:
  prometheus:
    version: ''
  velero:
    version: ''
```

Important: This patch needs to be included during upgrades to avoid adding the components later.

If you want to remove optional components that are already installed, use the following command:

```
kubectl delete ns/monitoring ns/velero
```

Load balancing

The following load balancer requirements are needed for a HA install:

- A network (L4, TCP) load balancer for port 6443 across primary nodes. This is required for Kubernetes components to continue operating in the event that a node fails. The port is only accessed by the Kubernetes nodes and any admins using kubectl.
• A network (L4, TCP) or application (L7, HTTP/S) load balancer for ports 80, and 443 across all primaries and secondaries. This maintains access to applications in event of a node failure. Include 8800 if you want external access to the Puppet Application Manager UI.

| Note: | Include port 8000 for webhook callbacks if you are installing Continuous Delivery for PE. |

| Important: | If you are using application load balancing, be aware that Ingress items use Server Name Indication (SNI) to route requests, which may require additional configuration with your load balancer. If your load balancer does not support SNI for health checks, enable Enable load balancer HTTP health check in the Puppet Application Manager UI Config page. |

**Generate a support bundle**

When seeking support, you might be asked to generate and provide a support bundle. This bundle collects a large amount of logs, system information and application diagnostics.

To create a support bundle:

1. In Puppet Application Manager UI, click Troubleshoot > Generate a support bundle.
2. Select a method for generating the support bundle:
   • **Generate the bundle automatically.** Click Analyze <APPLICATION NAME> (<APPLICATION NAME> is replaced in the UI by the name of the Puppet application you have installed), and Puppet Application Manager generates the bundle for you and uploads it to the Troubleshoot page.
   • **Generate the bundle manually.** Click the prompt to generate a custom command for your installation, then run the command on your cluster. Follow the prompts to upload the bundle to Puppet Application Manager.
3. Review the collected data before forwarding it to Puppet, as it may contain sensitive information that you wish to redact.
4. Return to the Troubleshoot page, download the newly created support bundle, and send it to your Puppet Support contact.

**Create a support bundle from the command line**

If installation of the Puppet Application Manager, or upload of an app, on an embedded kURL cluster fails, it may not be possible to access the UI to generate a support bundle.

You can generate a support bundle by using the default kots.io spec. To do this, run the following command:

```shell
kubectl support-bundle https://kots.io
```

On an offline server, you can copy the default kots.io spec by using the following command:

```shell
curl -o spec.yaml https://kots.io -H 'User-agent:Replicated_Troubleshoot/v1beta1'
```

The spec can then be uploaded to the server. Use the local spec by running:

```shell
kubectl support-bundle /path/to/spec.yaml
```

If the Puppet Application Manager UI is working and the app is installed, you can use:

```shell
ekubectl support-bundle http://<server-address>:8800/api/v1/troubleshoot/<app-slug>
```

If the app is not installed but the Puppet Application Manager UI is running:

```shell
ekubectl support-bundle http://<server-address>:8800/api/v1/troubleshoot
```
If you do not already have the support-bundle kubectl plugin installed, install it by using the command below:

```bash
curl https://krew.sh/support-bundle | bash
```

Or by installing `krew2` and running:

```bash
kubect1 krew install support-bundle
```

**Using `sudo` behind a proxy server**

Many of the commands you run to install or configure Puppet Application Manager (PAM) require root access. In the PAM documentation, commands that require root access use `sudo` to elevate privileges. If you're running PAM behind a proxy, `sudo` might not work correctly. If you're having trouble running commands with `sudo`, and you're behind a proxy, try switching to the `root` user and running the command without `sudo`.

**kURL can only be upgraded two minor versions at a time**

Because kURL does not support upgrading more than two Kubernetes versions at once, if you're upgrading from an older version of PAM, you might need to follow a specific upgrade path to avoid failures.

- If you're on PAM version 1.56.0 or earlier, you must upgrade to PAM 1.80.0 before upgrading to PAM 1.81.1 or later.

Attempting to upgrade too far at once returns the following error message:

```
The currently installed kubernetes version is <CURRENT VERSION>. The requested version to upgrade to is <INVALID_TARGET_VERSION>. Kurl can only be upgraded two minor versions at time. Please install <VALID_TARGET_VERSION> first.
```

**Installing**

To begin using Puppet Comply, you must first complete the initial setup process.

- System requirements on page 92
  Refer to these system requirements to allow your Puppet Comply application to connect to Puppet Enterprise (PE).
- Set up Comply on page 93
  To start using Puppet Comply, you must complete the setup process, using both Puppet Application Manager (PAM) and Puppet Enterprise (PE).
- Uninstall Comply and remove PAM on page 104
  If you are running Comply only in a Puppet-supported cluster, and you want to delete Comply, Puppet Application Manager (PAM), and any related files, uninstall Comply by deleting the Comply application and purging the Kubernetes cluster.
- Uninstall Comply without removing PAM on page 104
  If you are running Comply in a customer-supported cluster, or if you don’t want to remove Puppet Application Manager (PAM) for other reasons, you can remove Comply from the customer-supported cluster without taking further action.
- Remove the CIS-CAT Pro Assessor from a node on page 105
  In rare cases, you might want to remove the CIS-CAT Pro Assessor from a node. For example, you can remove the assessor to exclude the node from compliance scans if the node is no longer relevant or will expire. You can also remove the assessor if the node has issues that are causing the assessor to malfunction.

**System requirements**

Refer to these system requirements to allow your Puppet Comply application to connect to Puppet Enterprise (PE).
Open port requirements

Comply is deployed on a Kubernetes cluster that requires the following ports:

<table>
<thead>
<tr>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PE ports</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8140</td>
<td>TCP</td>
<td>Preflight checks</td>
<td>Comply</td>
<td>Puppet primary server</td>
</tr>
<tr>
<td>8143</td>
<td>TCP</td>
<td>PE integration</td>
<td>Comply</td>
<td>PE Orchestrator</td>
</tr>
<tr>
<td>8081</td>
<td>TCP</td>
<td>PE integration</td>
<td>Comply</td>
<td>PuppetDB</td>
</tr>
<tr>
<td>4433</td>
<td>TCP</td>
<td>PE integration</td>
<td>Comply</td>
<td>PE RBAC</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Comply ports</td>
<td></td>
<td></td>
</tr>
<tr>
<td>443</td>
<td>TCP</td>
<td>Comply access</td>
<td>User browser</td>
<td>Comply UI</td>
</tr>
<tr>
<td>443</td>
<td>TCP</td>
<td>Sending reports</td>
<td>Scan target node</td>
<td>Comply server</td>
</tr>
<tr>
<td>30303</td>
<td>TCP</td>
<td>Assessor downloads and sending reports</td>
<td>Scan target node</td>
<td>Comply</td>
</tr>
</tbody>
</table>

Tip: Port 30303 is not required if you bring your own ingress. You can also set a custom Comply port in the Comply port field on the Config tab in Puppet Application Manager if you do not want to use port 30303.

Supported Puppet Enterprise versions

The following versions of Puppet Enterprise (PE) are supported for use with Comply:

<table>
<thead>
<tr>
<th>PE version</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2019.8.4 and later</td>
<td></td>
</tr>
</tbody>
</table>

For more information about PE versions, see Puppet Enterprise lifecycle policy.

Java Runtime Environment requirements

If you install the Comply module with the default setting of true for the manage_java option, the correct version of Java Runtime Environment (JRE) is installed automatically, and no further action is required.

Restriction: You cannot use the manage_java option on some operating systems, such as Ubuntu 16.04 and Mac OS X.

If you are independently managing JRE, ensure that the appropriate version is installed on the host system where the CIS-CAT Pro Assessor resides. JRE v1.8 or later is required. For the latest information about JRE requirements, see the CIS-CAT Pro Assessor Configuration Guide.

Set up Comply

To start using Puppet Comply, you must complete the setup process, using both Puppet Application Manager (PAM) and Puppet Enterprise (PE).

Important: Before you set up Comply, ensure that you have installed Puppet Application Manager (PAM) and Puppet Enterprise (PE) and have reviewed the system requirements.

Setting up Comply involves the following steps:
1. Configure Comply in Puppet Application Manager (PAM) in an online or offline environment. You can use the default ingress or, if you prefer, a custom NGINX ingress.
2. Configure Comply TLS certificates in Puppet Enterprise (PE). You can configure these for the default ingress or, if you prefer, a custom NGINX ingress.
3. Install the comply module.
4. Classify the nodes you want to scan in PE.
5. Deploy Comply.
6. Add your PE credentials to Comply.

   • Configure Comply in an online environment on page 94
     The Comply configuration process creates a Kubernetes cluster and installs the application on that cluster.
   • Configure Comply in an offline environment on page 95
     Configure Puppet Comply in an air-gapped or offline environment where the Comply host server does not have direct access to the internet.
   • Configure Comply TLS certificates on page 96
     You need to generate certificates for Comply in Puppet Enterprise (PE) to enable automatic upgrades of the CIS-CAT Pro Assessor and for tasks to upload reports.
   • Configure Comply for a custom NGINX ingress (online environment) on page 97
     The Comply configuration process requires some extra configuration parameters if you use a custom NGINX ingress.
   • Configure Comply for a custom NGINX ingress (offline environment) on page 98
     Configure Puppet Comply in an air-gapped or offline environment where the Comply host server does not have direct access to the internet.
   • Configure Comply TLS certificates for a custom NGINX ingress on page 99
     You need to generate certificates for Comply in Puppet Enterprise (PE) to enable automatic upgrades of the CIS-CAT assessor and for tasks to upload reports.
   • Install the Comply module on page 101
     Install the Comply module from Puppet Forge.
   • Classify the nodes you want to scan on page 102
     In Puppet Enterprise (PE), classify the nodes you want to scan. You can scan a maximum of 5000 nodes in a batch.
   • Deploy Comply on page 103
     Now that you have completed the setup process, you can deploy Comply.
   • Add your PE credentials to Comply on page 103
     To allow Comply to communicate with PE, you must add your PE credentials to Comply.

**Configure Comply in an online environment**

The Comply configuration process creates a Kubernetes cluster and installs the application on that cluster.

<table>
<thead>
<tr>
<th>Before you begin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Follow the instructions to install Puppet Application Manager.</td>
</tr>
</tbody>
</table>

1. In Puppet Application Manager, upload your Comply license and follow the prompts.
   You’ll be guided through the process of setting up SSL certificates, uploading a license, and checking to make sure your infrastructure meets Comply system requirements.

**Note:** The license file is issued by Puppet. If you do not have a license file, contact your Puppet representative. You must also agree to our license agreement. If your license terms update, for example the expiry date or number of licensed nodes, upload your updated license file to Puppet Application Manager.
2. To configure your installation, click **Config**.
   a) In the **Hostname** field, enter the fully qualified domain name (FQDN) that you want to use to access Comply. For example, this could be the name of the node you have installed Comply on. If you choose to use an FQDN that is different from the name of this node, you must configure your domain name system (DNS) to resolve the FQDN to the IP address of the Comply node.
   b) Configure any other settings on the page relevant to your installation. For example, you can determine how often the Comply inventory retrieves node and fact information from Puppet Enterprise. The default refresh interval for the Comply inventory is 24 hours, but you can specify a different value in the **Inventory Refresh Interval** section.
   c) When you have finished making any necessary changes to the configuration, click **Continue**.

3. Monitor the new version’s preflight checks. The **Running Checks** indicator is shown on the screen while your system is checked to make sure your cluster meets minimum system requirements. When the preflight check is complete:
   - If the status is **Checks Failed**, click **View preflights**. Correct the issues and click **Re-run**. Repeat this step as needed.
     
     **Important:** Do not move on until all preflight checks pass.
   - If the status is **Ready to Deploy**, move on to the next step.

**Generate Comply certificates in PE.**

**Configure Comply in an offline environment**

Configure Puppet Comply in an air-gapped or offline environment where the Comply host server does not have direct access to the internet.

**Before you begin**

Follow the instructions to install Puppet Application Manager.

Obtain the Comply bundle for air-gapped and offline environments by taking the following actions:

1. Locate the email that you received with the Comply licensing information. The email should include a password and a custom URL from which to download the bundle. If you no longer have the email, open a ticket with Puppet support so that you can obtain a custom URL and reset your password.
2. Navigate to the download portal (for example, [https://get.replicated.com/airgap/#/kots/comply/](https://get.replicated.com/airgap/#/kots/comply/)) and log in with the password.
3. Select **Embedded cluster**.
4. Click **Download airgap bundle**.

**1.** In Puppet Application Manager (PAM), upload your Comply license and follow the prompts. You’ll be guided through the process of setting up SSL certificates, uploading a license, and checking to make sure your infrastructure meets Comply system requirements.

**Note:** The license file is issued by Puppet. If you do not have a license file, contact your Puppet representative. You must also agree to our **license agreement**. If your license terms update, for example the expiry date or number of licensed nodes, upload your updated license file to Puppet Application Manager.

**2.** When prompted, upload the `.airgap` bundle for the most recent version of Comply.
3. To configure your installation, click **Config**.
   a) In the **Hostname** field, enter the fully qualified domain name (FQDN) that you want to use to access Comply.
      
      For example, this could be the name of the node you have installed Comply on. If you choose to use an FQDN that is different from the name of this node, you must configure your domain name system (DNS) to resolve the FQDN to the IP address of the Comply node.
   b) Configure any other settings on the page relevant to your installation. For example, you can determine how often the Comply inventory retrieves node and fact information from Puppet Enterprise. The default refresh interval for the Comply inventory is 24 hours, but you can specify a different value in the **Inventory Refresh Interval** section.
   c) When you have finished making any necessary changes to the configuration, click **Continue**.

4. Monitor the new version's preflight checks. The **Running Checks** indicator is shown on the screen while Comply checks your system to make sure your cluster meets minimum system requirements. When the preflight check is complete:
   - If the status is **Checks Failed**, click **View preflights**. Correct the issues and click **Re-run**. Repeat this step as needed.
     
     **Important:** Do not move on until all preflight checks pass.
   - If the status is **Ready to Deploy**, move on to the next step.

**Generate Comply certificates in PE.**

### Configure Comply TLS certificates

You need to generate certificates for Comply in Puppet Enterprise (PE) to enable automatic upgrades of the CIS-CAT Pro Assessor and for tasks to upload reports.

Certificates are required when setting up Comply for the following interactions:

- **Interactions between Comply and PE.** Interactions between Comply and PE require correct configuration of the CA certificate. Any issues with the CA certificate with regard to communication between Comply and PE result in an error on the Comply UI.
- **Agent runs.** If you have set up the Comply module to download the assessor from the Comply server (as opposed to being hosted locally), the assessor is downloaded using Mutual Transport Layer Security (MTLS) with the client certificate from the node. The Comply mtls-proxy component requires the configured TLS and CA certificate.
- **Scan task runs.** Running a scan sends reports back into Comply via an HTTP POST. This POST goes through the mtls-proxy and uses MTLS with the client certificate from the node.

Configuring Comply TLS certificates involves first generating the certificates in Puppet Enterprise (PE) and then setting up MTLS in PAM. MTLS enables a secure authenticated connection between your nodes and Comply.

For information on troubleshooting problems with certificates, see **Troubleshooting TLS issues in Comply** on page 123.

1. SSH into your PE primary server and generate the certificates:

   ```bash
   puppetserver ca generate --certname <COMPLY-HOSTNAME>
   ```

   This command does the following:
   
   - Saves the private key to `/etc/puppetlabs/puppet/ssl/private_keys/<COMPLY-HOSTNAME>.pem`
   - Saves the certificate to `/etc/puppetlabs/puppet/ssl/certs/<COMPLY-HOSTNAME>.pem`

2. Log in to Puppet Application Manager, click the **Version history** tab, and click **Check for update**.

3. Click the **Config** tab, and scroll down to **Transport layer security (TLS) certificates to interact with PE.**
4. Ensure **Use a NodePort** is selected. If you want to change the Comply port from the default (30303), add the new port number in the **Comply port for PE nodes** field.

   **Note:** To host the assessor on your own supported cluster via NGINX ingress, see Configure Comply for a custom NGINX ingress (online environment) on page 97 and Configure Comply TLS certificates for a custom NGINX ingress on page 99.

5. Enter the hostname of your PE instance in the **PE hostname** field to enable validation of the keys and certificates added in the next step.

6. Upload the signed certificate public key, the private key files, and the CA certificate, with the following locations:
   - Paste the contents of `/etc/puppetlabs/puppet/ssl/certs/<COMPLY-HOSTNAME>.pem` into the **TLS certificate** field.
   - Paste the contents of `/etc/puppetlabs/puppet/ssl/private_keys/<COMPLY-HOSTNAME>.pem` into the **TLS private key** field.
   - Paste the contents of `/etc/puppetlabs/puppet/ssl/ca/ca_crt.pem` into the **CA certificate** field.

7. Click **Save Config**.

8. Monitor the new version's preflight checks. The **Running Checks** indicator is shown on the screen while your system is checked to make sure your cluster meets minimum system requirements.

   The **Config: Check if we can connect to PE using provided certificates** preflight passes if the certificates are configured correctly.
   - If the preflight check status is **Checks Failed**, click **View preflights**. Correct the issues and click **Re-run**. Repeat this step as needed.

   **Important:** Do not move on until all preflight checks pass.

   - If the preflight check status is **Ready to Deploy**, proceed with the next step.

9. Click **Deploy**.

   **Install the comply module.**

### Configure Comply for a custom NGINX ingress (online environment)

The Comply configuration process requires some extra configuration parameters if you use a custom NGINX ingress.

**Before you begin**

Follow the instructions to **install Puppet Application Manager**.

1. In Puppet Application Manager, upload your Comply license and follow the prompts.

   You'll be guided through the process of setting up SSL certificates, uploading a license, and checking to make sure your infrastructure meets Comply system requirements.

   **Note:** The license file is issued by Puppet. If you do not have a license file, contact your Puppet representative. You must also agree to our **license agreement**. If your license terms update, for example the expiry date or number of licensed nodes, upload your updated license file to Puppet Application Manager.
2. To configure your installation, click **Config**.
   a) In the **Hostname** field, enter the fully qualified domain name (FQDN) that you want to use to access Comply.
      For example, this could be the name of the node you have installed Comply on. If you choose to use an FQDN that is different from the name of this node, you must configure your domain name system (DNS) to resolve the FQDN to the IP address of the Comply node.
   b) In the **Configure access** section, add the following annotations to configure the Ingress if you use cert-manager:
      
      ```
      kubernetes.io/ingress.class: nginx
      cert-manager.io/cluster-issuer: letsencrypt-prod
      ```
      c) Configure any other settings on the page relevant to your installation. For example, you can determine how often the Comply inventory retrieves node and fact information from Puppet Enterprise. The default refresh interval for the Comply inventory is 24 hours, but you can specify a different value in the **Inventory Refresh Interval** section.
   d) When you have finished, click **Continue**.

3. Monitor the new version's preflight checks. The **Running Checks** indicator is shown on the screen while your system is checked to make sure your cluster meets minimum system requirements. When the preflight check is complete:
   - If the status is **Checks Failed**, click **View preflights**. Correct the issues and click **Re-run**. Repeat this step as needed.
   - **Important:** Do not move on until all preflight checks pass.
   - If the status is **Ready to Deploy**, move on to the next step.

Configure Comply TLS certificates for a custom NGINX ingress on page 99.

**Configure Comply for a custom NGINX ingress (offline environment)**

Configure Puppet Comply in an air-gapped or offline environment where the Comply host server does not have direct access to the internet.

**Before you begin**

Follow the instructions to install Puppet Application Manager.

Obtain the Comply bundle for air-gapped and offline environments by taking the following actions:

1. Locate the email that you received with the Comply licensing information. The email should include a password and a custom URL from which to download the bundle. If you no longer have the email, open a ticket with Puppet support so that you can obtain a custom URL and reset your password.
2. Navigate to the download portal (for example, [https://get.replicated.com/airgap/#/kots/comply/](https://get.replicated.com/airgap/#/kots/comply/)) and log in with the password.
3. Select **Embedded cluster**.
4. Click **Download airgap bundle**.

1. In Puppet Application Manager (PAM), upload your Comply license and follow the prompts.
   You’ll be guided through the process of setting up SSL certificates, uploading a license, and checking to make sure your infrastructure meets Comply system requirements.

   **Note:** The license file is issued by Puppet. If you do not have a license file, contact your Puppet representative. You must also agree to our [license agreement](#). If your license terms update, for example the expiry date or number of licensed nodes, upload your updated license file to Puppet Application Manager.

2. When prompted, upload the `.airgap` bundle for the most recent version of Comply.
3. To configure your installation, click **Config**.
   a) In the **Hostname** field, enter the fully qualified domain name (FQDN) that you want to use to access Comply.
      For example, this could be the name of the node you have installed Comply on. If you choose to use an FQDN that is different from the name of this node, you must configure your domain name system (DNS) to resolve the FQDN to the IP address of the Comply node.
   b) In the **Configure access** section, add the following annotations to configure the Ingress if you use cert-manager:

   ```
   kubernetes.io/ingress.class: nginx
   cert-manager.io/cluster-issuer: letsencrypt-prod
   ```
   c) Configure any other settings on the page relevant to your installation. For example, you can determine how often the Comply inventory retrieves node and fact information from Puppet Enterprise. The default refresh interval for the Comply inventory is 24 hours, but you can specify a different value in the **Inventory Refresh Interval** section.
   d) When you have finished making any necessary changes to the configuration, click **Continue**.

4. Monitor the new version's preflight checks. The **Running Checks** indicator is shown on the screen while Comply checks your system to make sure your cluster meets minimum system requirements. When the preflight check is complete:
   - If the status is **Checks Failed**, click **View preflights**. Correct the issues and click **Re-run**. Repeat this step as needed.
     - **Important**: Do not move on until all preflight checks pass.
   - If the status is **Ready to Deploy**, move on to the next step.

Configure Comply TLS certificates for a custom NGINX ingress on page 99.

**Configure Comply TLS certificates for a custom NGINX ingress**

You need to generate certificates for Comply in Puppet Enterprise (PE) to enable automatic upgrades of the CIS-CAT assessor and for tasks to upload reports.

**Before you begin**

Make sure you have set up Comply in Puppet Application Manager (PAM) and you have followed the instructions in Configure Comply for a custom NGINX ingress (online environment) on page 97 or Configure Comply for a custom NGINX ingress (offline environment) on page 98 as appropriate to your implementation.

This process involves generating certificates in Puppet Enterprise (PE) and setting up Mutual Transport Layer Security (MTLS) in Puppet Application Manager (PAM). MTLS enables a secure authenticated connection between your nodes and Comply.

Certificates are required when setting up Comply for the following interactions:

- **Interactions between Comply and PE**. Interactions between Comply and PE require correct configuration of the CA certificate. Any issues with the CA certificate with regard to communication between Comply and PE result in an error on the Comply UI.
- **Agent runs**. If you have set up the Comply module to download the assessor from the Comply server (as opposed to being hosted locally) then the assessor is downloaded using MTLS with the client certificate from the node. The Comply mtls-proxy component requires the configured TLS and CA certificate.
- **Scan task runs**. Running a scan sends reports back into Comply via an HTTP POST. This POST goes through the mtls-proxy and uses MTLS with the client certificate from the node.

Configuring Comply TLS certificates involves first generating the certificates in Puppet Enterprise (PE) and then setting up MTLS in PAM. MTLS enables a secure authenticated connection between your nodes and Comply.

For information on troubleshooting problems with certificates, see Troubleshooting TLS issues in Comply on page 123.
1. SSH into your PE primary server and generate the certificates:

```bash
puppetserver ca generate --certname <COMPLY-HOSTNAME>
```

This command does the following:

- Saves the private key to `/etc/puppetlabs/puppet/ssl/private_keys/<COMPLY-HOSTNAME>.pem`
- Saves the certificate to `/etc/puppetlabs/puppet/ssl/certs/<COMPLY-HOSTNAME>.pem`

2. Log in to Puppet Application Manager, click the Version history tab, and click Check for update.

3. Click the Config tab, and scroll down to Transport layer security (TLS) certificates to interact with PE.

4. Select Use an ingress with a hostname.

5. Enter the fully-qualified domain name in the PE TLS hostname field using the same fully-qualified domain name that you used to generate the TLS certificates.

   **Important:** The fully-qualified domain name in the PE TLS hostname field MUST be different from that used in the Hostname field in the Required set-up area.

6. Check that the following annotations are in the SSL Passthrough Annotation field and add them if not:

   ```
   kubernetes.io/ingress.class: nginx
   nginx.ingress.kubernetes.io/ssl-passthrough: "true"
   ```

   **Note:** If you are not using NGINX, replace these annotations with those specific to your chosen ingress controller. For example, add OpenShift as an MLTS proxy for PE certificates.

7. Enter the hostname of your PE instance in the PE hostname field to enable validation of the keys and certificates added in the next step.

8. Copy the signed certificate public key, the private key files, and the CA certificate, to the following locations:

   - Paste the contents of `/etc/puppetlabs/puppet/ssl/certs/<COMPLY-HOSTNAME>.pem` to the TLS certificate field.
   - Paste the contents of `/etc/puppetlabs/puppet/ssl/private_keys/<COMPLY-HOSTNAME>.pem` to the TLS private key field.
   - Paste the contents of `/etc/puppetlabs/puppet/ssl/ca/ca_crt.pem` to the CA certificate field.

9. Click Save Config.

10. Monitor the new version's preflight checks. The Running Checks indicator is shown on the screen while Comply checks your system to make sure your cluster meets minimum system requirements. When the preflight check is complete:

   - If the status is Checks Failed, click View preflights. Correct the issues and click Re-run. Repeat this step as needed.

     **Important:** Do not move on until all preflight checks pass.

If the status is Ready to Deploy, move on to the next step.
11. The nginx-ingress controller configuration is configured with the \`--enable-ssl-passthrough\` setting disabled by default. This feature is required to enable passthrough in Ingress controller, allowing for the connection to be accepted by the application backends.

   a) To edit the configuration inline with the running configuration, execute:

```
kubectl edit deployment -n <namespace> <ingress-controller>
```

   b) Find the \`spec:` configuration section in the \`ingress-nginx-deployment.yaml\` and append \`--enable-ssl-passthrough\` under \`containers: -args\`. For example:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  ...
spec:
  containers:
    - args:
    ...
    - --enable-ssl-passthrough
```

Install the **comply** module.

**Install the Comply module**

Install the Comply module from Puppet Forge.

**Before you begin**

Make sure you have generated the Comply certificates in PE.

Modules are self-contained, shareable bundles of code and data. The Comply module contains a Bolt task — the tool that runs the CIS assessor on your nodes.

The Comply module lives on Puppet Forge, a repository of thousands of modules. If you're new to PE and Comply, see **Managing environment content with a Puppetfile** for more information on the Puppetfile and installing modules.
1. Go to the comply module on the Forge.

Follow the instructions in the r10k or Code Manager drop-down menu to add the module declaration to your Puppetfile. You also need to add its dependencies. For example:

```
# Puppet comply module
mod 'puppetlabs/comply', '2.5.0'

# dependencies for comply
mod 'puppet/archive', '6.0.2'
mod 'puppetlabs/chocolatey', '6.1.1'
mod 'puppetlabs/inifile', '5.2.0'
mod 'puppetlabs/java', '7.3.0'
mod 'puppetlabs/ruby_task_helper', '0.6.0'
mod 'puppetlabs/stdlib', '8.1.0'
mod 'puppetlabs/powershell', '5.0.0'
mod 'puppetlabs/registry', '4.0.1'
mod 'puppetlabs/pwshlib', '0.10.1'
```

If you don’t specify options, Code Manager installs the latest version and does not update it automatically. To always have the latest version installed, specify :latest and it updates automatically when a new version is released. Make sure you are always running the latest version of Comply if you intend to use the :latest keyword to update the Comply module. To install a specific version of the module that does not update automatically, specify the version number as a string.

**Important:** If you choose a specific version of the module, it must be the same as the Comply version. For example, version 2.3.0 of the module must be installed for Comply 2.3.0.

**Note:** When configuring the Comply module for macOS or CentOS 8, you must set the `manage_java` parameter to `false` because the management of Java is not supported.

2. SSH into your PE primary server and deploy the code:

```
puppet-code deploy --all
```

Classify the nodes you want to scan in Puppet Enterprise (PE).

**Classify the nodes you want to scan**

In Puppet Enterprise (PE), classify the nodes you want to scan. You can scan a maximum of 5000 nodes in a batch.

**Before you begin**
Make sure you have installed the comply module.

*Classification* is when you create a node group, add nodes to the group, and assign *classes* to the group — in this case, the *comply* class. Classes are the blocks of Puppet code used to configure nodes and assign resources to them. If you are new to Puppet, see *Grouping and classifying nodes* for more information.

**Tip:** For guidelines about scanning thousands of nodes in a single batch, see *Guidelines for running scans at scale* on page 113.

1. In the PE console, click Node groups.
2. Create a new node group or select an existing node group that you want to scan.
3. On the Classes tab — in the Add new class field — select the comply class.
4. Click Add class.
5. In your new comply class, select the scanner_source Parameter.

**Note:** Parameters allow a class to request external data.
6. Change the default parameter value to one of the following assessor distribution files:
   • If using the Puppet supported cluster: https://<COMPLY-HOSTNAME>:30303/assessor
   • If using NGINX Ingress: https://<PE-TLS-FQDN>/assessor

7. Click Add to node group, and then commit the changes.

8. Run Puppet twice.

Deploy Comply.

Related information
Create a custom profile on page 110
Create a custom profile based on an existing benchmark.

Deploy Comply

Now that you have completed the setup process, you can deploy Comply.

Before you begin
Make sure you have classified the nodes you want to scan in Puppet Enterprise (PE).

1. Navigate to Puppet Application Manager. Once the version is ready to deploy, click Deploy. On the Application tab, monitor the application for readiness.

   The application's status is shown as Missing for several minutes while deployment is underway.

   Tip: You can monitor the deployment's progress by running kubectl get pods --watch.

   When the deployment is complete, the application status changes to Ready. Comply is now deployed.

2. Navigate to https://<COMPLY-HOSTNAME> using the name of the Hostname FQDN you created earlier and sign into Comply using the default values:
   • username: comply
   • password: compliance

   You are then prompted to create a new password.

Add your PE credentials to Comply.

Add your PE credentials to Comply

To allow Comply to communicate with PE, you must add your PE credentials to Comply.

Before you begin
Make sure you have deployed Comply.

Adding your PE credentials authenticates Comply with Role Based Access Control (RBAC). Your PE account requires the following permissions:

<table>
<thead>
<tr>
<th>Type</th>
<th>Action</th>
<th>Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Console</td>
<td>View</td>
<td>-</td>
</tr>
<tr>
<td>Job Orchestrator</td>
<td>Start, stop and view jobs</td>
<td>-</td>
</tr>
<tr>
<td>Node Groups</td>
<td>View</td>
<td>All</td>
</tr>
<tr>
<td>Nodes</td>
<td>View node data from PuppetDB</td>
<td>-</td>
</tr>
<tr>
<td>Tasks</td>
<td>Run Tasks</td>
<td>Task: comply::backup_assessor Permitted on: All nodes</td>
</tr>
<tr>
<td>Type</td>
<td>Action</td>
<td>Instance</td>
</tr>
<tr>
<td>------</td>
<td>------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>Tasks</td>
<td>Run Tasks</td>
<td>Task: comply::ciscat_scan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Permitted on: All nodes</td>
</tr>
</tbody>
</table>

For more information on permissions, see User permissions and user roles.

2. Enter your PE hostname, username, and password.
3. Click Submit.

**Tip:** You can refresh the PE node and fact information by clicking Refresh data.

You’ll now see a list of your classified nodes on the Nodes page.

You have completed the Comply setup process! You can now start running CIS scans on your nodes. If you’re new to Comply, try out the beginner’s guide.

### Uninstall Comply and remove PAM

If you are running Comply only in a Puppet-supported cluster, and you want to delete Comply, Puppet Application Manager (PAM), and any related files, uninstall Comply by deleting the Comply application and purging the Kubernetes cluster.

**CAUTION:** By completing this procedure, you reset the Replicated installation.

1. From the command line of the node where Comply is installed, run the following command to delete the Comply application:

   ```
   # kubectl delete $(kubectl api-resources --verbs=delete -o name | paste -sd "," -) -A -l app.kubernetes.io/part-of=comply
   ```

2. On the same node, run the following command to uninstall the embedded Kubernetes cluster:

   ```
   curl https://k8s.kurl.sh/comply/tasks.sh | sudo bash -s reset
   ```

**Tip:** This command resets the Replicated installation with a purge.

3. Reboot your node to clear the kube-ipvs0 device.

### Uninstall Comply without removing PAM

If you are running Comply in a customer-supported cluster, or if you don’t want to remove Puppet Application Manager (PAM) for other reasons, you can remove Comply from the customer-supported cluster without taking further action.

From the command line of the node where Comply is installed, run the following command to delete the Comply application:

```bash
# kubectl delete $(kubectl api-resources --verbs=delete -o name | paste -sd "," -) -A -l app.kubernetes.io/part-of=comply
```
Remove the CIS-CAT Pro Assessor from a node

In rare cases, you might want to remove the CIS-CAT Pro Assessor from a node. For example, you can remove the assessor to exclude the node from compliance scans if the node is no longer relevant or will expire. You can also remove the assessor if the node has issues that are causing the assessor to malfunction.

To remove the assessor:

1. Declasify the node by taking the following actions:
   a. In the Puppet Enterprise console, click **Node groups**.
   b. Select the node group where the node is classified with a Comply class.
   c. If the node is pinned to a rule, click the **Rules** tab. Select the node name and click **Unpin**. If the node is not pinned to a rule, remove the class from the entire node group by clicking the **Classes** tab. Then, select the `scanner_source` parameter and click **Remove**.

2. On the command line of the node where the assessor is installed, run the appropriate command.
   On a Linux operating system, run the following command:
   ```bash
   rm -rf /opt/puppetlabs/comply/
   ```
   On a Microsoft Windows operating system, run the following command:
   ```bash
   Remove-Item -path C:\ProgramData\PuppetLabs\comply -recurse
   ```

3. Update the facts in Puppet Enterprise by running Puppet.
4. Retrieve the latest inventory from Puppet Enterprise by taking the following actions:
   a. In Comply, click **Settings**.
   b. Click **Refresh data**.

The assessor folder is removed along with the assessor JAR file and any backup copies of the JAR file. Because the node is declassified in Puppet Enterprise, Puppet will not reinstall the assessor during future runs. Declassified nodes are no longer visible to Comply and will be skipped in future compliance scans.

Optionally, if you want to resume scans on the node, you must classify the node so that it will be visible to Comply and the assessor will be reinstalled. For instructions, see [Classify the nodes you want to scan](#).

**Upgrading**

New versions of Puppet Comply are released regularly. Upgrading to the current version helps you take advantage of the latest features, fixes, and improvements.

**Important:** The CIS-CAT Pro Assessor setup process is embedded in the `comply` module. If you are upgrading to the latest version of the CIS-CAT Pro Assessor, upgrade the `comply` module **before** you upgrade the Comply application. Note that you cannot run scans until you complete both of these upgrades.

**Upgrade from Comply 1.0.4 to 2.10.0**

Comply 2.10.0 automatically upgrades the CIS-CAT Pro Assessor to the latest version when upgrading Comply. However, by adjusting your configuration you can choose to stay on a previous version of the assessor.

The upgrade process involves generating certificates in Puppet Enterprise (PE) and setting up Mutual Transport Layer Security (MTLS) in Puppet Application Manager (PAM). MTLS enables a secure authenticated connection between your nodes and Comply.

**Tips:**
• If you are upgrading Comply in an environment with thousands of nodes, see Guidelines for running scans at scale on page 113.

• If you are upgrading Comply to a version that includes a new assessor, you can expedite the process of installing the assessor on all nodes. In the PAM Config tab, in the CIS-CAT Pro Assessor upgrade section, select the checkbox to automatically start two Puppet runs after an assessor upgrade. To help prevent performance issues, enable this option only in small to medium Puppet deployments. If you enable this option, you can verify that a PE job was run: In Comply, select Activity Feed > Assessor Upgrades and click the assessor version to see the PE job number.

1. SSH into your PE primary server and generate the certificates:

   puppetserver ca generate --certname <COMPLY-HOSTNAME>

   This command does the following:

   • Saves the private key to /etc/puppetlabs/puppet/ssl/private_keys/<COMPLY-HOSTNAME>.pem
   • Saves the certificate to /etc/puppetlabs/puppet/ssl/certs/<COMPLY-HOSTNAME>.pem

2. Log in to PAM, click the Version history tab, and click Check for update.

3. Click the Config tab and scroll down to Transport layer security (TLS) certificates to interact with PE.

4. Upload the signed certificate public key, the private key files, and the CA certificate, with the following locations:

   • Paste the contents of /etc/puppetlabs/puppet/ssl/certs/<COMPLY-HOSTNAME>.pem to the TLS certificate field.
   • Paste the contents of /etc/puppetlabs/puppet/ssl/private_keys/<COMPLY-HOSTNAME>.pem to the TLS private key field.
   • Paste the contents of /etc/puppetlabs/puppet/ssl/ca/ca_crt.pem to the CA certificate field.

   Note: To host the assessor on your own supported cluster via NGINX ingress, select the Bring your own NGINX ingress checkbox and enter the FQDN in the PE TLS FQDN field — using the same FQDN that you used to generate the TLS certificates.

5. Navigate to CIS-CAT Pro Assessor version and ensure that the correct version of the CIS-CAT Pro Assessor is selected.

6. Click Save Config.

7. Navigate to Puppet Enterprise (PE), and update the default value of the comply class scanner_source parameter to one of the following assessor distribution files:

   • If using the Puppet supported cluster: https://<COMPLY-HOSTNAME>:30303/assessor
   • If using NGINX Ingress: scanner_source to: https://<PE-TLS-FQDN>/assessor

   For more information, see Classify the nodes you want to scan in PE.

8. Click Add to node group, and then commit the changes.

9. Upgrade the comply module:

   a) Update your Puppetfile with the latest version of the comply module and its dependencies.

   b) To stay on a previous version of the CIS-CAT Pro Assessor, configure the module's scanner_version and scanner_source class parameters to the desired version of the assessor. The version configured must match the version selected in step 5. To upgrade to the latest version of the CIS-CAT Pro Assessor, remove those parameters and the module defaults to the latest version.

   c) CAUTION: Only the latest version of the CIS-CAT Pro Assessor has the latest security fixes. Customers on older versions of the CIS-CAT Pro Assessor might be vulnerable to security issues.

   d) Deploy code by running the puppet-code deploy --all command.
10. Navigate back to PAM. After the pre-flight checks are successfully completed, click **Go to updated version**, and then click **Deploy**.

If the upgrade of an assessor on a node fails, the node is marked in red on the **Inventory** page. Failures may be due to network issues. If that is the case, Comply attempts to upgrade the node when connectivity returns. An hourly background task runs to check if nodes are upgraded. If a node is not upgraded and remains red on the **Inventory** page, run Puppet. If the upgrade continues to fail, see the Puppet agent logs for more information as described in **Agent logs**.

---

**Upgrade from Comply 2.2.2 to 2.3.0**

Comply 2.3.0 automatically upgrades the CIS-CAT assessor to the latest version every time you upgrade Comply.

**Before you begin**

Make sure you have generated certificates in Puppet Enterprise (PE) and set up Mutual Transport Layer Security (MTLS) in Puppet Application Manager (PAM). MTLS enables a secure authenticated connection between your nodes and Comply. For more information, see **Configure Comply TLS certificates** on page 96.

1. If you want Comply to update the CIS-CAT Assessor automatically, select **Automatically kick off PE jobs on assessor upgrade** on the **Config** page in Puppet Application Manager.

   If you select this option, on upgrade Comply kicks off 2 PE agent runs: the first to download the new assessor, and the second update the facts in PE.

   **Tip:** Because this option starts PE jobs automatically on upgrading Comply, systems administrators, especially of larger implementation, may wish to consider leaving this option unchecked. Assessor upgrade then takes place automatically when the next two PE jobs are run.

   Comply requires the latest version of the assessor on the node in order to perform runs. A background task runs to check if nodes have been upgraded every 15 minutes if this option is selected and every hour if it is not selected. If a node does not upgrade and remains red on the **Inventory** page, run the Puppet agent. If the upgrade continues to fail, see the Puppet agent logs for more information.

2. Click **Save Config**.

3. If you have not already configured the **comply class scanner_source** parameter, you can do so at this point. Otherwise proceed to the next step. Navigate to Puppet Enterprise (PE), and update the default **value** of the **comply class scanner_source** parameter to one of the following assessor distribution files:

   - If using the Puppet supported cluster: https://<COMPLY-HOSTNAME>:30303/assessor
   - If using NGINX Ingress scanner_source: https://<PE-TLS-FQDN>/assessor

   For more information, see **Classify the nodes you want to scan in PE**.

   Click **Add to node group**, and then commit the changes.

4. Upgrade the **comply module**.

   a) Update your Puppetfile with the latest version of the **comply module** and its dependencies.

   b) Deploy code by running the `puppet-code deploy --all` command.

   **Warning:** When upgrading the **comply module**, running the agent before Comply is updated may cause an error.
5. Navigate back to Puppet Application Manager. After pre-flight checks have completed successfully, click **Go to updated version**, and then click **Deploy**.

**Note:** If the upgrade of an assessor on a node fails, the node is marked in red on the **Inventory** page. Failures may be due to network issues. If that is the case, Comply attempts to upgrade the node once connectivity returns. An hourly background task runs to check if nodes have been upgraded or not. If a node does not upgrade and remains red on the **Inventory** page, run the Puppet agent. If the upgrade continues to fail, see the Puppet agent logs for more information.

## Upgrade Comply in an online environment

Check for downloads and deploy updates from the **Version history** tab in the Puppet Application Manager (PAM) UI.

### Before you begin
Upgrade the **comply** module.

1. In the PAM UI, click **Version history**.
2. Click **Check for updates**.
   Configure an automatic update check by clicking **Configure automatic updates**. You can check for updates hourly, every four hours, daily, weekly, or at a custom interval.
3. If an update is available, PAM downloads it for you and performs preflight checks on your system to make sure your cluster meets system requirements for the new version. Review the outcome of these checks by clicking **View preflight**.
4. When you're ready to upgrade to the new version of Comply, click **Deploy**.

## Upgrade Comply in an offline environment

If your environment does not have direct access to the internet, follow the documented procedure to upgrade Comply to the latest version.

### Before you begin
Upgrade the **comply** module.

1. Navigate to the portal provided to you by Puppet in the licence email, for example, https://get.replicated.com/airgap/#/kots/comply/, and log in with the password.
2. Select **Embedded cluster** and click **Download airgap bundle**.
3. Log into Puppet Application Manager:
   ```
   https://<PLATFORM-ADMIN-CONSOLE-ADDRESS>:8800
   ```
4. Select **Version history** and upload the new version of the `.airgap` file that you downloaded in step 2.
5. Click **Deploy**.

## Desired compliance

Set your desired compliance. This is the benchmark and profile that you assign to a particular node and that is scanned on that node by default. Generally, you set compliance only once for your nodes.
By default, Comply automatically assigns an appropriate benchmark for each operating system, along with a Level 1 profile, to nodes that have not been set based on fact information from PE. Accepting this option is the quickest way to get up and running with desired compliance.

The ## sign in the Profile assigned column on the Inventory page tells you that the desired compliance is set. You can view the node's information, including its assigned benchmark and profile, by clicking on the row assigned to the node. To change a node's desired compliance, follow the instructions in Manually set desired compliance on page 109. You can also follow the manual instructions to assign a different benchmark and profile to a node, or to assign a custom profile.

**Restriction:** Only one benchmark and profile can be assigned to each node.

### Manually set desired compliance

If you don't want to use the benchmark and profile that Comply assigns automatically to your node, you can set the benchmark and profile that you prefer from the Inventory page.

1. In Comply, click the **Node** column header on the Inventory page.
   
   Comply lists the nodes that have been classified with the comply class. If you do not see any nodes, ensure you have classified your nodes correctly.

2. Click the node for which you want to specify desired compliance.

   In the **Information** window that appears on the right, you can see facts about the node and whether desired compliance has been set.

3. Choose the CIS Benchmark and profile that you want to assign to the node.

   The benchmark and profile you set here is the desired compliance option for future scans.

   If you have created a custom profile, you can set it as the desired compliance by clicking **Use an associated custom profile?**

4. Click **Update**.

The ## sign in the Profile assigned column tells you that the desired compliance is set. You can view the node's information, including its assigned benchmark and profile, by clicking on the node. If you want to change a node's desired compliance, use the drop-down menu and click **Update**.

**Tip:** You can update compliance on **multiple** nodes simultaneously if the nodes are running on the same operating system, and the latest version of the CIS-CAT Pro Assessor is installed on each node. To update multiple nodes, go to the Inventory page and select the nodes. From the **Actions** menu, select **Set desired compliance**. In the **Benchmark**, **Profile**, and **Custom profile** fields, specify the desired compliance and click **Update**.

Now that you have applied desired compliance, you can run scans based on your selection.

**Related information**

- **Create a custom profile** on page 110
- Create a custom profile based on an existing benchmark.
- **Run an ad hoc scan** on page 114
  
  Run your desired compliance scan or an ad hoc scan on your nodes.

### Custom profiles

A custom profile is a benchmark profile that you customize to fit your organization's internally defined standards. You can base a custom profile on an existing benchmark and profile combination, and then specify which rules to apply.
For example, assume that your Center for Internet Security (CIS) Benchmark includes a rule that prohibits users from reusing any of the last 24 passwords that they specified. However, your organization enforces a stricter password policy. In this case, you could create a custom profile that enforces all other benchmark rules but excludes the CIS password rule. In this way, you would achieve more realistic compliance scores.

Custom profiles are typically created for long-term use. During an audit, you can note that a custom profile is applied to meet your organization’s requirements.

**Create a custom profile**

Create a custom profile based on an existing benchmark.

1. Navigate to **Custom profiles**.
2. Click **Create custom profile**.
3. Select a **Benchmark** and **Profile**.
4. Deselect rules in the profile that you do not want to scan and click **Next**.
5. Enter the name of the profile and, optionally, a description.
6. Click **Save custom profile**.

Your custom profile appears as an option when you assign the associated benchmark to a node.

Navigate to **Nodes** to set your custom profile as the desired compliance for your nodes or perform an ad hoc scan by selecting your custom profile on the **Scans** page.

To apply a custom profile to several nodes simultaneously, go to the **Inventory** page and select the nodes. From the **Actions** menu, select **Set desired compliance**. In the **Benchmark**, **Profile**, and **Custom profile** fields, specify the desired compliance and click **Update**.

**Restriction:** The selected nodes must be running on the same operating system, and the latest version of the CIS-CAT Pro Assessor must be installed on each node.

**Related information**

- [Manually set desired compliance](#) on page 109
- If you don't want to use the benchmark and profile that Comply assigns automatically to your node, you can set the benchmark and profile that you prefer from the **Inventory** page.
- [Run an ad hoc scan](#) on page 114
  
  Run your desired compliance scan or an ad hoc scan on your nodes.

**Delete a custom profile**

When a custom profile is no longer necessary, you can delete the profile.

⚠️ **CAUTION:** After you delete a custom profile, you cannot restore it.

1. In the **Custom profiles** table, select one or more profiles to delete.
2. In the **Actions** drop-down menu, select **Delete selected**.
3. When you are prompted to confirm the choice, click **Delete**.

Any nodes that were assigned to the deleted custom profile are unassigned. During future scans, the nodes will not be checked against the deleted custom profile. Any nodes that were assigned to the deleted custom profile will be reassigned to their default profile.

Previously run scan reports will continue to show results that reflect the custom profile. However, the custom profile appears in red and will be flagged with a warning triangle. The hover help will indicate that the custom profile no longer exists.
Exceptions

Each Center for Internet Security (CIS) Benchmark specifies many controls, commonly known as rules. In some cases, you might find it useful to create a temporary exception to a rule and apply the exception to one node, several nodes, or all nodes.

For example, assume that your environment includes legacy nodes that are installed on an operating system that is not CIS compliant, and you plan to decommission those nodes. You create an exception that specifies the rule, the affected nodes, the expiration date, the reason for the exception, and the name of the approver. On the next scan, the rule is not applied to the specified nodes, and the compliance score accurately reflects the exception. Later, after the nodes are decommissioned, the exception expires on your specified date. If an audit occurs, a record of the exception remains available on the Exceptions page.

Create an exception

When you create an exception to a rule, you prevent the rule from being applied to one or more nodes. If you run a scan while the exception is active, the compliance score of the rule is excluded from the overall compliance score of any specified nodes.

Tip: Exceptions are typically temporary with a specified expiration date and time. However, you can create an exception with no expiration date or time.

1. Click Scans > Scan reports and select a scan to which you want to add an exception.
2. On the Scan report page, on the Rules tab, locate the rule for which you want to create an exception. Click View report.
3. On the Scan report: Rule performance page, next to the rule name, click View rule detail.
4. On the Rule detail page, click Create exception and follow the exception creation workflow:
   a. Select a profile and, optionally, a custom profile. Click Next.
   b. Select one or more nodes to which the exception will apply. Click Set expiry.
   c. Optionally, set an expiration date, time, and time zone. Click Add details and review.
   d. Provide a name and reason for the exception.
   e. Optionally, for audit or tracking purposes, you can specify the name of the person who approved the exception and the associated ticket number, if applicable.
   f. Click Save exception and exit.

Tip: Alternatively, you can create an exception by going to the Comply navigation pane, clicking Exceptions and then clicking How do I create an exception?

Optionally, to see how the exception affects the compliance score, run a scan.

View an exception

To view one or more exceptions, go to the Puppet Comply navigation pane and click Exceptions.

You can filter exceptions by Active, Resolved, or Expired. For each exception, you can view the associated benchmark and profile. You can also see the rule, the number of nodes affected, and the expiration information.

When viewing exceptions, select an exception and then click View exception detail for a detailed view of the exception. Here you can find the nodes for which the exception is active. You can also edit the exception details by clicking Edit details, or resolve the exception by clicking Resolve. For more information on resolving exceptions, visit Resolve an exception on page 112.
The **Exceptions** page also includes the **How do I create an exception?** button. You can click the button for instructions on how to create an exception.

## Resolve an exception

To stop using an exception before its expiration date, resolve the exception for all nodes or a subset of nodes. After an exception is resolved, the rule scan results again count towards the overall compliance score for the impacted nodes.

1. Go to the Puppet Comply navigation pane and click **Exceptions**.
2. Specify the exception to resolve, and then click **View exception detail**.
3. To resolve the exception for all nodes, click **Resolve**.
   - Provide a reason for resolution, and an approver if applicable, and then click **Submit**.
4. To resolve the exception for only some nodes, select the checkboxes for the nodes on which you would like to resolve the exception, and then select **Resolve selected** from the **Actions** dropdown menu.
   - Provide a reason for resolution, and an approver if applicable, and then select **Submit**.

## Delete an exception

In general, exceptions should not be deleted because an auditor might want to see a record of the exception. However, you might want to delete an exception in rare cases. For example, if you create an exception by mistake, create an exception incorrectly, or you no longer require a record of the exception, you can delete it.

⚠️ **CAUTION:** After you delete an exception, you cannot restore it.

1. Go to the Puppet Comply navigation pane and click **Exceptions**.
2. Specify the exception to delete, then select **View exception detail**.
3. Select **Delete**.
4. Provide a reason for deletion, and an approver if applicable, then select **Delete**.

## CIS scans

Run your desired compliance scan or an ad hoc scan on your nodes.

*Ad hoc,* or manual, scans give you the freedom to run a CIS scan on your network straight away. Alternatively, you can schedule scans to run on a regular basis and at times when network traffic is low.

Scan reports provide you with in depth information on not just the timing of scans but the nodes affected and the rules that passed or failed.

- **Guidelines for running scans at scale** on page 113
You can run Puppet Comply to scan a maximum of 5000 nodes in a batch. Before you run Comply at scale, review the guidelines for configuring the environment and running the scan. The process of running Comply at scale was tested at Puppet Labs. Because many factors affect performance, results in your system environment might vary.
- **Run an ad hoc scan** on page 114
Run your desired compliance scan or an ad hoc scan on your nodes.
- **Scheduled scans** on page 115
In addition to manual ad hoc scans, you can also schedule scans.
- **CIS scan reports** on page 117
The **Scans** page displays all scans run within the defined scan data retention period.
**Guidelines for running scans at scale**

You can run Puppet Comply to scan a maximum of 5000 nodes in a batch. Before you run Comply at scale, review the guidelines for configuring the environment and running the scan. The process of running Comply at scale was tested at Puppet Labs. Because many factors affect performance, results in your system environment might vary.

**Configure the scan process**

To help optimize the scan process, follow the guidelines:

- In Puppet orchestrator, ensure that the `task_concurrency` parameter is set to the default value of 250. This value sets the maximum number of task or plan actions that can run concurrently in the orchestrator. If you set the parameter to 250 and run a scan of 5000 nodes, the orchestrator will be fully consumed until the scans are completed on all 5000 nodes. (For more information about optimizing performance, see [Tune task and plan performance in Puppet Enterprise (PE)].)
- Schedule scans to coincide with periods of minimal workflow to help ensure adequate network throughput.
- Plan adequate time for the initial inventory ingestion from Puppet Enterprise (PE). In lab testing, the ingestion of 5000 nodes took 2.5 minutes.
- Plan adequate time for the scan. In lab testing, a scan of 5000 nodes took 50 minutes.
- Configure scans in batches of up to 5000 nodes.

**Upgrade Comply in a large-scale environment**

Before you upgrade Comply in an environment with thousands of nodes, review the limitations and consider the best strategy for your environment.

During the standard upgrade process, a new version of the CIS-CAT Pro Assessor is downloaded to each Puppet-managed node. However, Comply supports a limited number of concurrent downloads of the assessor. In lab testing, a maximum of about 120 concurrent downloads was achieved. Thus, if you initiate an upgrade of thousands of nodes, not all nodes will be updated on the first run.

You can resolve the issue in one of the following ways:

- Run Puppet manually on a maximum of 120 nodes. Repeat the process until all nodes are updated.
- Configure Comply to host the assessor file on an internal web server and then upgrade Comply.

To host the assessor file internally and upgrade Comply, complete the following steps:

1. In the Puppet Enterprise (PE) console, click **Node Groups > PE Infrastructure > PE Agent > Classes**.
2. In the **Add new class** field, select the Comply class.
3. In the **Parameter name** field, select `scanner_source`.
4. Set the value of the scanner source to the URL where the assessor will be hosted. For example, the URL can have the following structure, where `server-hosting-assessor-ip` specifies the IP address of the server that will host the assessor:

   ```text
   http://server-hosting-assessor-ip/assessor/assessor.zip
   ```

5. Commit the changes.
6. In the PE console, click **Run > Puppet**.
7. Complete the upgrade process by selecting the relevant nodes and running the job.

**Optimize scanning and reporting at scale**

You can compare the results of your scanning and reporting processes against the results obtained in lab testing. If performance is not adequate in your environment, determine the cause of bottlenecks and address the issues.

In lab testing, with no other tasks running, the average run times for scans were as follows:

- A scan of 1000 nodes required about 10 minutes.
- A scan of 2000 nodes required about 20 minutes.
- A scan of 5000 nodes required about 50 minutes.

The run times for scans are affected by the host type. In general, scans on Microsoft Windows systems take longer than scans on *nix systems. Run times can vary significantly, depending on many other factors. For example, run times are longer for nodes with many user accounts and for nodes with many types of software installed. Results obtained in the lab represent an optimal use case.

The time required to generate and load a report increases with the number of nodes scanned. In lab testing, the initial inventory ingestion from PE took 2.5 minutes for 5000 nodes. The process of loading a report for 5000 nodes took 30 seconds to 3 minutes. The average report size was 5 MB.

To help understand performance issues, you can analyze log files. For more information, see Access logs on page 122.

**Run an ad hoc scan**

Run your desired compliance scan or an ad hoc scan on your nodes.

You can run scans on individual nodes by selecting the Scan nodes drop-down menu on the node's Node detail page, and then selecting Desired compliance or Custom options if you have those set up. Then, follow the scan wizard as outlined in steps 4-7.

You can also scan all nodes by selecting the Scan all nodes drop-down menu on the Dashboard page, and then selecting Desired compliance or Custom options if you have those set up. Then, follow the scan wizard as outlined in steps 4-7.

1. In Comply, click Scans, and then Run an ad hoc scan.
2. In the Benchmark drop-down menu, select Desired compliance or a benchmark and profile of your choice.
   
   If you have not set desired compliance, follow the instructions in Setting desired compliance.
3. Next, select an option from the Profile drop-down menu. To use a custom profile for this scan, select the Use an associated custom profile? option and choose the relevant option from the Custom profile drop-down menu.
4. Click Next to review the PE credentials and environment you want the scan to run on.
5. Click Next to see the nodes selected for scanning.
   
   To scan only a subset of nodes, deselect any nodes that you want to exclude.

   **Debug mode:** By default, assessor logs are set to WARN level. To troubleshoot an issue, you can set the logging level to DEBUG for the scan by clicking Run in debug mode. The assessor logs can then be retrieved from the individual node.

   On Linux and macOS platforms the assessor log is located at:

   /opt/puppetlabs/comply/Assessor-CLI/logs/assessor-cli.log

   On Windows the assessor log is located at:

   C:/ProgramData/PuppetLabs/comply/Assessor-CLI/logs/assessor-cli.log

   Note that scanning in debug mode increases the size of the assessor log file significantly.

6. Click Scan. To confirm, click Scan again.
   
   You are taken to the Activity feed, which lists each scan. Scans are run as a task in PE. To see the details of the job, click the job ID to be taken to PE.

   **Tip:** You can also run a scan by clicking the Scan nodes button at the top right corner on several pages. This option uses the nodes listed on the page you are currently viewing.
7. Optionally, to review the results of your scan, navigate to the **Compliance Dashboard** page. See **Scan results** for a description of the scan data.

**Related information**

- **Enforce CIS benchmarks** on page 121
  Puppet Comply provides visibility into your compliance status, but it cannot fix your failing nodes. Instead, you can use Puppet’s Compliance Enforcement Modules (CEM).

- **Custom profiles** on page 109
  A custom profile is a benchmark profile that you customize to fit your organization’s internally defined standards. You can base a custom profile on an existing benchmark and profile combination, and then specify which rules to apply.

- **Desired compliance** on page 108
  Set your desired compliance. This is the benchmark and profile that you assign to a particular node and that is scanned on that node by default. Generally, you set compliance only once for your nodes.

**Scheduled scans**

In addition to manual ad hoc scans, you can also schedule scans.

From the **Scan schedules** tab on the **Scans** page you can:

- View scheduled scans
- Create and edit scans
- Pause and resume scans
- End and delete scans.

Comply gives you many options for scheduling scans. You can schedule a one-off scan or a scan that repeats periodically:

- You can set scans to run on individual days of the week, every day, weekdays only, or weekends only.
- You can set scans to run on any day of the month. Be aware that if you, for example, set a scan to run on Day 31, the scan does not run in months that do not have 31 days.

**Scheduled scans table**

The table on the **Scan schedules** tab provides the following information:

- **Name** - The name assigned to the scan.
- **Nodes** - The number of nodes to be scanned.
- **Start** - The date and time scheduled for the scan to start according to the selected time zone.
- **End** - The date and time scheduled for the scan to finish according to the selected time zone.
- **Frequency** - How often the scan is scheduled to run.
- **Last run** - Date and time when the last scan of this type started.
- **Status** - Whether the scan is not currently running (*Not started*), has completed processing (*Ended*), or is currently running (*Active*).

**View details about a scheduled scan**

You can view details about a scheduled scan in the **Scheduled scan detail** window.

To view details about a scheduled scan:

1. On the **Scans** page, click **Scan schedules**.
2. Click the schedule for which you want to view details.
3. In the **Scheduled scan information** window, click **View detail**.
   On the **Scheduled scan detail** page, you can view details about the scan schedule, scan history, and affected nodes.
Pause and resume a scheduled scan

You can pause or resume a scheduled scan by using the Scheduled scan detail window.

To pause and resume a scheduled scan:

1. On the Scans page, click Scan schedules.
2. Click the schedule that you want to pause or resume.
3. In the Scheduled scan information window, click View detail.
4. To pause a scheduled scan, click Pause.
5. To resume a paused scan, click Resume.

You can also end a scan by clicking End.

**Restriction:** After you click End, the scan cannot be restarted.

Edit a scheduled scan

You can edit a scheduled scan by using the Scheduled scan detail window.

To edit a scheduled scan:

1. On the Scans page, click Scan schedules.
2. Click the schedule that you want to edit.
3. In the Scheduled scan information window, click View detail.
4. Click Edit details.
5. In the Edit scheduled scan window, update the details of the scan. For example, you can change the schedule type, the frequency, the start date and time, and the end date.

**Restriction:** You cannot add or remove nodes from the scheduled scan.

6. To save your changes, click Submit.

Delete a scheduled scan

You can delete a scheduled scan in any state (active, paused, or ended) by using the Scheduled scan detail window.

**CAUTION:** After you delete a scheduled scan, the scan cannot be restarted. However, scan reports that were generated by the scan remain available.

To delete a scheduled scan:

1. On the Scans page, click Scan schedules.
2. Click the schedule that you want to delete.
3. In the Scheduled scan information window, click View detail.
4. Click Delete.
5. When you are prompted to confirm the deletion, click Delete scheduled scan.

Create a one-off scheduled scan

You can schedule a compliance scan to run once.

To create a one-off scheduled scan:

1. On the Scans page, click Create a scheduled scan.
2. In the Environment field, select the environment and click Next.
3. Specify the nodes to scan by selecting the relevant checkboxes and click Next.

**Note:** If desired compliance is not set on any of the nodes, you are prompted to select a suggested desired compliance profile from the existing options or exclude those nodes from the scan.
4. Set the schedule:
   a) Select **Once only**.
   b) Specify the date for your scheduled scan.
   c) Specify the time and time zone for starting the scheduled scan.
   d) Click **Next**.

5. Enter a name in the **Scheduled scan name** field and, optionally, enter information in the **Description** field. Click **Save scheduled scan**.
   The scan schedule information is listed on the **Scan schedules** tab on the **Scans** page.

**Create a repeating scheduled scan**

You can schedule a compliance scan to run regularly.

To create a scheduled scan that repeats:

1. On the **Scans** page, click **Create a scheduled scan**.
2. In the **Environment** field, select the environment and click **Next**.
3. Specify the nodes to scan by selecting the relevant checkboxes and click **Next**.

   **Note:** If desired compliance is not set on any of the nodes, you are prompted to select a suggested desired compliance profile from the existing options or exclude those nodes from the scan.

4. Set the schedule:
   a) Select **Repeating**.
   b) Select the desired **Frequency** of the scan:
      - If you select the **Weekly** option, you can specify a day of the week for a weekly scan, or you can specify that the scan runs daily, only on weekdays, or only on weekends.
      - If you select the **Monthly** option, specify a day of the month for the scan.

   **Remember:** Your scan does not run if the month does not include the day you selected. For example, if you choose day 31, your scan cannot run in those months that do not have 31 days.

c) Choose the time and time zone when you want the scan to run.

d) Optionally, add an **End Date** for the scan schedule.
   
   If you choose an end date for your scheduled scan, no more scans run on or after that date and no scan reports are produced.

e) Click **Next**.

5. Enter a name in the **Scheduled scan name** field and, optionally, enter information in the **Description** field. Click **Save scheduled scan**.
   The scan schedule information is listed on the **Scan schedules** tab on the **Scans** page.

**CIS scan reports**

The **Scans** page displays all scans run within the defined scan data retention period.

The **Scan reports** tab on the **Scans** page provides the following information:

- **Name** - the name given to the scan when it was run.
- **Scan type** - ad hoc or scheduled.
- **Environment** - for example, production.
- **Nodes scanned** - the total number of nodes included in the scan.
- **Compliance** - the percentage of nodes that passed compliance in the scan.
- **Time started** - the date and time stamp when the scan was initiated.

Click the row assigned to any scan to go to its **Scan report** page, which provides details.
For more information on defining the scan data retention period, see Scan data retention policy on page 121.

Related information
CIS scan report details on page 118
The Scan report page provides detailed information on a selected CIS scan.
Scan results on page 118
View the results of your CIS scans and find out whether your nodes are compliant.

CIS scan report details
The Scan report page provides detailed information on a selected CIS scan.
The metrics bar at the top of the Scan report page is divided into two sections: Compliance scan status and Puppet Enterprise job status.

The Compliance scan status section provides a brief overview of the number of nodes that have passed and failed compliance, the error percentage, the rules that couldn't be evaluated across nodes, and the scan initiation date and time.

The Puppet Enterprise job status section shows the number of nodes that ran the CIS scanner job successfully, the number that failed to run the scanner job, and the number of nodes that showed an error for the scanner job.
On the Scans page, you can click Run an ad hoc scan to kick off a new scan.

More detailed information on the success and failure of rules is given on the Scan report page on the Nodes tab. The Rules tab provides detail on the performance of individual rules in the scan.

Rules tab
On the Scan report page, the table on the Rules tab lists all the rules that were assessed as part of the latest scan. The table provides information on the rule profile and the number of nodes on which the rule failed. To view details about a rule, locate the row that is associated with that rule and click View report.

Nodes tab
The table on the Nodes tab lists all nodes that were part of the latest scan. The table provides information on the node profile, and the percentage of rules in compliance on each node. To view details about a node, locate the row that is associated with that node and click View report.

Scan results
View the results of your CIS scans and find out whether your nodes are compliant.

Scan report metrics bar
On the Scan report metrics bar, the Compliance scan status section displays the compliance score with any applicable exceptions. The metrics bar also displays the percentage of nodes that passed, failed, or could not be evaluated, and the scan initiation date and time. The Puppet Enterprise job status section displays the status of scan jobs in Puppet Enterprise.

Compliance Dashboard
The Compliance Dashboard provides a breakdown of your latest CIS scan.
The dashboard has three components:
• The Compliance score component displays the percentage of nodes that were compliant during the latest scan. If exceptions were defined, the exceptions are exempted from the compliance score.
• The **Scan status** donut chart shows the percentage of rules that passed, failed, or reported a status of **Error** or **Unknown** across all nodes. The legend to the right shows the number of rules in each category.

• The **Node results** table lists information about the latest scan for each node.

### Node compliance

From the **Compliance Dashboard**, click a node name to navigate to the **Node detail** page and see the results of the latest scan on that node:

• The **Scan status** pane shows a status breakdown for the latest scan, including the total number of rules and the number of rules that passed, failed, reported an error, or had an unknown status. You can hover over the statuses in the legend to see percentages in the donut chart. The chart and legend reflect only the statuses that are subject to scoring. Non-scoring statuses (for example, cases in which a recommendation is not applicable or cannot be automatically assessed) are excluded. Statuses are described in the following table:

<table>
<thead>
<tr>
<th>Value</th>
<th>Included in scoring?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass</td>
<td>Yes</td>
<td>The target system or component state satisfied all the conditions of any checks or rules for the recommendation.</td>
</tr>
<tr>
<td>Fail</td>
<td>Yes</td>
<td>The target system or component state did not satisfy at least one condition of any checks or rules for the recommendation.</td>
</tr>
<tr>
<td>Error</td>
<td>Yes</td>
<td>The assessor checking engine encountered a system error and could not complete the test. The status of the target's compliance is not certain.</td>
</tr>
<tr>
<td>Unknown</td>
<td>Yes</td>
<td>The assessor was unable to collect, interpret, or evaluate against any check or rule conditions associated with the recommendation.</td>
</tr>
<tr>
<td>Other</td>
<td>No</td>
<td>The Other status includes all statuses that do not fall into the categories of Pass, Fail, Error, or Unknown. For details about the statuses that are included in the Other category, see the following rows.</td>
</tr>
<tr>
<td>Manual</td>
<td>No</td>
<td>This recommendation cannot be fully automated and requires manual evaluation. This status occurs when, in the CIS Benchmarks, a recommendation is deemed important but cannot be fully and reliably verified without a manual check by an organization. This status corresponds to the Extensible Configuration Checklist Description Format (XCCDF) term, <strong>Informational</strong>.</td>
</tr>
<tr>
<td>Not Applicable</td>
<td>No</td>
<td>Rules, checks, or both were not applicable to the target. This situation typically occurs when the benchmark and platform are mismatched.</td>
</tr>
<tr>
<td>Not Checked</td>
<td>No</td>
<td>The recommendation was not evaluated as there are no rule or check properties.</td>
</tr>
<tr>
<td>Not Selected</td>
<td>No</td>
<td>This recommendation was not part of the profile selected for the configuration assessment.</td>
</tr>
<tr>
<td>Informational</td>
<td>No</td>
<td>This is the same result that is displayed as Manual on the HTML report. The recommendation cannot be fully automated and requires manual evaluation.</td>
</tr>
</tbody>
</table>

• The **Rule scan results** table lists each rule that was checked and the status of that rule from the latest scan. The table also shows the date and time of the last successful scan for each rule.
Rule results

From the Compliance Dashboard, in the Node results table, click a node. Then, in the Rule scan results table, click a rule. The Rule detail page includes the following information:

- The Scan status pane shows the total number of nodes scanned and detailed results. You can hover over the results to see percentages in the donut chart. The compliance score in the chart and legend reflects only the statuses that are subject to scoring. Non-scoring statuses (for example, cases in which a recommendation is not applicable or cannot be automatically assessed) are excluded.
- A tabbed section displays information about each rule:
  - Fix — the steps you can take to fix the rule if it is failing on a node.
  - Description — information on what is being checked.
  - Rationale — the reason why it is important to check that rule.
- The Node results table lists each node the rule has been checked against and shows the current status, including when the node was last checked and when it last passed that rule. The table shows the profile, the environment in which the scan took place (for example, production or test), and any exceptions that apply.
- The Exceptions tab displays any exceptions that are relevant to the selected rule.

Exporting results

To export your results as a .csv file, select Export CSV at the top right of the Node results tab, and then choose whether to export raw data or a report summary. After exporting, you can download past reports from the Generated reports tab in the left menu.

The raw data export contains detailed scan results for each rule, including the rule's name, ID, and status, whether the rule has an exception against it, and details about the exception if applicable.

Rather than raw data, the summary export provides an exception score and an adjusted compliance score for each rule. The exception score is the latest overall compliance score for all nodes. This score accounts for any temporary compliance rule exceptions in place, and any rules with exceptions are excluded from the overall compliance score. The adjusted compliance score does not account for any temporary compliance rule exceptions, instead providing a true compliance score for all nodes.

Scan rule report

You can view a report about scan results for a single rule. The Scan report: Rule performance page lists the nodes on which the rule was run and the results.

From the Scans page, click a scan report. Ensure that the Rules tab is displayed. Locate a rule in the table and click View report.

The data includes:

- Overall compliance status for the nodes on which the rule was run
- The date and time when the scan was started
- The scan status for each node, including an indication of whether exceptions apply

Scan node report

You can view a report about scan results for a single node. The Scan report: Node performance page lists the rules that were run on the node and the results.

From the Scans page, click a scan report. Ensure that the Nodes tab is displayed. In the table, locate a node and click View report.

The data includes:

- Overall compliance status for the node
- The date and time when the scan was started
- The scan status for each rule, including an indication of whether exceptions apply

© 2022 Puppet, Inc., a Perforce company
Scan data retention policy

By default, no retention period is defined for scan data in Comply. You can, however, enable this feature on the Config tab in Puppet Application Manager.

Click Enable data retention policy in the Data retention policy area to define a data retention period for the default period of 14 weeks. Enter a numerical value in the Scan data retention period in weeks field to define a custom period in weeks that Comply must retain scan data.

Enforce CIS benchmarks

Puppet Comply provides visibility into your compliance status, but it cannot fix your failing nodes. Instead, you can use Puppet’s Compliance Enforcement Modules (CEM).

Available to premium content subscribers, CEM consists of two modules — cem_linux and cem_windows. These are supported Puppet modules developed specifically to bring your Puppet Enterprise (PE) managed nodes under CIS (Center for Internet Security) compliance.

By default, CEM enforces the latest CIS Level 1 benchmarks on your nodes, automating hundreds of operating system settings — the default profile depends on your operating system. You can also customize these configurations to suit your organization’s policies.

Tip: Starting with CEM for Linux 1.4.0, CEM also enforces the Security Technical Implementation Guides (STIG) developed by the US Defense Information Systems Agency (DISA). The DISA STIG standard, widely used by US government agencies, can be enforced by CEM on the Red Hat Enterprise Linux 7 operating system.

To get started with CEM, see Introducing the Compliance Enforcement Modules on page 125.

Troubleshooting

Use this section to troubleshoot issues with your Puppet Comply installation.

Reset your Comply password

If you forget your password, you can reset it in the user admin console.

1. SSH into your Comply node and run the following commands to retrieve the admin username and password:

   ```bash
   kubectl exec $(kubectl get pod -l app.kubernetes.io/name=comply-auth -o jsonpath="{.items[0].metadata.name}") -- /bin/bash -c 'cat /etc/keycloak/admin-user'
   kubectl exec $(kubectl get pod -l app.kubernetes.io/name=comply-auth -o jsonpath="{.items[0].metadata.name}") -- /bin/bash -c 'cat /etc/keycloak/admin-password'
   ```

2. Navigate to https://<COMPLY-HOSTNAME>/auth/admin using the FQDN of your Comply node.
3. Login using the credentials from step 1.
4. Navigate to Users.
5. Click **View all users** and select the user account you want to update, and click **Edit**.

6. Select the **Credentials** tab and the reset password.

### Access logs

If you run into issues with Puppet Comply, you can download the relevant log files. The Comply logs are stored in Puppet Application Manager.

1. Log into Puppet Application Manager — **https://<PUPPET-APPLICATION-MANAGER-ADDRESS>:8800**.
2. Select the **Troubleshoot** tab, and click **Analyse Comply**.
3. Download the bundle of log files.

### Resolve the Comply domain

If the Puppet Comply gatekeeper is unable to resolve the Comply domain, try the following troubleshooting steps.

When you assign a hostname to Comply, it needs to be resolved by the pods in your Kubernetes cluster. A preflight check verifies the domain you specified in the configuration is resolvable. You must ensure that the nodes can resolve their own hostnames, through either local host mapping or a reachable DNS server.

1. To verify your whether your hostname is resolvable, run the following commands:

   ```bash
   kubectl exec $(kubectl get pod -l app=kotsadm -o jsonpath="{.items[0].metadata.name}") -- /bin/sh -c 'curl --SI <hostname>
   
   If the hostname was resolved, the command returns an exit code 0 with no output.
   If the hostname cannot be resolved, the command returns an exit code 6. Proceed to step 2 to add DNS entries.
   
2. To add DNS entries for CoreDNS, run the following command to open the CoreDNS configuration maps:

   ```bash
   kubectl -n kube-system edit configmaps coredns
   
3. Add a **hosts** entry below **kubernetes**. This is where you configure the DNS entry for Comply. For example:

   ```yaml
   kubernetes cluster.local in-addr.arpa ip6.arpa {
     pods insecure
     fallthrough in-addr.arpa ip6.arpa
     ttl 30
   }
   hosts {
     10.23.24.25 comply.mycompany.net comply // IP_address canonical_hostname
     [aliases...]
     fallthrough
   }
   prometheus :9153
   
4. Run the command from step 1 to verify whether the DNS entry was updated:

   ```bash
   kubectl exec $(kubectl get pod -l app=kotsadm -o jsonpath="{.items[0].metadata.name}") -- /bin/sh -c 'curl --SI <hostname>'
   
5. Re-run the preflight checks.
Resolve a failed assessor upgrade

If an upgrade of the assessor has failed on one of your nodes, try the following troubleshooting step.

If the upgrade of an assessor on a node fails, the node is marked in red on the Inventory page. Failures may be due to network issues. If that is the case, Comply attempts to upgrade the node once connectivity returns. An hourly background task runs to check if nodes have been upgraded or not. If a node does not upgrade and remains red on the Inventory page, run the Puppet agent. If the upgrade continues to fail, see the Puppet agent logs for more information.

Resolve a failed scan

If an inappropriate version of Java Runtime Environment (JRE) is installed on the host system where the CIS-CAT Pro Assessor resides, you might see an error message about a failed scan.

The error message is similar to the following example:

```
Error: Scan did not complete successfully 'java _Server -b/opt/puppetlabs/comply/Assessor-CLI/benchmarks/CIS_Red_Hat_Enterprise_Linux_7_Benchmark_v3.1.1-xccdf.xml -Dciscat.license.filepath=/opt/puppetlabs/comply/Assessor-CLI/license/license.xml', 'Exception in thread "main" java.lang.UnsupportedClassVersionError: org/cisecurity/assessor/cli/Assessor : Unsupported major.minor version 52.0 at java.lang.ClassLoader.defineClass1(Native Method) at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142) at java.net.URLClassLoader.defineClass(URLClassLoader.java:443) at java.net.URLClassLoader.access$100(URLClassLoader.java:65) at java.net.URLClassLoader$1.run(URLClassLoader.java:355) at java.security.AccessController.doPrivileged(Native Method) at java.net.URLClassLoader.findClass(URLClassLoader.java:348) at java.lang.ClassLoader.loadClass(ClassLoader.java:363) at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.java:482)
```


To resolve the issue, ensure that JRE v1.8 or later is installed. For the latest information about JRE requirements, see the CIS-CAT Pro Assessor Configuration Guide.

Troubleshooting TLS issues in Comply

Incorrect configuration of TLS certificates when setting up Comply to work with PE can cause agents and/or scan tasks to fail.

There are two main certificate configuration errors that cause problems:

- If the CA certificate configured for Comply in Puppet Application Manager does not match the CA certificate the Puppet Enterprise certificate is signed with, then a trust store issue is returned upon setting up PE or trying to run a scan.
- A hostname issue can arise if a server identity check does not pass. The `dnsName` in the `subjectAltName` of the server certificate must match the hostname Comply is communicating with. The hostname configured in the Comply settings page for Puppet Enterprise must match one of the `dnsName` entries in the PE certificate.
Troubleshooting agent issues

Agents can fail if the certificate is configured incorrectly in two ways:

- The hostname could be the issue - the Comply TLS certificate must have the dnsName with which the agent is trying to contact the Comply server. This is relevant only if you set up the Comply module to download the assessor from the Comply server. The hostname the Comply server is contacted with is the hostname in the configured scanner_source parameter URL.
- The trust store may also be the problem if the Comply server TLS certificate and the client certificate are not signed by the same CA.

Troubleshooting scan task issues

Scan tasks can also fail if the certificate is configured incorrectly in two ways:

- If the hostname is incorrectly configured - The Comply TLS certificate must have the dnsName with which the agent is trying to contact the Comply server. The hostname used is passed through by the task and can be seen in the PE UI by checking the task parameters. Verify the task parameters to ensure that the hostname is correct.
- If the Comply server TLS certificate and the client certificate are not signed by the same CA a trust store issue occurs and this too can cause the scan task to fail.

Troubleshoot TLS from a node

You can run a simple cURL command to troubleshoot TLS issues from a node.

To troubleshoot TLS from a node:

1. SSH into a node and change to the SSL subdirectory:

   ```bash
   cd /etc/puppetlabs/puppet/ssl/
   ```

2. Issue the following cURL command replacing values in angle brackets with values relevant to your installation:

   ```bash
   curl -G --key private_keys/<local host key> --cacert certs/ca.pem --
cert ../certs/<local host cert> https://<comply-fqdn>:30303/assessor --
output /tmp/assessor.zip
   ```

   **Note:** If you are using your own ingress, issue the following command:

   ```bash
   curl -G --key private_keys/<local host key> --cacert certs/ca.pem --
cert ../certs/<local host cert> https://<PE TLS hostname>/assessor --
output /tmp/assessor.zip
   ```

   If this command fails, there is an issue with certificates. The error message can help to identify if there is a CA, client certificate or hostname issue.

   - If there is a hostname error, the output resembles the following error message:

     ```bash
     curl: (60) SSL: no alternative certificate subject name matches target host name '<comply-fqdn>'
     ```

   - If it is a CA issue, the output is likely to be similar to the example below:

     ```bash
     curl: (60) SSL certificate problem: unable to get local issuer certificate
     ```

   **Important:** This may be because the CA of the https://<comply-fqdn>:30303/assessor (or https://<PE TLS hostname>/assessor) does not match the CA passed to the cURL command or because there is a mismatch with the CA of the client certificate.
Introducing the Compliance Enforcement Modules

The Puppet Compliance Enforcement Modules (CEM) were developed to bring your Puppet Enterprise (PE)-managed nodes into compliance. CEM enforces Center for Internet Security (CIS) compliance rules. CIS Benchmarks are internationally recognized standards for securely configuring systems.

After you install and configure CEM, PE runs on any classified nodes without user intervention to scan for compliance. By default, CEM enforces CIS rules for the Level 1 profile.

Starting with CEM for Linux 1.4.0, CEM also enforces the Security Technical Implementation Guides (STIG) developed by the US Defense Information Systems Agency (DISA). The DISA STIG standard, widely used by US government agencies, can be enforced by CEM on the Red Hat Enterprise Linux 7 operating system.

The following sections provide instructions for installing CEM and customizing the configuration settings, if necessary, to meet your organization’s requirements.

Separate modules are provided for Linux nodes and for Microsoft Windows nodes:

- To manage Linux nodes, see CEM for Linux on page 125.
- To manage Windows nodes, see CEM for Windows on page 143.

CEM for Linux

You can deploy CEM for Linux to help ensure that your servers and workstations on Linux operating systems are compliant with CIS Benchmarks.

CEM for Linux supports the following operating systems: Red Hat Enterprise Linux 7, Red Hat Enterprise Linux 8, and CentOS Linux 7.

To take advantage of new features, fixes, and improvements, install the latest version of CEM. You can learn about the latest release by reviewing the Release notes on page 126. Then, to install CEM, follow the instructions in Installing CEM on page 132. By default, CEM runs automatically on any classified nodes and does not require configuration. However, if you want to configure CEM to meet your organization's requirements, follow the instructions in Configuring CEM on page 134.

- Release notes on page 126
  Review the release notes to learn about updates and resolved issues in the Compliance Enforcement Module (CEM) for Linux.
- Installing CEM on page 132
  Before you install CEM, complete the preparation steps: review the system requirements, install and configure Puppet Enterprise (PE), and purchase CEM. Then, install CEM and classify the nodes on which you want to enforce compliance.
- Upgrading CEM on page 134
  You can upgrade CEM for Linux to take advantage of the latest features, fixes, and improvements.
- Configuring CEM on page 134
  Configuration of CEM is optional. If you installed CEM and assigned the cem_linux class to one or more node groups in the Puppet Enterprise (PE) console, PE will run automatically and enforce the Center for Internet Security (CIS) Server Level 1 profile. However, if the default values leave your infrastructure in an undesirable state, or if you want to customize compliance to meet your organization's requirements, you can configure CEM.
- Auditing and querying issues identified during scans on page 143
  In some cases, a CIS compliance scan might identify an issue that you want to investigate and fix. To get started, you can run an audit or query.
- Reference: Benchmarks and controls on page 143
  For help with configuring CEM, review the Reference topics on Puppet Forge.
Release notes

Review the release notes to learn about updates and resolved issues in the Compliance Enforcement Module (CEM) for Linux.

v1.4.2
Released 8 November 2022

Added

• Added the ability to configure multiple rsyslog remote hosts to CEM for Linux. In previous releases, only single remote hosts were fully configurable. This software update simplifies the process of using the rsyslog software utility to forward logs to remote servers.
• Added an audit script for the V-204392 control, which is included in a Defense Information Systems Agency (DISA) Security Technical Implementation Guide (STIG) standard. The DISA STIG control helps to ensure that file permissions, ownership, and group membership of system files and commands match vendor values. You can use the new audit script to troubleshoot issues related to the control.

Changed

• Updated the Advanced Intrusion Detection Environment (AIDE) utility class to add support for the crontab scheduling utility. As a result, AIDE scans can be scheduled by using a crontab task rather than systemd timers.
• Updated CEM for Linux to ensure that the nullok option cannot be included in the system-auth file. The nullok option determines whether users can access a service with a blank password. This software update is designed to prevent unauthorized access to the system.

Fixed

• Fixed an issue that prevented certificates from being checked for Public Key Infrastructure (PKI) authentication. This software update affects users who are enforcing DISA STIG controls on a Red Hat Enterprise Linux (RHEL) operating system.
• Fixed an issue to help ensure that any new password must contain at least 8 characters that differ from the previous password. This software update affects users who are enforcing DISA STIG controls on a RHEL operating system.
• Fixed an issue related to the Center for Internet Security (CIS) Red Hat Enterprise Linux 8 Benchmark 2.0.0, Control 3.3.2: Ensure ICMP redirects are not accepted. This software update helps to ensure that the control is enforced so that Internet Control and Error Message Protocol redirects are prevented.
• Fixed an issue that caused the cem_semanage fact to run and log errors on an unsupported operating system, RHEL 6. semanage is a Security-Enhanced Linux (SELinux) management tool.
• Fixed an issue that caused catalog compilation errors when users selected the Network Time Protocol (NTP) synchronization service.

v1.4.1
Released 24 October 2022

Fixed

• Fixed an issue that prevented the cem_mount_info fact from resolving on Puppet Enterprise (PE) versions 2019.x.x. The issue prompted the following error message:

Facter: error while resolving custom facts...

To resolve the issue, you can install CEM Linux v1.4.1. To help avoid the issue, you can install the latest version of PE.
v1.4.0
Released 20 October 2022

Added
• Support for the DISA STIG standard on Red Hat Enterprise Linux (RHEL) 7:
  • For the first time, CEM supports a Security Technical Implementation Guide (STIG) standard developed by the US Defense Information Systems Agency (DISA). DISA STIG compliance is required for some infrastructures managed by the US government.
  • For the RHEL 7 operating system, STIG can be enabled by adding the following Hiera data to the control repository:
    ```
    cem_linux::benchmark: 'stig'
    ```
  • STIG supports Mission Assurance Category (MAC) levels 1, 2, and 3 and their associated “public,” “sensitive,” and “classified” profiles. STIG controls can be configured with their vulnerability ID (V-nnn) or rule ID (SV-nnn).
  • To support STIG controls that require information audits, new Bolt tasks were added.
  • The following new Facter facts were added: `cem_mount_info`, `cem_nfs_exports`, `cem_semanage`, and `cem_sssd_domains`.
  • For a list of supported STIG controls and configurations, see the CEM Linux Reference.

Changed
• The product documentation was revised to improve usability and retrievability:
  • The changelog was migrated from Puppet Forge to the central location for Puppet documentation, Puppet Docs. The changelog was renamed to Release notes on page 126.
  • The readme file was transformed into a series of topics with a structure similar to other Puppet documentation. The CEM topics can now be viewed on Puppet Docs, starting with Introducing the Compliance Enforcement Modules on page 125.
  • The Reference, Tasks, and Dependencies documentation, which is generated automatically, remains on Puppet Forge.
  • To facilitate the implementation of DISA STIG standards, new parameters were introduced for some resources. The new parameters do not affect default configurations and are backward compatible with previous user configurations. All parameters are documented in the CEM Linux Reference.

Fixed
• Fixed an issue that caused the `auditd` service to restart multiple times. The problem was caused by an incorrect sequence when setting a rule for immutable configuration.
• Fixed an issue that caused catalog compilations to fail although the specified configuration was valid. The failures occurred when certain time-server options were specified for the chrony implementation of the Network Time Protocol.

v1.3.2
Released 8 September 2022

Added
• The `Ensure core dump storage is disabled` and `Ensure core dump backtraces are disabled` controls are now enforced on Red Hat Enterprise Linux (RHEL) 8 systems.
• Added a new enforcement mode, `disabled`, so that you can disable Security Enhanced Linux (SELinux) in your environment.

Changed
• The **Ensure audit log is disabled when audit logs are full** control is updated to halt the machine when the audit log is full. This change helps to ensure better compliance with Center for Internet Security (CIS) recommendations.

• To simplify configuration, the `ntp` and `chrony` classes were combined into the `timesync` class.

**Fixed**

• The **Disable USB Storage** control is updated to work as designed.

• The regular expression for matching Linux username patterns is updated to accept capital letters.

• Rules in the `/etc/auditd/rules.d` directory are now loaded by using the `augenrules --load` command. This fix helps to ensure that all rule files within the directory are loaded into the kernel.

• Fixed the per-resource ordering process by using the correct metaparameter `before` instead of `subscribe`.

• Fixed a parsing error for `chrony` that caused catalog compilation failures.

• Fixed a command injection vulnerability that could occur when unsanitized user input was used in the `command`, `onlyif`, or `unless` parameters of an `exec` resource.

• Fixed an issue with the permissions of Secure Shell (SSH) host private keys to ensure that the permissions are sufficiently restrictive.

• Fixed the `cem_systemctl` feature to return a result of `false` without error messages in Puppet run logs when the feature is evaluated on Microsoft Windows machines.

• Fixed an issue with the `cem_mta` fact that caused errors in RHEL 6.

**v1.3.1**

Released 18 August 2022

**Fixed**

• Controls that configure `journald` now properly configure the `journald.conf` file.

• The `cem_coredump` fact will no longer attempt to resolve on nodes that do not support `systemctl`.

• The `cem_grub_cfg` fact will now identify the correct GRUB2 configuration file on Red Hat Enterprise Linux (RHEL).

• The Center for Internet Security (CIS)-specific parameters `enable_systemd_journal` and `enable_nopasswd_sudo_prune` now function correctly.

• Fixed how Ruby code is loaded during Continuous Delivery for Puppet Enterprise impact analysis. This update fixes a bug that caused impact analysis to fail after upgrading CEM for Linux to v1.3.0.

• Fixed invalid default parameter values that caused catalog compilation failures when enforcing the control `ensure_password_creation_requirements_are_configured`.

• Fixed a duplicate resource defaults statement that caused catalog compilation failures when selecting `ntp` as the time synchronization service.

**v1.3.0**

Released 3 August 2022

**Changed**

• The core architecture for the module has changed. These changes should be transparent to the user. However, using Hiera automatic parameter lookup to set configurations directly on classes in the `cem_linux::benchmarks::controls::*` namespace will no longer work. This configuration method was not supported previously, and with the new architecture those classes have been removed and replaced with module Hiera data.

• For more information on the new architecture, see Configuring CEM on page 134.

• The Reference: Benchmarks and controls on page 143 was revised to improve usability. Sample configurations are provided for each supported control.

**Fixed**

© 2022 Puppet, Inc., a Perforce company
• Added proper containment to the `cem_coredump` fact so that it will no longer run on operating systems that do not support it.
• Fixed how Network Time Protocol (NTP) options are handled. This fix resolves failures that occurred when using certain timeserver options.

v1.2.0
Released 24 May 2022

Added
• Added the Center for Internet Security (CIS) Level 2 Server profile for Red Hat Enterprise Linux (RHEL) 7.

Changed
• Updated the CIS RHEL 8 benchmark to version 2.0.0.
• Removed support for CentOS 8 because the operating system has reached End of Life (EOL). CEM for Linux has never supported CentOS Stream, and with non-stream CentOS 8 being EOL, support for it was removed entirely.

Fixed
• Fixed an issue that prevented the `coredump` configuration setting from being properly enforced. Now, you can use the module to configure core dumps.
• Fixed an issue related to file system mount points, which were not properly remounted after changes in mount-option enforcement. This issue prevented certain configuration changes from being applied.

v1.1.4
Released 25 March 2022

Changed
• Updated the `audit_user_homedir` task to prevent the task from modifying permissions on top-level directories: /boot, /boot/, /etc, /lib, /lib64, /proc, /proc/, /home, /opt, /tmp, /var, and /srv/. The `audit_user_homedir` task can still modify permissions on subdirectories within the listed directories, except for /boot and /proc.
• In the `audit_user_homedir` task, added `rtkit` to the list of ignored usernames. Because `rtkit` is a system user, CIS states that the home directory permissions for `rtkit` should not be audited.

v1.1.3
Released 24 March 2022

Fixed
• Fixed a bug in the `audit_user_homedir` task to prevent the inadvertent modification of permissions on bin directories: /bin, /sbin, /usr/bin, and /usr/sbin.

v1.1.2
Released 16 March 2022

Added
• Added a section to the CEM Reference about configuring `chrony/ntp` time servers.

Changed
• Expanded the range of versions in the `metadata.json` file so that users can install the latest modules to meet dependency requirements.

Fixed
• Fixed a bug in the `cem_linux::utils::timesync` configuration option that caused Puppet run failures when Network Time Protocol (NTP) was selected for time synchronization.
- Fixed a bug that caused a Puppet run failure during attempts to use a template to provide the Message of the Day (MOTD).
- Fixed a bug relating to unsupported options in the `auditd` config template on Red Hat Enterprise Linux (RHEL) 7. The bug caused startup failures for the `auditd` service.

**v1.1.1**

Released 25 January 2022

**Fixed**

- Fixed an issue related to non-idempotent resources when managing permissions for the `Grub2` bootloader configuration. This issue affected Red Hat Enterprise Linux (RHEL) systems that did not use Extensible Firmware Interface (EFI) mode.

**v1.1.0**

Released 14 December 2021

**Added**

- Enforcement for Center for Internet Security (CIS) Red Hat Enterprise Linux (RHEL) 8 Server Level 2 recommendations.
- Updates related to bootloader configurations. Configurations, including password settings, can now be managed through the CEM module on systems that use the `grub2` bootloader. You can also opt in to automatically regenerate the bootloader config files after changes are made. For details, see the CEM for Linux readme file.
- Permissions management for log files in the `/var/log` directory is now available in the module. Previously, you had to run a Bolt task to manage permissions for log files. Because this feature is now supported natively, the Bolt task `cem_linux::logfile_permissions` was removed.
- Added a new fact, `cem_grub_cfg`. This fact contains information related to general `grub` configuration on the machine.

**Changed**

- Replaced the `camptocamp-systemd` module with the supported `puppet-systemd` module. To help ensure compatibility, you must update your Puppetfile to use the `puppet-systemd` module v3.5.0 or later.
- The `cem_uefi_boot` fact was changed to `cem_efi` and more information was added to the fact. The new name is more representative because the fact now includes boot and other information.

**Restriction**

- When you scan a node with Puppet Comply after applying CEM, some recommendations that are enforced by CEM might be reported as having failed the scan. This issue is due to bugs in the CIS-CAT Pro Assessor that is used by Comply. For more information, see the readme file.

**v1.0.0**

Released 28 September 2021

This is the initial public release of CEM for Linux.

**Known issues and limitations**

The current release includes known issues and limitations. In most cases, workarounds are provided.

**Comply scan issues**

During a Comply scan, you might see errors about Center for Internet Security (CIS) recommended guidelines that are not enforced. These error messages are triggered by bugs in the CIS-CAT Pro Assessor that is bundled with Comply. CEM does correctly enforce these settings.

The following Comply scan errors might be reported:
• Red Hat Enterprise Linux (RHEL) Benchmark v2.0.0:
  • 1.4.2 - Ensure permissions on bootloader are configured
    • On EFI systems, the script that was run by the CIS-CAT Pro Assessor did not locate the correct grub file path. Permissions are set correctly by CEM. No action is required.
  • 1.4.1 - Ensure bootloader password is set
    • On EFI systems, the script that was run by the CIS-CAT Pro Assessor did not locate the correct grub file path. It is not mandatory to set a bootloader password. However, if you want to set a password to protect your system against unauthorized startup, follow the instructions in Set a bootloader password on page 140.
  • 4.1.2.3 Ensure system is disabled when audit logs are full
    • This is set to halt by CEM. The CIS-CAT Pro Assessor incorrectly shows this as a scan failure. No action is required.
  • 5.2.18 Ensure SSH MaxSessions is set to 10 or less
    • This is set to 10 by default. The CIS-CAT Pro Assessor incorrectly shows this as a scan failure. The scanner is looking for <=4 instead of <=10. No action is required.

General issues and limitations

• Multifactor controls and configurations are outside the scope of CEM for Linux. However, you can set up multifactor authentication for an infrastructure that is protected by CEM for Linux by implementing a network authentication system. For example, you can set up one-time password authentication on the client side by following the instructions in Setting up multi-factor authentication on Linux systems.
• If you are enforcing the DISA STIG standard on the RHEL 7 operating system, the V-204392 auditing control is not working as designed. The control is missing a script that audits file permissions, ownership, and group membership of system files and commands. As a workaround, you can audit file permissions manually.
• Starting with v1.3.0, CEM for Linux implements a new architecture. If you upgrade CEM from v1.2.0 or earlier to v1.3.0 or later, and you encounter errors, try restarting the pe-puppetserver service or restarting or reloading Puppet Server. For instructions, see Restarting Puppet Server.
• You cannot use the iolog_dir option to specify a directory for sudo log files. If you attempt to use the iolog_dir option in the sudoers file to specify a log directory other than the default, errors are reported by the Augeas program. Augeas is a tool used for configuration editing in CEM.
• CEM cannot create file system partitions. This limitation can cause certain scanner checks to fail.
• CEM cannot set permissions on removable media partitions. To set the required permissions on these partitions, ensure that nodev,nosuid,noexec exists in the options portion of /etc/fstab for the partition.
• Support for the eXecute Disable/No eXecute (XD/NX) hardware feature is dependent on the host kernel and cannot be configured by CEM. If you plan to enable XD/NX support, ensure that you are using up-to-date kernels. If you plan to enable XD/NX support on newer kernels, be aware that CEM cannot manage this feature.
• To comply with CIS recommendations, you must prevent root users from logging onto the system console. Because this action requires knowledge of the site, you must configure this control manually by removing entries in /etc/security for consoles that are not in secure locations.
• CEM does not enforce authselect controls for CIS 2.0.0 5.4.x on Red Hat Enterprise Linux 8. Enforcement requires site knowledge and can break network authentication. CIS recommends that you do not enforce this control. CEM includes a Bolt task, audit_authselect, to audit these controls.
• You can configure the ensure_nodev_option_set_on_home_partition control only if the /home setting is mounted on its own partition. Puppet does not create a partition for /home.
• If your system is running on Red Hat Enterprise Linux 8:
  • The ensure_nis_server_is_not_installed control is dependent on ensure_rpcbind_is_not_installed_or_the__rpcbind_services_are_masked. If you enforce ensure_nis_server_is_not_installed, you must also enforce ensure_rpcbind_is_not_installed_or_the__rpcbind_services_are_masked.
  • The ensure_nfs_utils_is_not_installed_or_the__nfs_server_service_is_masked control is dependent on
ensure_RPCbind_is_not_installed_or_the__RPCbind_services_are_masked. If you do not enforce
ensure_RPCbind_is_not_installed_or_the__RPCbind_services_are_masked, you must also not enforce
ensure_nfs_utils_is_not_installed_or_the__nfs_server_service_is_masked.

• The ensure_the_running_and_on_disk_configuration_is_the_same control is always enforced if auditd is managed by CEM.

• The ensure_users_must_provide_password_for_escalation control is disabled by default. You might want to enable this control to help ensure CIS compliance. However, a potential risk exists: It is possible that removing \texttt{NOPASSWD:} from sudoers files could invalidate the syntax of those files and break system authentication. If you accept the risk and want to enable this control, set the top-level configuration option enable_nopasswd_sudo_prune to true.

• If your system is running on Red Hat Enterprise Linux 7 or CentOS 7:
  
  • The ensure_RPCbind_is_not_installed_or_the__RPCbind_services_are_masked control is dependent on ensure_nfs-utils_is_not_installed_or_the__nfsserver_service_is_masked. If you enforce ensure_RPCbind_is_not_installed_or_the__RPCbind_services_are_masked, you must also enforce ensure_nfs-utils_is_not_installed_or_the__nfsserver_service_is_masked.
  
  • The disable_wireless_interfaces control requires that you install the NetworkManager package and that the service is running.

### Installing CEM

Before you install CEM, complete the preparation steps: review the system requirements, install and configure Puppet Enterprise (PE), and purchase CEM. Then, install CEM and classify the nodes on which you want to enforce compliance.

- **Prepare to install the module** on page 132
  
  To help ensure the successful deployment of CEM, complete the preparation steps.

- **Install the module and classify nodes** on page 133
  
  To deploy CEM, you must install the module and then classify the nodes on which you want to enforce Center for Internet Security (CIS) compliance.

- **Uninstall the module** on page 133
  
  To stop using CEM, you can uninstall the \texttt{cem_linux} module. Alternatively, to stop using CEM on one or more nodes, declassify the nodes to remove their association with the \texttt{cem_linux} class.

### Prepare to install the module

To help ensure the successful deployment of CEM, complete the preparation steps.
1. Review the following table to ensure that CEM can meet your organization’s requirements. CEM for Linux supports the following operating systems, frameworks, and standards:

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Framework or standard</th>
<th>Level</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Enterprise Linux (RHEL) 7</td>
<td>CIS Benchmarks v3.1.1</td>
<td>1, 2</td>
<td>Server</td>
</tr>
<tr>
<td>CentOS Linux 7</td>
<td>CIS Benchmarks v3.1.2</td>
<td>1, 2</td>
<td>Server</td>
</tr>
<tr>
<td>RHEL 8</td>
<td>CIS Benchmarks v2.0.0</td>
<td>1, 2</td>
<td>Server</td>
</tr>
</tbody>
</table>

2. Review the dependencies to ensure that your infrastructure will meet the requirements. Go to Puppet Forge and review the Dependencies tab.

3. Install Puppet Enterprise (PE) 2019.8.4 or later. For instructions, see Installing.

4. Configure PE. For instructions, see Configuring Puppet Enterprise.

5. Ensure that the Puppet utility, which is built into PE, is at level 6.23.0 or later.

6. To purchase CEM software, contact a Puppet by Perforce sales representative. For more information, see Contact Sales.

Install the module and classify nodes

To deploy CEM, you must install the module and then classify the nodes on which you want to enforce Center for Internet Security (CIS) compliance.

Before you begin

In some cases, compliance controls can negatively impact services that run on nodes. To help avoid possible issues, you can install and evaluate CEM in a test environment before running CEM in a production environment.

1. Download CEM from Puppet Forge. The module is available as a premium content subscription. For more information, see the Premium content page.
2. Install the module by using Code Manager, a code management tool. For instructions, see the Premium content page.
3. Specify the nodes on which you want Puppet Enterprise (PE) to run and enforce compliance. To specify the nodes, open the PE console and assign the `cem_linux` class to one or more node groups. For instructions about creating and classifying node groups, see Grouping and classifying nodes.

Uninstall the module

To stop using CEM, you can uninstall the `cem_linux` module. Alternatively, to stop using CEM on one or more nodes, declassify the nodes to remove their association with the `cem_linux` class.

**CAUTION:** If you uninstall CEM or declassify one or more nodes, Puppet Enterprise (PE) no longer runs on the affected nodes to enforce Center for Internet Security (CIS) compliance. When you uninstall CEM or declassify a node, the node does not revert to its pre-CEM state.

Take one of the following actions:

- To uninstall the CEM module, follow the instructions in Uninstalling modules.
- To declassify nodes, you can remove the nodes from the node group that is associated with the `cem_linux` class. For instructions, see Remove nodes from a node group. Alternatively, to stop using CEM on an entire node group, you can remove the `cem_linux` class from the node group. For instructions, see Remove classes from a node group.
Upgrading CEM

You can upgrade CEM for Linux to take advantage of the latest features, fixes, and improvements. To upgrade the module, you must update the CEM declaration in the Puppetfile. For instructions about modifying the Puppetfile, see Declare Forge modules in the Puppetfile.

For example, to upgrade the `cem_linux` module to version 1.3.0, you would specify the CEM declaration as shown:

```mod 'puppetlabs/cem_linux', '1.3.0'```

**Troubleshooting tip:** Starting with v1.3.0, CEM for Linux implements a new architecture. If you upgrade CEM from v1.2.0 or earlier to v1.3.0 or later, and you encounter errors, try restarting the `pe-puppetserver` service or restarting or reloading Puppet Server. For instructions, see Restarting Puppet Server.

Configuring CEM

Configuration of CEM is optional. If you installed CEM and assigned the `cem_linux` class to one or more node groups in the Puppet Enterprise (PE) console, PE will run automatically and enforce the Center for Internet Security (CIS) Server Level 1 profile. However, if the default values leave your infrastructure in an undesirable state, or if you want to customize compliance to meet your organization's requirements, you can configure CEM.

For example, if a CIS control sets the maximum password age at 365 days, but your organization requires a password change every 90 days, you can configure CEM accordingly.

You configure CEM by using the Hiera tool in your control repository. For more information, see About Hiera and Getting started with Hiera.

For general information about CEM configuration options, see Overview of configuration options on page 134. For detailed information about CEM configuration options, see the Reference: Benchmarks and controls on page 143.

For configuration examples, see How to configure the module: Examples and guidelines on page 138.

**CAUTION:** CEM default settings are fully CIS compliant. Too much customization can cause your configuration to become non-compliant.

- **Overview of configuration options** on page 134
  Configuration options include top-level options, benchmark options, and Center for Internet Security (CIS)-specific options.
- **How to configure the module: Examples and guidelines** on page 138
  Configuration examples are provided to help you understand how CEM is used in a production environment. Guidelines are provided to help optimize your configuration.

Overview of configuration options

Configuration options include top-level options, benchmark options, and Center for Internet Security (CIS)-specific options.

Find and set configuration options

You can find the configuration options for a specific control in the Reference: Benchmarks and controls on page 143. The reference is divided into sections, with each section representing a CIS Benchmark. Each benchmark has a list of associated controls, and additional information is provided for each control:

- **Parameters:** Configuration options for a control, along with the data type and default value.
- **Supported Levels:** The supported levels for a CIS control.
- **Supported Profiles:** The supported profiles for a CIS control.
- **Hiera Configuration Example:** Snippet of Hiera that can be used to configure a control.
• **Alternate Config IDs**: The alternate config IDs for a control. Any of these config IDs, along with the full control name, can be used as a key in the `control_config` hash.

• **Resource**: The name of the Puppet resource that enforces the control.

**Guidelines for specifying config IDs**

You can specify controls in the `control_config` hash by referencing the full control name, the control number, the normalized control name, or the normalized control number. You cannot mix and match these forms and must pick a single config ID form to use for your config. Full control names and control numbers are copied verbatim from the benchmarks and are case-sensitive. Normalized control names have lowercase letters and contain only alphanumeric characters and underscores. Normalized control numbers are always prefixed with a `c` and contain only numeric characters separated by underscores.

Example of alternate config IDs:

• Full control name: (L1) Ensure 'Enforce password history' is set to '24 or more password(s)'
• Control number: 1.1.1
• Normalized control name: `ensure_enforce_password_history_is_set_to_24_or_more_passwords`
• Normalized number: `c1_1_1`

**Top-level configuration options**

In the Hiera tool, the top-level configuration options are found directly under the `cem_linux` namespace. If you must customize CEM to meet your organization's requirements, you can specify values for the top-level configuration options.

In Hiera, the top-level options are prefixed with `cem_linux:`. The following list describes the options:

• `benchmark` - Enum ['cis'] - the compliance framework to use. CEM supports only cis. Default: cis.
• `config` - Optional [Hash] - the location for all non-top-level configuration options. Default: undef.
• `allow_on_kubernetes_node` - Boolean - If `cem_linux` detects that it is running on a Kubernetes cluster node or host, CEM does not enforce controls, and it logs a warning. In this way, CEM helps to prevent the accidental enforcement of incorrect compliance settings that can render Kubernetes non-functional. Default: false.
• `regenerate_grub2_config` - Boolean - Some configurations in CEM for Linux modify the Grub2 bootloader configuration. To regenerate the Grub2 configuration after applying a change, set this parameter to true. If you do not set this parameter to true, you must manually regenerate the Grub2 configuration. Default: false.
• `set_grub2_password` - Boolean - Set the password for the Grub2 bootloader. If you set this value to true, you must also set the `grub2_superuser` and `grub2_superuser_password` parameters, or configure the specific bootloader password control by using the `control_configs` option. Default: false.
• `grub2_superuser` - Optional [String [1]] - The superuser for the Grub2 bootloader if you set the `set_grub2_password` parameter to true. Default: undef.
• `grub2_superuser_password` - Optional [Sensitive [String]] - The superuser password for the Grub2 bootloader if you set the `set_grub2_password` parameter to true. This value is sensitive in terms of security, and should be stored in a sensitive data type. Default: undef.

**Hiera example**

The following example configures CEM for Linux to regenerate the Grub2 bootloader config on a node using the CIS benchmark:

```bash
cem_linux::benchmark: 'cis'
cem_linux::allow_on_kubernetes_node: false
cem_linux::regenerate_grub2_config: true
cem_linux::config:
  ...
```

© 2022 Puppet, Inc., a Perforce company
**Benchmark configuration options**

Each Center for Internet Security (CIS) Benchmark is associated with a set of configuration options. You can use the Hiera tool to specify values for the benchmark configuration options.

The benchmark configuration options are available as key-value pairs within the `cem_linux::config` hash. The following options are available:

- **only**: Optional [Array[String]] — takes an array of control class names (manifests/benchmarks/<benchmark>/controls/*.pp). Classes specified here are included in the catalog. This option takes precedence over `ignore`. Default: `undef`.
- **ignore**: Optional [Array[String]] — takes an array of control class names (manifests/benchmarks/<benchmark>/controls/*.pp). The classes specified here are not included in the catalog. If `only` is specified, this option does nothing. Default: `undef`.

**CIS-specific configuration options**

To meet your organization's requirements, you can specify CIS-specific configuration options, such as settings related to firewalls and log files. Use the Hiera tool to specify these options.

The CIS-specific configuration options are available as key-value pairs within the `cem_linux::config` hash:

- **profile**: Optional [Enum['server', 'workstation']] — the name of the benchmark profile. The only value supported by CEM is `server`. Default: `server`.
- **level**: Optional [Enum['1', '2']] — the name of the profile level. The only value supported by CEM is `1`. Default: `1`.
- **firewall_type**: Optional [Enum['iptables', 'firewalld', 'unmanaged']] — the preferred firewall provider. If set to `unmanaged`, CEM will not enforce firewall-related rules. Default: `firewalld`.
- **enable_systemd_journal**: Optional [Boolean] - Whether to enable the systemd-journal logging service. The default value is `false`. If this option is enabled, the systemd-journal-remote package will be installed and the systemd-journal-upload.service service will be enabled. However, several configuration parameters are required to ensure that the systemd-journal-upload.service functions correctly:

  ```ruby
  cem_linux::config:
  control_configs:
    'ensure_systemd_journal_remote_is_configured':
      address: '<IP address or FQDN of the remote host>'
      server_key_file: '<path to the server key file>'
      server_certificate_file: '<path to the server certificate file>'
      trusted_certificate_file: '<path to the trusted certificate file>'
  ```

**Options specific to Red Hat Enterprise Linux 8**

The authselect utility can be used to configure user authentication on a Red Hat Enterprise Linux (RHEL) host. If you installed CEM on a RHEL 8 operating system, authselect options are available, but should be avoided in almost all cases. The authselect utility is disabled by default because enablement of authselect can break authentication methods, and use of the utility requires extensive configuration.

**Authselect options**

The following authselect options are available for RHEL 8:

- **use_authselect**: Optional [Boolean] - Whether to use authselect to manage most authentication options. Defaults to `false`.
- **authselect_profile**: Optional [String] - Profile for authselect configuration options. If using the authselect utility, you must specify an authselect profile. Defaults to `undef`.

© 2022 Puppet, Inc., a Perforce company
Restrictions

CAUTION:

- If a node is joined to an Active Directory domain or to Red Hat Identity Management (idM), do not enable the authselect utility. Enabling the authselect utility on these nodes will break your authentication configurations.
- You cannot enable the authselect option if you are using pluggable authentication modules (PAMs) for application management.

Enabling the authselect utility

Both of the authselect options must be set directly in the cem_linux::config hash for the authselect utility to work properly. All authselect configurations are managed via the ensure_custom_authselect_profile_is_used control, regardless of whether you use a custom profile.

To enable the authselect utility:

1. Set the config option use_authselect to true.
2. Specify an authselect profile with the config option authselect_profile.

By default, cem_linux uses standard PAM rules to configure the authentication controls specified by CIS. However, if you are enforcing CIS compliance on RHEL 8, CIS guidelines call for the authselect utility to be used. The following configuration example shows how to enable the authselect utility on a node by using the minimal system default profile:

```yaml
# control-repo/data/nodes/<node name>.yaml
---
cem_linux::config:
  use_authselect: true
  authselect_profile: 'minimal'
```

Custom authselect profiles

If you are enforcing CIS compliance on a RHEL 8 system and you want to enable additional features for your authselect profile, you can create a custom profile.

To create a custom authselect profile in cem_linux, prefix the profile name in authselect_profile with custom/. If the custom profile does not exist on the node, the profile will be created automatically. The following example shows how to create and use a custom profile, my_custom_profile, which is based on the minimal system profile with additional features enabled:

```yaml
# control-repo/data/nodes/<node name>.yaml
---
cem_linux::config:
  use_authselect: true
  authselect_profile: 'custom/my_custom_profile'
control_configs:
  ensure_custom_authselect_profile_is_used:
    custom_profile_base: 'minimal'
    profile_features:
      - with-faillock
      - with-mkhomedir
```

For more information about authselect features, see the authselect documentation in the Red Hat Customer Portal.
How to configure the module: Examples and guidelines

Configuration examples are provided to help you understand how CEM is used in a production environment. Guidelines are provided to help optimize your configuration.

- **Basic configuration example** on page 138
  When you specify a compliance framework, CEM is configured to provide rule enforcement and configuration for that framework. For example, to enforce the Center for Internet Security (CIS) Server Level 1 benchmark for a node, you must classify the node with the `cem_linux` class, set the `benchmark` parameter to `cis`, and run Puppet.

- **Advanced configuration example** on page 139
  Building on the basic configuration example, the advanced configuration example customizes the Advanced Intrusion Detection Environment (AIDE) configuration file in Hiera.

- **Automatically regenerate and enforce bootloader configurations** on page 139
  In rare cases, it might be useful to enable automatic regeneration of the bootloader configuration, and you might want to set a bootloader password. By setting a bootloader password, you can provide an extra layer of security for your infrastructure.

- **Configure DISA STIG** on page 140
  The US Defense Information Systems Agency (DISA) has developed Security Technical Implementation Guide (STIG) standards that are designed to secure information systems and software.

- **Configure custom logrotate rules** on page 140
  To help ensure that logs are pruned on a regular basis to conserve system space, you can specify logrotate rules.

- **Configure sudo without a password** on page 141
  You can give users and user groups the ability to run some or all commands as root without a password.

- **Configure user SSH keys** on page 141
  To use the Secure Shell (SSH) protocol for communication between computers, you must configure SSH keys. You can also configure SSH keys for individual users.

- **Configure SSH permissions for users and groups** on page 141
  You can configure Secure Shell (SSH) protocol settings at a granular level to specify permissions for users and groups.

- **Configure the firewall type** on page 142
  To help protect your infrastructure, CEM enforces a firewall solution, `firewalld`, by default. `firewalld` is zone-based software that is designed to monitor traffic and take appropriate action. To change the firewall type or specify that CEM does not manage a firewall, you can update the firewall configuration.

- **Configure rules that rely on site-specific information** on page 142
  Some Center for Internet Security (CIS) rules require information that is specific to a customer site. You can use Bolt tasks to configure these rules.

**Basic configuration example**

When you specify a compliance framework, CEM is configured to provide rule enforcement and configuration for that framework. For example, to enforce the Center for Internet Security (CIS) Server Level 1 benchmark for a node, you must classify the node with the `cem_linux` class, set the `benchmark` parameter to `cis`, and run Puppet.

In the following example, CEM enforces the CIS Level 1 server recommendations Ensure AIDE is installed and Ensure filesystem integrity is regularly checked on a CentOS 7 node:

1. Add the following Hiera data to your control repository, control repo:

   ```yaml
   # control-repo/data/nodes/<node name>.yaml
   cem_linux::benchmark: 'cis'
   cem_linux::config:
     profile: 'server'
     level: '1'
     only:
       - 'ensure_aide_is_installed'
       - 'ensure_filesystem_integrity_is_regularly_checked'
   
   2. Classify the node with the `cem_linux` class.
   3. Run Puppet.
Some CIS recommendations require you to run a Bolt task. To determine which task to run, review the output of the Puppet debug logs.

**Advanced configuration example**

Building on the basic configuration example, the advanced configuration example customizes the Advanced Intrusion Detection Environment (AIDE) configuration file in Hiera.

1. Add the following code to the node’s Hiera file:

```yaml
# control-repo/data/nodes/<node name>.yaml
cem_linux::benchmark: 'cis'
cem_linux::config:
  profile: 'server'
  level: '1'
  only:
    - 'ensure_aide_is_installed'
    - 'ensure_filesystem_integrity_is_regularly_checked'
  control_configs:
    ensure_aide_is_installed:
      conf_rules:
        - 'PERMS = p+u+g+acl+xattrs'
        - 'CONTENT_EX = sha256+ftype+p+u+g+n+acl+xattrs'
      conf_checks:
        - '/root/\..* PERMS'
        - '/root/ CONTENT_EX'
```

2. Classify the node with the `cem_linux` class.

3. Run Puppet.

4. Run the Bolt task that is specified in the debug log.

The AIDE configuration file now reflects the changes in Hiera.

**Automatically regenerate and enforce bootloader configurations**

In rare cases, it might be useful to enable automatic regeneration of the bootloader configuration, and you might want to set a bootloader password. By setting a bootloader password, you can provide an extra layer of security for your infrastructure.

**Restriction:** The only bootloader supported by CEM for Linux is `grub2`.

CEM for Linux enforces various bootloader configurations as required by the selected compliance framework and benchmark. However, because changes to bootloader configurations can be potentially dangerous, a minimalist approach to configuration changes is used by CEM for Linux.

Several CIS recommendations modify the bootloader config. If you run CEM for Linux with the full range of default settings, these changes will be applied, but the bootloader config will not be regenerated. While changes are pending on the node, bootloader operations remain the same until the configurations are regenerated. The exception to this is the bootloader password, which is **not set** by default. To learn how to configure CEM for Linux to automatically regenerate the bootloader config and set the bootloader password, see the following topics:

- **Regenerate bootloader configs automatically** on page 139
  You can regenerate bootloader configs automatically by editing a Hiera `.yaml` file.

- **Set a bootloader password** on page 140
  You can set a bootloader password by editing a Hiera `.yaml` file.

**Regenerate bootloader configs automatically**

You can regenerate bootloader configs automatically by editing a Hiera `.yaml` file.

Edit the `.yaml` file to specify the `regenerate` setting:

```bash
# control-repo/data/nodes/<node name>.yaml
---
```
Set a bootloader password
You can set a bootloader password by editing a Hiera .yaml file.

You can set a bootloader password as shown:

```yaml
# control-repo/data/nodes/<node name>.yaml
---
cem_linux::regenerate_grub2_config: true
cem_linux::set_grub2_password: true
cem_linux::grub2_superuser: 'root'
cem_linux::grub2_superuser_password: 'password'
lookup_options:
  cem_linux::grub2_superuser_password:
    convert_to: 'Sensitive'
```

Restriction: The `cem_linux::grub2_superuser_password` key must be of type `Sensitive[String]`. Setting a lookup option for that key to convert it to `Sensitive` is the best way to ensure that the value is `Sensitive[String]`.

CAUTION: Do not store plain-text passwords in Hiera. To help protect passwords, use a hierarchy entry such as `hiera-eyaml`.

Configure DISA STIG
The US Defense Information Systems Agency (DISA) has developed Security Technical Implementation Guide (STIG) standards that are designed to secure information systems and software.

If DISA STIG is available for your operating system, you can enable DISA STIG by adding the following Hiera data to your control repository, control-repo:

```yaml
# control-repo/data/nodes/<node name>.yaml
cem_linux::benchmark: 'stig'
```

Configure custom logrotate rules
To help ensure that logs are pruned on a regular basis to conserve system space, you can specify logrotate rules.

The following example creates custom logrotate rules for the primary Puppet server's `puppetserver` logs.

```yaml
# control-repo/data/nodes/<your puppetserver>.yaml
---
cem_linux::config:
  control_configs:
    ensure_logrotate_is_configured:
      rules:
        puppetserver:
          path:
            - '/var/log/puppetlabs/puppetserver/puppetserver.log'
            - '/var/log/puppetlabs/puppetserver/pcp-broker.log'
            - '/var/log/puppetlabs/puppetserver/puppetserver-access.log'
            - '/var/log/puppetlabs/puppetserver/puppetserver-daemon.log'
            - '/var/log/puppetlabs/puppetserver/puppetserver-status.log'
            - '/var/log/puppetlabs/puppetserver/code-manager-access.log'
            - '/var/log/puppetlabs/puppetserver/file-sync-access.log'
            - '/var/log/puppetlabs/puppetserver/masterhttp.log'
          create_owner: 'puppet'
          create_group: 'puppet'
```
Configure sudo without a password
You can give users and user groups the ability to run some or all commands as root without a password.

The following example configures the `admins` group to grant sudo access without a password:

```bash
cem_linux::benchmark: 'cis'
cem_linux::config:
  profile: 'server'
  level: '1'
  control_configs:
    ensure_sudo_is_installed:
      package_ensure: 'installed'
      options:
        user_group:
          %admins:
            options:
              - 'NOPASSWD:'
```

Configure user SSH keys
To use the Secure Shell (SSH) protocol for communication between computers, you must configure SSH keys. You can also configure SSH keys for individual users.

In the following example, keys are configured for `testuser1` and `testuser2`:

```bash
cem_linux::benchmark: 'cis'
cem_linux::config:
  profile: 'server'
  level: '1'
  control_configs:
    ensure_permissions_on_etccesshsshd_config_are_configured:
      permit_root_login: 'yes'
      user_ssh_keys:
        testuser1:
          username: testuser1
          home_dir: /home/testuser1
          ssh_key: ssh-rsa A...ZcTFw== rsa-key-20201022
        testuser2:
          username: testuser2
          home_dir: /home/testuser2
          ssh_key: ssh-rsa A...ZcTFw== rsa-key-20201022
```

Configure SSH permissions for users and groups
You can configure Secure Shell (SSH) protocol settings at a granular level to specify permissions for users and groups.

The following example configures SSH to grant permissions to some users and groups and deny permissions to other users and groups:

```bash
cem_linux::benchmark: 'cis'
cem_linux::config:
  control_configs:
    ensure_permissions_on_etccesshsshd_config_are_configured:
      allow_users:
        - testuser1
        - the_dude
      allow_groups:
        - testgroup1
        - goonies
      deny_users:
        - testuser2
        - the_emperor
      deny_groups:
```

© 2022 Puppet, Inc., a Perforce company
Configure the firewall type

To help protect your infrastructure, CEM enforces a firewall solution, `firewalld`, by default. `firewalld` is zone-based software that is designed to monitor traffic and take appropriate action. To change the firewall type or specify that CEM does not manage a firewall, you can update the firewall configuration.

Restriction: Firewalls that are based on the `nftables` framework are not supported. Use the `firewalld` or `iptables` setting instead.

The following examples show how to configure a firewall type.

The default setting is `firewalld`:

```yaml
cem_linux::benchmark: 'cis'
cem_linux::config:
  profile: 'server'
  level: '1'
  firewall_type: 'firewalld'
```

You can also specify a value of `iptables`:

```yaml
cem_linux::benchmark: 'cis'
cem_linux::config:
  profile: 'server'
  level: '1'
  firewall_type: 'iptables'
```

You can also specify a value of `unmanaged`. If you specify `unmanaged`, CEM does not enforce a state on any firewall resource:

```yaml
cem_linux::benchmark: 'cis'
cem_linux::config:
  profile: 'server'
  level: '1'
  firewall_type: 'unmanaged'
```

Configure rules that rely on site-specific information

Some Center for Internet Security (CIS) rules require information that is specific to a customer site. You can use Bolt tasks to configure these rules.

By using Puppet Enterprise (PE), you can run Bolt tasks and plans to audit or configure specific parts of a node. To run Bolt tasks, open the PE console and select the **Tasks** menu. Then, select **cem_linux**.

You can also run Bolt tasks from the command line:

1. Install Puppet Development Kit (PDK) and Bolt.
2. In the root of the CEM directory, run the `pdk bundle exec rake 'spec_prep'` command. This command downloads the required dependencies as RSpec fixtures, and then creates a symbolic link from the module directory to the fixtures directory.
3. Run the tasks on one or more hosts. For example:

   ```bash
   bolt task run 
   comply_enforcement_module::audit_unowned_files_and_directories -t $nodefqdn --modulepath spec/fixtures/modules
   ```

   You must add the `--modulepath spec/fixtures/modules` option to Bolt commands. Otherwise, Bolt is not able to find the tasks and plans.
Auditing and querying issues identified during scans

In some cases, a CIS compliance scan might identify an issue that you want to investigate and fix. To get started, you can run an audit or query.

CEM for Linux provides tasks that you can use to run an audit or query. For more information, go to Puppet Forge and review the Tasks list.

**Tip:** In the Tasks list, you can click the green downward arrow to display any parameters that are associated with a task.

Reference: Benchmarks and controls

For help with configuring CEM, review the Reference topics on Puppet Forge.

In the CEM Linux Reference, locate the relevant CIS Benchmark and then the control.

CEM for Windows

You can deploy CEM for Windows to help ensure that your servers and workstations on Microsoft Windows operating systems comply with CIS Benchmarks.

By default, CEM enforces the Level 1 server profile on Windows Server 2016 and Windows Server 2019. CEM enforces the Level 1 corporate enterprise profile on Windows 10 Enterprise. For more information about CIS Benchmarks, see Center for Internet Security (CIS) rules.

To take advantage of new features, fixes, and improvements, install the latest version of CEM. You can learn about the latest release by reviewing the Release notes on page 143. Then, to install CEM, follow the instructions in Installing CEM on page 149. By default, CEM runs automatically on any classified nodes and does not require configuration. However, if you want to configure CEM to meet your organization's requirements, follow the instructions in Configuring CEM on page 151.

- **Release notes** on page 143
  Review the release notes to learn about updates and resolved issues in the Compliance Enforcement Module (CEM) for Windows.
  - **Installing CEM** on page 149
    Before you install CEM, complete the preparation steps: review the system requirements, install and configure Puppet Enterprise (PE), and purchase CEM. Then, install CEM and classify the nodes on which you want to enforce compliance.
  - **Upgrading CEM** on page 151
    You can upgrade CEM for Windows to take advantage of the latest features, fixes, and improvements.
  - **Configuring CEM** on page 151
    Configuration of CEM is optional. If you installed CEM and assigned the cem_windows class to one or more node groups in the Puppet Enterprise (PE) console, PE will run automatically and enforce the default compliance profile on the classified nodes. However, if the default values leave your infrastructure in an undesirable state, or if you want to customize compliance to meet your organization's requirements, you can configure CEM.
  - **Reference: Benchmarks and controls** on page 157
    For help with configuring CEM, review the Reference topics on Puppet Forge.

Release notes

Review the release notes to learn about updates and resolved issues in the Compliance Enforcement Module (CEM) for Windows.
v1.2.3
Released 25 October 2022

• **Added**
  • Added a Puppet Bolt task, `cem_delete_securitypolicy_inf`, to use for error resolution. The Bolt task resolves a corruption error that can affect the temporary file that is used by Desired State Configuration (DSC) to manage the local security policy:
    • The error is indicated by the following message in the Puppet run log:
      ```
      Index operation failed; the array index evaluated to null
      ```
    • To resolve the error, run the `cem_delete_securitypolicy_inf` task and re-run Puppet on the affected node.

• **Changed**
  • The product documentation was revised to improve usability and retrievability:
    • The change log was migrated from Puppet Forge to the central location for Puppet documentation, Puppet Docs. The change log was renamed to **Release notes** on page 143.
    • The readme file was transformed into a series of topics with a structure similar to other Puppet documentation. The CEM topics are now available on Puppet Docs, starting with **Introducing the Compliance Enforcement Modules** on page 125.
    • The **Reference** and **Dependencies** documentation, which is generated automatically, remains on Puppet Forge.

• **Fixed**
  • Fixed an error that prevented catalog retrieval from Puppet Enterprise (PE) during Continuous Delivery for Puppet Enterprise pipeline runs. This error occurred when the impact analysis tool was used to set up a temporary environment, which was then deleted. The `_FILE_` variable continued to point to the deleted environment. As a result, the Puppet run returned an error message: **Could not retrieve catalog from remote server.**

v1.2.2
Released 10 August 2022

**Fixed**
• Fixed typos in Microsoft Windows firewall logging paths managed by the following controls:
  • **CIS Windows 10**
    • 9.1.5
    • 9.2.5
    • 9.3.7
  • **CIS Windows Server 2016**
    • 9.1.5
    • 9.2.5
    • 9.3.7
  • **CIS Windows Server 2019**
    • 9.1.5
    • 9.2.5
    • 9.3.7
• Fixed an issue that could cause the following controls to not be enforced:
  • CIS Windows 10
    • 18.9.17.2
    • 18.9.64.1
    • 18.9.65.3.10.1
    • 18.9.65.3.10.2
    • 18.9.65.3.2.1
    • 18.9.72.1
    • 18.9.75.1
    • 18.9.103.1
  • CIS Windows Server 2016
    • 18.9.45.10.1
  • CIS Windows Server 2019
    • 18.9.41.1
    • 18.9.45.1
    • 18.9.47.11.1
    • 18.9.65.3.10.1
    • 18.9.65.3.10.2
    • 18.9.65.3.2.1
    • 18.9.65.3.3.1
    • 18.9.65.3.3.3
    • 18.9.65.3.3.4
    • 18.9.67.2
    • 18.9.72.1
    • 18.9.89.1
    • 18.9.90.3
    • 18.9.102.2.2
    • 18.9.103.1
    • 18.9.47.5.1.2

v1.2.1
Released 31 May 2022

Fixed
  • Fixed a bug related to profile configuration on Microsoft Windows 10 nodes.

v1.2.0
Released 24 May 2022

• Changed
  • Updated the Center for Internet Security (CIS) Windows Server 2019 Benchmark to version 1.3.0.
• Fixed
  • Resolved issues leading to scan failures for the following CIS controls on Windows Server 2019:
    • 9.3.7
    • 9.2.5
    • 9.1.5
    • 18.9.108.4.1
    • 18.9.65.3.9.1
    • 18.8.3.1
    • 18.8.21.5
    • 18.5.21.1
    • 18.4.x
    • 18.2.1

v1.1.2
Released 12 May 2022
• Changed
  • Updated the minimum required version of the dsc/auditpolicydsc module to 1.4.0-0-4. That dependency contains bug fixes and features required by cem_windows. Update your Puppetfile accordingly.
• Fixed
  • Updated the default value for the Windows Attack Surface Reduction (ASR) rules to Audit instead of Block.
    • While the value of Audit is not CIS-compliant, setting the ASR rules to Block prevented the Puppet agent from successfully configuring the node.
    • If you see Puppet run errors like Could not evaluate: undefined method []' for nil:NilClass when enforcing CEM, manually set the Windows ASR rules to Audit. To learn more about Windows ASR rules, see Attack surface reduction rules overview.
  • Fixed an issue that applied more controls to a node than required by the configured profile and level.
  • Fixed an issue that caused controls that should be ignored to be applied. This issue occurred when the controls were mapped to a parameter of a resource that was not ignored.
  • Fixed several issues related to configuration backward-compatibility.

| Upgrade requirement: | To ensure that the updates in this release take effect, you might have to restart the pe- puppetserver service on your Puppet primary server after Code Manager deploys the new code. |

v1.1.1
Released 7 April 2022
• Changed
  • Improved the display of controls in the CEM Windows Reference.
• Fixed
  • Fixed several instances in which configurations from versions previous to v1.1.0 were not recognized. The v1.1.1 configuration is backward compatible with versions prior to v1.1.0.
  • Fixed an issue that required the cem_windows module to exist in the same environment as the Puppet primary server. You can now deploy the module to a different environment than your primary server. The module will be operational.
  • Fixed incorrect Puppet Strings in init.pp file.
**v1.1.0**

Released 24 March 2022

- **Added**
  - The documentation was updated to list the controls that will be reported as failed or unknown in Comply after `cem_windows` is applied.
  
  **Tip**: A failed or unknown status is reported because the CIS-CAT Pro Assessor looks for registry keys that are configured by Microsoft Group Policy Objects rather than keys that are set locally by the `cem_windows` user. The CIS Windows benchmarks are designed to work only for domain-joined systems. At the time of the v1.1.0 release, CIS was working on Windows benchmarks for a standalone system to resolve the issue.

- **Changed**
  - Updated the CIS Windows 10 Benchmark to v1.12.0 to match the latest benchmark version released with Comply 2.4.0.
  - The `cem_windows` module was updated to implement a new architecture. The new architecture, applied in the background, provides more flexibility for system configuration. For details, see the readme file.

**v1.0.7**

Released 16 December 2021

- **Removed**
  - Removed unnecessary resource defaults in two Windows Server 2016 control classes.

**v1.0.6**

Released 16 December 2021

- **Removed**
  - Removed unnecessary resource defaults in Windows Server 2016 control classes.

**v1.0.5**

Released 8 December 2021

- **Fixed**
  - Fixed non-idempotent Desired State Configuration (DSC) resources.
  - Fixed the registry key for Windows 10 CIS control 1.1.6. Now, this control will be properly configured.

**v1.0.4**

Released 7 December 2021

- **Added**
  - In the readme file, added a link to premium content installation instructions. To use CEM, you must be a premium content subscriber.

- **Fixed**
  - Fixed an issue that caused values for the `dsc_accountpolicy` parameter to be set incorrectly.

**v1.0.3**

Released 13 October 2021
• Fixed
  • Fixed the default value for CIS control 2.3.1.1 to align with the expected value provided by CIS.
  • Fixed the cem_windows::allow_local_account_rdp parameter so that it works as intended.

v1.0.2
Released 11 October 2021
• Fixed
  • Fixed firewall profiles to align with the CIS specification.

v1.0.1
Released 30 September 2021
• Fixed
  • Fixed the Windows 10 Hiera name to ensure that Windows 10 can be used. For more information about Hiera, see Configure settings with Hiera.

Known issues and limitations
The current release includes known issues and limitations. In most cases, workarounds are provided.

• After an upgrade, you might have to restart Puppet Server or the pe-puppetserver service. Starting with v1.1.0, CEM for Windows implements a new architecture. If you upgrade CEM from v1.0.7 or earlier to v1.1.0 or later, and you encounter errors, try restarting the pe-puppetserver service or restarting or reloading Puppet Server. For instructions, see Restarting Puppet Server.

• You might have to manually set Windows Attack Surface Reduction (ASR) rules to Audit. In cem_windows releases prior to v1.1.2, a default value of Block was set in the module to comply with CIS guidelines. However, the Block value prevented the Puppet agent from successfully configuring the node. For this reason, the default value was changed to Audit, which is not CIS compliant. If you see Puppet run errors like Could not evaluate: undefined method []' for nil:NilClass when enforcing CEM, manually set the Windows ASR rules to Audit. To learn more about Windows ASR rules, see Attack surface reduction rules overview.

• Some controls can fail scans. During a Comply scan, you might see error messages about CIS recommended guidelines that are not enforced. These error messages are triggered by bugs in the CIS-CAT Pro Assessor that is bundled with Comply. CEM correctly enforces these settings. The following controls are affected:
  • 1.1.5 - Windows Server 2016 and Windows Server 2019
  • 1.1.6 - Windows Server 2016 and Windows Server 2019
  • 2.3.10.7 - Windows Server 2016
  • 18.2.1 - Windows Server 2019
  • 18.4.1 - Windows Server 2016 and Windows Server 2019
  • 18.4.8 - Windows Server 2016
  • 18.4.9 - Windows Server 2016 and Windows Server 2019
  • 18.4.12 - Windows Server 2016
  • 18.8.21.5 - Windows Server 2016
  • 18.9.47.5.1.2 - Windows Server 2019
  • 18.9.62.3.9.1 - Windows Server 2016

• Puppet runs are not idempotent. If you see Desired State Configuration (DSC) resources showing corrective changes in a Puppet run, for example, Unknown feature "custom_isync", you are running an incompatible version of Puppet. CEM for Windows requires that Puppet agents at the version 6 level must be v6.23.0 or later, and agents at the version 7 level must be v7.8.0 or later.

• If the Puppet agent fails to upgrade when you use the puppetlabs/puppet_agent module, restart the computer or virtual machine where the Puppet agent is running to help ensure that updates are applied.
• If you use remote desktop protocol (RDP) to access nodes, users who are members of the groups Guests and local accounts will not be able to log in by default. To provide access to these groups, set the cem_windows::allow_local_account_rdp parameter to true.

• If non-admin users cannot log in to nodes, the issue might be related to event logs. By default, Windows Event Log does not clear events. When the event log of a node is full, only administrators can log in. To clear the event logs manually, find the specific recommendation in your compliance framework and configure the setting. In the Windows registry, locate the following key:

\HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\EventLog\Application:Retention

Then, set the Retention value to 0.

• You cannot disable Windows Remote Management (WinRM). The WinRM service is required for the DSC modules and cannot be disabled.

Controls ignored by default to prevent operational issues
Some controls are ignored by default to prevent operational issues. However, you can enable the controls if necessary.

• 2.3.1.1 (Ensure 'Accounts: Administrator account status' is set to 'Disabled') - If this control is applied, it can cause non-idempotent runs. The control can also cause Puppet run failures if you attempt to run Puppet manually while logged in as Administrator.

• 2.3.1.5 (Configure 'Accounts: Rename administrator account') - If this control is applied, it can cause non-idempotent runs. The control can also cause Puppet run failures if you attempt to run Puppet manually while logged in as Administrator.

To enable controls ignored by default, create an ignore config that doesn’t include the controls. For example, the following configuration ignores control 1.1.1, thus overriding the default ignore list:

cem_windows::config:
  ignore:
    - 'c1_1_1'

The following configuration removes all controls from the ignore list:

cem_windows::config:
  ignore: []

Installing CEM
Before you install CEM, complete the preparation steps: review the system requirements, install and configure Puppet Enterprise (PE), and purchase CEM. Then, install CEM and classify the nodes on which you want to enforce compliance.

• Prepare to install the module on page 149
To help ensure the successful deployment of CEM, complete the preparation steps.

• Install the module and classify nodes on page 150
To deploy CEM, you must install the module and then classify the nodes on which you want to enforce Center for Internet Security (CIS) compliance.

• Uninstall the module on page 150
To stop using CEM, you can uninstall the cem_windows module. Alternatively, to stop using CEM on one or more nodes, you can declassify the nodes to remove their association with the cem_windows class.

Prepare to install the module
To help ensure the successful deployment of CEM, complete the preparation steps.
1. Review the following table to ensure that CEM can meet your organization’s requirements. `cem_windows` supports the following operating systems and CIS benchmarks:

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Framework</th>
<th>Level</th>
<th>Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2019</td>
<td>CIS Benchmarks v1.3.0</td>
<td>1</td>
<td>Member Server</td>
</tr>
<tr>
<td>Windows Server 2016</td>
<td>CIS Benchmarks v1.3.0</td>
<td>1</td>
<td>Member Server</td>
</tr>
<tr>
<td>Windows 10 Enterprise</td>
<td>CIS Benchmarks v1.12.0</td>
<td>1</td>
<td>Corporate Enterprise</td>
</tr>
</tbody>
</table>

**Tip:** CEM uses Desired State Configuration (DSC) modules and the `validation_mode` parameter to ensure that resources do not remain in a "flapping" state. For more information, see `securitypolicydsc`.

2. Install Puppet Enterprise (PE). To use PE at a 2019 level, install version 2019.8.4 or later. To use PE at a 2021 level, install 2021.2.0 or later. For instructions, see `Installing`.

3. Review the dependencies to ensure that your infrastructure will meet the requirements. Go to Puppet Forge and review the `Dependencies` tab.

4. Configure PE. For instructions, see `Configuring Puppet Enterprise`.

5. Ensure that the Puppet utility is at the correct level. If you are using a level 6 utility, it must be v6.23.0 or later. If you are using a level 7 utility, it must be v7.8.0 or later.

6. To purchase CEM software, contact a Puppet by Perforce sales representative. For more information, see `Contact Sales`.

### Install the module and classify nodes

To deploy CEM, you must install the module and then classify the nodes on which you want to enforce Center for Internet Security (CIS) compliance.

**Before you begin**

In some cases, compliance controls can negatively impact services that run on nodes. To help avoid possible issues, you can install and evaluate CEM in a test environment before running CEM in a production environment.

1. Download CEM from Puppet Forge. The module is available as a premium content subscription. For more information, see the `Premium content page`.

2. Install the module by using Code Manager, a code management tool. For instructions, see the `Premium content page`.

3. Specify the nodes on which you want Puppet Enterprise (PE) to run and enforce compliance. To specify the nodes, open the PE console and assign the `cem_windows` class to one or more node groups. For instructions about creating and classifying node groups, see `Grouping and classifying nodes`.

### Uninstall the module

To stop using CEM, you can uninstall the `cem_windows` module. Alternatively, to stop using CEM on one or more nodes, you can declassify the nodes to remove their association with the `cem_windows` class.

**CAUTION:** If you uninstall CEM or declassify one or more nodes, Puppet Enterprise (PE) no longer runs on the affected nodes to enforce Center for Internet Security (CIS) compliance. When you uninstall CEM or declassify a node, the node does not revert to its pre-CEM state.

Take one of the following actions:

- To uninstall the CEM module, follow the instructions in `Uninstalling modules`.
- To declassify nodes, you can remove the nodes from the node group that is associated with the `cem_windows` class. For instructions, see `Remove nodes from a node group`. Alternatively, to stop using CEM on an entire node group, you can remove the `cem_windows` class from the node group. For instructions, see `Remove classes from a node group`.

© 2022 Puppet, Inc., a Perforce company
Upgrading CEM

You can upgrade CEM for Windows to take advantage of the latest features, fixes, and improvements.

To upgrade CEM, update the CEM declaration in the Puppetfile. Specify the CEM version number to which you are upgrading.

For example, to upgrade cem_windows to version 1.2.2, specify the CEM declaration as shown:

```
mod 'puppetlabs/cem_windows', '1.2.2'
```

For instructions about modifying the Puppetfile, see Declare Forge modules in the Puppetfile.

**Troubleshooting tip:** Starting with v1.1.0, CEM for Windows implements a new architecture. If you upgrade CEM from v1.0.7 or earlier to v1.1.0 or later, and you encounter errors, try restarting the `pe-puppetserver` service or restarting or reloading Puppet Server. For instructions, see Restarting Puppet Server.

Configuring CEM

Configuration of CEM is optional. If you installed CEM and assigned the cem_windows class to one or more node groups in the Puppet Enterprise (PE) console, PE will run automatically and enforce the default compliance profile on the classified nodes. However, if the default values leave your infrastructure in an undesirable state, or if you want to customize compliance to meet your organization's requirements, you can configure CEM.

By default, CEM for Windows enforces the Level 1 Member Server profile on classified Windows Server 2016 and Windows Server 2019 nodes, and the Level 1 Corporate Enterprise profile on classified Windows 10 Enterprise nodes.

You can customize the settings. For example, if a CIS control sets the maximum password age at 365 days, but your organization requires a password change every 90 days, you can configure CEM accordingly.

You configure CEM by using the Hiera tool in your control repository. For more information, see About Hiera and Getting started with Hiera.

For general information about configuration options, see Overview of configuration options on page 151.

For detailed information about configuration options, see the Reference: Benchmarks and controls on page 157.

For configuration examples, see How to configure the module: Examples and guidelines on page 153.

**CAUTION:** CEM default settings are fully CIS compliant. Too much customization can cause your configuration to become non-compliant.

- Overview of configuration options on page 151
- How to configure the module: Examples and guidelines on page 153

The following examples demonstrate the use of CEM in a production environment.

Overview of configuration options

Configuration options include top-level options, framework options, and Center for Internet Security (CIS)-specific options.

If you installed CEM and assigned the cem_windows class to a node group in the PE console, the default profile is automatically enforced. However, to customize CEM to meet your organization's requirements, you can configure benchmarks by using the Hiera tool. For more information, see Hiera.
CAUTION: CEM’s default settings are fully CIS compliant. Too much customization can cause your configurations to be noncompliant.

Controls ignored by default: Controls 2.3.1.1 and 2.3.1.5 are ignored by default because of non-idempotent and Puppet run issues. For details and configuration options, see Controls ignored by default to prevent operational issues on page 149.

Find and set configuration options

You can find the configuration options for a specific control in the Reference: Benchmarks and controls on page 157. The reference is divided into sections, with each section representing a benchmark. In those benchmarks, you will see each control listed with several subsections:

- **Parameters**: Configuration options for a control, along with the data type and default value.
- **Supported Levels**: The supported levels for a CIS control.
- **Supported Profiles**: The supported profiles for a CIS control.
- **Hiera Configuration Example**: Snippet of Hiera that can be used to configure a control.
- **Alternate Config IDs**: The alternate config IDs for a control. Any of these config IDs, along with the full control name, can be used as a key in the control_config hash.
- **Resource**: The name of the Puppet resource that enforces the control.

Guidelines for specifying config IDs

You can specify controls in the control_config hash by referencing the full control name, the control number, the normalized control name, or the normalized control number. You cannot mix and match these forms and must pick a single config ID form to use for your config. Full control names and control numbers are copied verbatim from the benchmarks and are case-sensitive. Normalized control names have lowercase letters and contain only alphanumeric characters and underscores. Normalized control numbers are always prefixed with a c and contain only numeric characters separated by underscores.

Example of alternate config IDs:

- Full control name: (L1) Ensure 'Enforce password history' is set to '24 or more password(s)'
- Control number: 1.1.1
- Normalized control name: ensure_enforce_password_history_is_set_to_24_or_more_passwords
- Normalized number: c1_1_1

Top-level configuration options

These options are configured at the top level of the module.

In Hiera, these options are prefixed with cem_windows:

- **framework**: Enum['cis'] - the compliance framework to use. CEM supports only cis. Default: cis.
- **config**: Optional[Hash] - the location for all non-top-level configuration options. Default: undef.
- **allow_on_domain_controller**: Boolean - If cem_windows detects that it is running on a domain controller, CEM does not enforce controls and logs a warning to inform the user. In this way, CEM helps to prevent the enforcement of compliance settings on domain controllers that could negatively impact an entire domain. Default: false.
- **enable_long_paths**: Boolean - Enables support for long path names in the Windows registry. Setting this option to false can cause issues with some Desired State Configuration (DSC) modules used in cem_windows. Default: true.
- **privileged_user**: Optional[String] - If the Puppet agent does not run under a user with local administrator privileges, you must supply the name of a user with local administrator privileges. This is used by DSC to enforce a state on a machine. Default: undef.
• privileged_password - Sensitive[Any] - If you specified a privileged user, use this option to specify a password for that user account. Default: undef.

• allow_local_account_rdp - Boolean - By default, cem_windows disables remote desktop protocol (RDP) access for non-domain accounts. If you set this option to true, local accounts on the node can make RDP connections to the node. Default: false.

Framework configuration options
The framework configuration options are available as key-value pairs within the cem_windows::config::hash.

• control_configs - Optional[Hash] — location for all rule-specific configurations. Default: undef.

• only - Optional[Array[String]] — takes an array of control class names (manifests/benchmarks/<benchmark>/controls/*.pp). The classes specified here are included in the catalog. Takes precedence over the ignore: option. Default: undef.

• ignore - Optional[Array[String]] — takes an array of control class names (manifests/benchmarks/<benchmark>/controls/*.pp). The classes specified here are not included in the catalog. If only: is specified, this option has no effect. Default: undef.

CIS-specific configuration options
The CIS-specific configuration options are available as key-value pairs within the cem_windows::config::hash. These options are applicable only to the CIS compliance framework.

• profile - Optional[Enum['member_server', 'corporateenterprise']]) — the name of the benchmark profile. corporateenterprise is supported only on Windows 10 Enterprise operating systems. Default for Windows Server operating systems: member_server. Default for Windows 10 Enterprise operating systems: corporate_enterprise.

• level - Optional[Enum['1', '2']]) — the name of the profile level. The only value supported by CEM is 1. Default: 1.

For more details about configuration options, see Reference: Benchmarks and controls on page 157.

How to configure the module: Examples and guidelines
The following examples demonstrate the use of CEM in a production environment.

• Basic configuration examples on page 154
When you specify a compliance framework, CEM is configured to provide rule enforcement and configuration for that framework. For example, to enforce the Center for Internet Security (CIS) Server Level 1 benchmark for a node, you must classify the node with the cem_windows class, set the benchmark parameter to cis, and run Puppet.

To learn more about CEM configuration, see the following examples.

• Advanced configuration example on page 154
Building on the basic configuration examples, the control_configs section specifies advanced options for controls.

• Run Desired State Configuration resources as a specific user on page 155
Desired State Configuration (DSC) requires local administrator privileges to modify Windows resources. Typically, the Puppet agent runs under a user account with these permissions. However, if the Puppet agent on a node does not have local administrator permissions, you can use Hiera to configure a user account that does have the required permissions.

• Allow local accounts to access nodes on page 155
To allow a local user account to access a node with RDP, set the top-level option allow_local_account_rdp to true.

• Enforce specific rules on page 155
To configure CEM to enforce only specific rules, use the only key.

• Ignore specific rules on page 155
To configure CEM to ignore specific rules, use the ignore key.

• Customize rules on page 156
You can customize most rules by using the control_configs key and supplying the key with a hash value.

• Rename the Administrator and Guest accounts on page 156
To help protect your infrastructure, rename the Administrator and Guest accounts.
Basic configuration examples
When you specify a compliance framework, CEM is configured to provide rule enforcement and configuration for that framework. For example, to enforce the Center for Internet Security (CIS) Server Level 1 benchmark for a node, you must classify the node with the `cem_windows` class, set the `benchmark` parameter to `cis`, and run Puppet. To learn more about CEM configuration, see the following examples.

Example 1
In the following example, CEM applies only the following controls on a Windows 10 node: 'c1_1_1' and 'c2_3_1_1'.

1. Add the following Hiera data to your control repository, `control_repo`:

   ```yaml
   # control-repo/data/nodes/<node name>.yaml
   cem_windows::benchmark: 'cis'
cem_windows::config:
   profile: 'server'
   level: '1'
   only:
   - 'c1_1_1'
   - 'c2_3_1_1'
   
2. Classify the node with the `cem_windows` class.
3. Run Puppet.

Example 2
In the following example, CEM applies all controls in the configured benchmark and profile on a Windows 10 node except for the following controls: 'c1_1_1' and 'c2_3_1_1'.

1. Add the following Hiera data to your control repository, `control_repo`:

   ```yaml
   # control-repo/data/nodes/<node name>.yaml
   cem_windows::benchmark: 'cis'
cem_windows::config:
   profile: 'server'
   level: '1'
   ignore:
   - 'c1_1_1'
   - 'c2_3_1_1'
   
2. Classify the node with the `cem_windows` class.
3. Run Puppet.

Advanced configuration example
Building on the basic configuration examples, the `control_configs` section specifies advanced options for controls.

1. Add the following code to the node's Hiera file:

   ```yaml
   cem_windows::benchmark: 'cis'
cem_windows::config:
   profile: 'server'
   level: '1'
   only:
   - 'c1_1_1'
   - 'c2_3_1_1'
   - 'c2_3_7_2'
   - 'c2_3_1_3'
   - 'c2_3_1_4'
   - 'c2_3_1_5'
   - 'c2_3_1_6'
   ```
2. Classify the node with the `cem_windows` class.
3. Run Puppet.

**Run Desired State Configuration resources as a specific user**

Desired State Configuration (DSC) requires local administrator privileges to modify Windows resources. Typically, the Puppet agent runs under a user account with these permissions. However, if the Puppet agent on a node does not have local administrator permissions, you can use Hiera to configure a user account that does have the required permissions.

Use a configuration based on the following structure:

```
# control-repo/data/nodes/winserv2019.contoso.com.yaml
---
cem_windows::privileged_user: <user name>
cem_windows::privileged_pass: <user password>
```

**Allow local accounts to access nodes**

To allow a local user account to access a node with RDP, set the top-level option `allow_local_account_rdp` to `true`.

Use a configuration based on the following structure:

```
# control-repo/data/nodes/winserv2019.contoso.com.yaml
---
cem_windows::allow_local_account_rdp: true
```

**Enforce specific rules**

To configure CEM to enforce only specific rules, use the `only` key.

Use a configuration that is similar to the following example:

```
# control-repo/data/nodes/winserv2019.contoso.com.yaml
---
cem_windows::framework: 'cis'
cem_windows::config:
  profile: 'member_server'
  level: '1'
  only:
    - 'c18_9_97_1_1'
    - 'c18_9_97_1_2'
```

**Ignore specific rules**

To configure CEM to ignore specific rules, use the `ignore` key.

Use a configuration that is similar to the following example:

```
# control-repo/data/nodes/winserv2019.contoso.com.yaml
---
cem_windows::framework: 'cis'
cem_windows::config:
  profile: 'member_server'
  level: '1'
```
**Ignore:**
- 'c18_9_97_1_1'
- 'c18_9_97_1_2'

**Restriction:** The only key and the ignore key are mutually exclusive, with only taking precedence. If you specify both keys, CEM does not use the value of the ignore key and enforces only the rules specified with only.

**Customize rules**
You can customize most rules by using the control_configs key and supplying the key with a hash value.

To customize rules, use a configuration based on the following structure:

```yaml
<recommendation name>:
  <recommendation param>: <value>
```

For example, to configure the rules used in the previous examples, the configuration would look like this:

```yaml
# control-repo/data/nodes/winserv2019.contoso.com.yaml
---
cem_windows::framework: 'cis'
cem_windows::config:
  profile: 'member_server'
  level: '1'
  control_configs:
    c18_9_97_1_1:
      allowbasic: '0'
    c18_9_97_1_2:
      allowunencryptedtraffic: '0'
```

**Rename the Administrator and Guest accounts**
To help protect your infrastructure, rename the Administrator and Guest accounts.

1. To rename the local Administrator account to user_1, use the following configuration:

```yaml
# control-repo/data/nodes/winserv2019.contoso.com.yaml
---
cem_windows::config:
  control_configs:
    c2_3_1_5:
      value: 'user_1'
```

**Restriction:** If you do not specify a name for the Administrator account, the account is renamed to magic by default.

2. To rename the local Guest account to user_2, use the following configuration:

```yaml
# control-repo/data/nodes/winserv2019.contoso.com.yaml
---
cem_windows::config:
  control_configs:
    c2_3_1_6:
      value: 'user_2'
```

**Restriction:** If you do not specify a name for the Guest account, the account is renamed to pumpkin by default.
Reference: Benchmarks and controls

For help with configuring CEM, review the Reference topics on Puppet Forge.

In the CEM Windows Reference, locate the relevant CIS Benchmark and then the control.