Remediate
Contents

Welcome to Puppet Remediate™ .......................................................... 4

Release notes ..................................................................................... 4

Installing Remediate .......................................................................... 6
  Architecture ..................................................................................... 7
  System requirements ...................................................................... 10
  Install Remediate on online nodes ............................................... 11
  Install Remediate on offline nodes .............................................. 14
  Install Remediate from a custom container registry .................. 16
  Install Remediate in multi-network deployments ..................... 17
  Verify Docker Compose file for online installs ......................... 18
  Analytics data collection ............................................................... 18
  Uninstall Remediate ................................................................. 19

Upgrading Remediate ...................................................................... 19
  Upgrade Remediate on online nodes ......................................... 20
  Upgrade Remediate on offline nodes ...................................... 20

Managing user access ...................................................................... 21
  Add new users ............................................................................. 23
  Update a user ............................................................................. 23
  Delete a user ............................................................................. 24
  Force log out ............................................................................. 24
  LDAP configuration ................................................................... 24
  Active Directory configuration ................................................. 25
  LDAP mappers .......................................................................... 27
  Work with user groups in Remediate ....................................... 28

Remediate Vulnerability Dashboard .............................................. 32
  Dashboard Metrics bar ................................................................. 32
  Dashboard Vulnerability Overview tab ..................................... 33
  Dashboard Node Overview tab .................................................. 33

Configuring Remediate .................................................................. 34

Add sources ..................................................................................... 41

Adding node credentials .................................................................. 46
  Secure Shell (SSH) ................................................................. 46
  Add SSH private key files ....................................................... 47
Add SSH username and password......................................................................................................... 47
Windows Remote Management (WinRM)............................................................................................... 47
Add WinRM credentials .......................................................................................................................... 48

Prioritizing vulnerabilities.................................................................................................................... 48
Viewing vulnerability details .................................................................................................................. 49
Filtering and exporting data .................................................................................................................. 50
Create custom filters .............................................................................................................................. 50
Customize table views ............................................................................................................................ 51
Export data ........................................................................................................................................... 51

Remediating vulnerabilities.................................................................................................................. 51
Upload scripts ......................................................................................................................................... 51
Upload modules ..................................................................................................................................... 52
Run tasks ............................................................................................................................................... 52
Installing Puppet agents .................................................................................................................... 53
Running shell commands ................................................................................................................. 53
Managing packages ............................................................................................................................. 54
Managing system services .................................................................................................................. 55
Remediate your top vulnerabilities ........................................................................................................ 56
Remediate high risk vulnerabilities ........................................................................................................ 57
Using tasks tutorial ............................................................................................................................... 57

Discovering and managing resources.................................................................................................. 58
Discovering resources ............................................................................................................................. 58
Viewing resource details ....................................................................................................................... 60
Node attributes ..................................................................................................................................... 60
Package attributes .............................................................................................................................. 67
Container attributes ............................................................................................................................ 68
Filtering and exporting data .................................................................................................................. 68
Create custom filters ............................................................................................................................. 68
Customize table views .......................................................................................................................... 69
Export data ........................................................................................................................................... 69

Review recent events ......................................................................................................................... 69
Discovery events ................................................................................................................................. 69
Tasks events .......................................................................................................................................... 70

Integration status ................................................................................................................................. 71

Troubleshooting ................................................................................................................................... 71
Welcome to Puppet Remediate™

Remediate helps you minimize the risk of external attacks and data breaches by providing you visibility into the vulnerabilities across your infrastructure, and the tools to prioritize and take action. With Remediate, you can eliminate the repetitive and error-prone steps of manual data handovers between teams.

Remediate has the following capabilities built-in:

- Shared vulnerability data: Integrating with Tenable, Qualys and Rapid7.
- Risk-based prioritization: A dashboard that displays your most critical vulnerabilities, prioritized based on infrastructure context.
- Task-based remediation: Allowing you to fix vulnerabilities at scale. You can upload your own scripts for Remediate to convert to a task or use task-based modules from Puppet Forge.

<table>
<thead>
<tr>
<th>Helpful Puppet Remediate docs links</th>
<th>Other useful places</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Before you upgrade or install:</strong></td>
<td>Docs for related Puppet products:</td>
</tr>
<tr>
<td>System requirements</td>
<td>Puppet Enterprise</td>
</tr>
<tr>
<td>Architecture</td>
<td>Bolt</td>
</tr>
<tr>
<td><strong>Getting started:</strong></td>
<td>Open source Puppet</td>
</tr>
<tr>
<td>Install Remediate</td>
<td>Why and how people are using Remediate:</td>
</tr>
<tr>
<td>Manage user access</td>
<td>Read recent blog posts about Remediate</td>
</tr>
<tr>
<td>Add sources</td>
<td>Get support:</td>
</tr>
<tr>
<td>Add node credentials</td>
<td>Search the Support portal and knowledge base</td>
</tr>
<tr>
<td><strong>Prioritize and fix vulnerabilities:</strong></td>
<td>Share and contribute:</td>
</tr>
<tr>
<td>Prioritize vulnerabilities</td>
<td>Engage with the Puppet community</td>
</tr>
<tr>
<td>Remediate vulnerabilities</td>
<td>Puppet Forge</td>
</tr>
<tr>
<td><strong>Discover and manage resources:</strong></td>
<td>Open source projects from Puppet on GitHub</td>
</tr>
<tr>
<td>Discover resources</td>
<td></td>
</tr>
<tr>
<td>Manage resources</td>
<td></td>
</tr>
</tbody>
</table>

Release notes

New features, enhancements, resolved issues, and known issues for Puppet Remediate 1.x release series.

**Version 1.2.1**

Released 23 March 2020
Resolved issues in this release:

- **Service logs timestamps** - RemEDIATE has been updated to fix an issue where timestamps were missing from service logs.
- **Custom registry installs** - A fix was added for an issue that stopped the install image from loading when using a custom container registry.
- **Vault container shutdown issue** - A fix was added to RemEDIATE to resolve an issue that caused the vault container to be inadvertently shut down.
- **Non-UTF-8 encoded status message issue** - RemEDIATE was updated to fix an issue where the controller panicked if it encountered a non-UTF-8 encoded status message.
- **Non-admin login issue** - This release fixes an issue where non-admin accounts were permitted to log in before the initial configuration was complete.

**Version 1.2.0**

Released 26 February 2020

New in this release:

- **Vulnerabilities dashboard redesign** - The RemEDIATE Vulnerabilities dashboard has been redesigned with a new Metrics bar, data visualizations, and reorganization of vulnerability and affected node information.
- **Performance improvements** - The database behind RemEDIATE has been completely redesigned for this release to provide significant performance and scale improvements.
- **Puppet Risk Score** - To enhance risk-based prioritisation, the addition of the Puppet Risk Score (the risk score assigned by your vulnerability scanner multiplied by the number of nodes affected) allows you to reduce the risk in your environment even faster.
- **Offline install updated to use custom container registries** - The offline install process has been expanded to allow users to install from their own custom container registries.
- **Integration status redesign** - The **Integration status** indicator has been moved to navigation sidebar to give you immediate information on the health of your latest scans.
- **Source credential testing** - You can now test access credentials when setting up a scanner on the **Add sources** page before you run a scan.

Resolved issues in this release:

- Duplication of nodes - The RemEDIATE database redesign incorporates a fix to prevent possible duplication of nodes information in the RemEDIATE UI.
- Tenable.sc integration - RemEDIATE has been updated to correct an issue where some vulnerabilities reported in Tenable.sc were not reflected in the RemEDIATE UI.

**Version 1.1.1**

Released 9 January 2020

New in this release:

- **Scan refresh functionality** - You can now manually trigger a rescan of all or selected resources from the **Manage Sources** page.
- **SUSE Linux 11 integration** - RemEDIATE now fully supports SUSE Linux 11.
- **Activity feed updates** - The **Recent Events** table has been updated to include the username of the event initiator.

Resolved issues in this release:

- Offline install - RemEDIATE now uses a dedicated Docker image bundle and docker-compose.yml file for offline installs.
- SSH Updates - RemEDIATE has been updated to enable SSH access to hosts that use CBC ciphers.
- Container DNS issue - RemEDIATE has now been updated to fix a DNS issue where the container added ndots configuration to the /etc/resolv.conf file.

© 2020 Puppet, Inc.
• **Qualys API integration** - Remediate is now able to parse human-readable durations used by the Qualys API that may be reported during daylight savings.

**Version 1.1.0**

Released 31 October 2019

New in this release:

• **Tenable.sc support** - Remediate now supports the Tenable.sc (Security Center) vulnerability scanner.
• **Multi-user support** - User management in Remediate has been completely updated. Administrators can now:
  • Create and manage multiple users accounts.
  • Assign different group privileges to user accounts.
  • Configure Remediate to work with LDAP or Active Directory servers.
• **RBAC Permissions** - Assign user privileges to:
  • Add, remove, or run tasks
  • Add or remove credentials
  • Add or remove sources
• **Remediation workflow improvements** - The vulnerability remediation workflow has been improved and additional information on the vulnerability and the steps needed to remediate it are provided.

Resolved issues in this release:

• **Offline install not working** - The `-o` flag has been introduced to the `remediate start` command to ensure you can start remediate when there is no internet access.
• **OpenSSH private key support** - Remediate now supports the latest version of OpenSSH private keys.

**Version 1.0.1**

Released 1 August 2019.

This is the initial release of Remediate.

Known issues:

• **Unable to install Remediate on Debian8 with the default kernel module.** Upgrade to Kernel 4.9 and install Remediate again.
• **Network discovered nodes being shown as cloud instances.** Hosts discovered via their IP address will be counted as a cloud instance and visible in the top cloud instance by region card.
• **Due to inconsistent DNS lookups, tasks fail to run on discovered hosts.** When discovered hosts are running on the same domain, an inconsistent DNS lookup between discovering hosts and running tasks on discovered hosts results in tasks failing.
• **In a multi-network environment, the first discovery run might not identify the IP or hostname.** Wait for the second discovery run, which happens automatically after four hours.

**Installing Remediate**

• **Architecture** on page 7
Puppet Remediate consists of a number of components and services, each one running as an individual Docker container.
• **System requirements** on page 10
Before installing Puppet Remediate, check to ensure your system meets these requirements.
• **Install Remediate on online nodes** on page 11
Install Puppet Remediate on a Linux or Windows machine that is connected to the internet.
• **Install Remediate on offline nodes** on page 14
  If any of your swarm nodes are offline (do not have external connectivity), you must manually import the Puppet Remediate images in order to install the product.

• **Install Remediate from a custom container registry** on page 16
  You can also use a custom Docker registry to install Puppet Remediate.

• **Install Remediate in multi-network deployments** on page 17
  Puppet Remediate connects to security providers to discover hosts with vulnerabilities. To take action and fix the vulnerabilities, the system needs to connect directly to the hosts. If the host you want to fix is deployed in different network segments that are not directly accessible from where you installed Remediate, you can setup a multi-network deployment.

• **Verify Docker Compose file for online installs** on page 18
  With each Puppet Remediate release, a digital signature is created using the private key portion of an asymmetric key. You can manually validate the signature using the public key portion of the same asymmetric key.

• **Analytics data collection** on page 18
  Puppet Remediate automatically collects data about how you use the product. If you want to opt out of providing this data, you can do so, either during or after installing.

• **Uninstall Remediate** on page 19
  You can uninstall Puppet Remediate easily with a couple of commands.

### Architecture

Puppet Remediate consists of a number of components and services, each one running as an individual Docker container.
<table>
<thead>
<tr>
<th>Service</th>
<th>Container</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Licensing</td>
<td>remediate_licensing</td>
<td>Stores the user licensing information and is queried by the UI using the licensing API.</td>
</tr>
<tr>
<td>Ingress</td>
<td>remediate_frontdoor</td>
<td>The nginx front end listening on port 8443 (HTTPS).</td>
</tr>
<tr>
<td>VR</td>
<td>remediate_vr</td>
<td>Connects to and retrieves host and vulnerability data from Tenable, Qualys and Rapid7.</td>
</tr>
<tr>
<td>Storage</td>
<td>remediate_storage</td>
<td>The storage layer for discovered data which communicates with the remediate_gopdp container on port 5432.</td>
</tr>
<tr>
<td>Identity</td>
<td>remediate_identity</td>
<td>Generates the bearer token for the default user roles: admin and viewer.</td>
</tr>
<tr>
<td>Export</td>
<td>remediate_export</td>
<td>The export service consumes API requests from the UI on port 9200 (HTTPS) and queries the data platform on port 8082 (GRPC).</td>
</tr>
<tr>
<td>gopdp</td>
<td>remediate_gopdp</td>
<td>The data platform is an ingest service implementation that processes discovered data from the edge on port 8082 (GRPC), and exposes the query API to the UI on port 8084.</td>
</tr>
<tr>
<td>Service</td>
<td>Container</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Controller | remediate_controller | The controller manages the discovery process by:  
|          |                  | • Authenticating each API request by validating the bearer token with remediate_identity on port 5556 (HTTPS).  
|          |                  | • Retrieving source and host credentials from the vault on port 8200 (HTTPS).  
|          |                  | • Consuming the discovery and task API requests from the UI on port 9999 (HTTPS).  
|          |                  | • Dispatching discovery and task commands to the edge on port 8081 (GRPC). |
| Edge    | remediate_edge  | The edge consumes the discovery API requests from the controller and invokes each source the user adds. It discovers vulnerabilities, resources, executes ad hoc tasks on target hosts, and submits data to the data platform. The edge services consist of a set of pluggable providers that are determined by which sources are added. |
| Vault   | remediate_vault | The secure store for source and node credentials. |
| UI      | remediate_ui    | The UI enables you to add sources, credentials, and run tasks on target hosts by initiating discovery API requests to the controller on port 9999 (HTTPS). To populate the dashboards and provide a high-level summary view of your infrastructure, the UI queries the data platform on port 8084 (HTTPS) for vulnerabilities and discovered resources. |
## System requirements

Before installing Puppet Remediate, check to ensure your system meets these requirements.

### Supported operating systems

You can install Remediate on the following operating systems.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Versions</th>
<th>System</th>
<th>Prerequisite</th>
</tr>
</thead>
<tbody>
<tr>
<td>CentOS</td>
<td>7 or higher</td>
<td>• Architecture: x64</td>
<td></td>
</tr>
<tr>
<td>Red Hat Enterprise Linux</td>
<td>7 or higher</td>
<td>• Minimum Memory: 8GB</td>
<td></td>
</tr>
<tr>
<td>Debian</td>
<td>8 or higher</td>
<td>• Minimum Storage: 20GB</td>
<td></td>
</tr>
<tr>
<td>Ubuntu</td>
<td>14.04 or higher</td>
<td>• Minimum CPUs: 2</td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>10</td>
<td>Docker CE 17.04.0-ce or higher, or Docker EE 17.06.1-ee or higher for Linux.</td>
<td></td>
</tr>
</tbody>
</table>

When you install Remediate on a virtual machine, you must enable nested virtualization. For more information, see the documentation for the hypervisor you are using.

### Docker requirements

Both Docker CE and Docker EE editions include the option to run Kubernetes as a single-node cluster on a local machine using port 8080. Remediate requires that no other application use ports 8080 and 8443. Note that Docker swarm mode requires additional ports.

If using Docker for Windows, the virtual machine must be configured with 8 GB of memory.

To prevent running out of storage, configure log rotation by editing the `log-driver` and `log-opts` parameters within the daemon configuration file that is located here:

- Linux: `/etc/docker/daemon.json`
- Windows: `%programdata%\docker\config\daemon.json`

To deploy the Docker Compose file, you need to install Docker Compose version 1.24.1.

### Discoverable operating systems

Discover resources that run on these operating systems.

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Versions</th>
<th>Prerequisite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Linux:</td>
<td>5 or higher</td>
<td>For the account Remediate authenticates with, configure bash as the login shell.</td>
</tr>
<tr>
<td>• CentOS</td>
<td>7 or higher</td>
<td></td>
</tr>
<tr>
<td>• Red Hat Enterprise Linux</td>
<td>7 or higher</td>
<td></td>
</tr>
<tr>
<td>Debian</td>
<td>14.04 or higher</td>
<td></td>
</tr>
<tr>
<td>SUSE Linux Enterprise Server</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Ubuntu</td>
<td>14.04 or higher</td>
<td></td>
</tr>
</tbody>
</table>

© 2020 Puppet, Inc.
Operating system | Versions | Prerequisite
--- | --- | ---
Windows Server | 2012 or higher | WinRM is enabled and PowerShell 3.0 is installed.

**System configuration**

Before installing Remediate, make sure that your network is properly configured, and that the time is correctly set and managed on each server. These are the port requirements for a Remediate installation.

<table>
<thead>
<tr>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
</table>
| • 443 (HTTPS) | Required to install or update to the latest version of Remediate from:  
  • storage.googleapis.com  
  • gcr.io |
| • 8443 (HTTPS) | Required to view the Remediate dashboards. |
| • 22 | Required for SSH authentication on discovered Linux hosts. |
| • 5986 (HTTPS)  
• 5985 (HTTP) | Required for WinRM authentication on discovered Windows hosts. |

**Hardware requirements**

Remediate requires:

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage</td>
<td>20.0 GB</td>
</tr>
<tr>
<td>Memory</td>
<td>8.0 GB</td>
</tr>
<tr>
<td>CPUs</td>
<td>2</td>
</tr>
</tbody>
</table>

**Supported browsers**

The Remediate user interface is supported on the latest versions of the following browsers:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge

**Install Remediate on online nodes**

Install Puppet Remediate on a Linux or Windows machine that is connected to the internet.

**Before you begin**

Make sure you meet the system requirements for installing and running Remediate, and that you have received your license.
1. Install and run Docker on the node where you want to install Remediate.
   a) Download Docker for your operating system:
      - Linux: Docker CE 17.04.0-ce or higher, or Docker EE 17.06.1 or higher.
      - Windows: Docker for Windows.
   b) Download Docker Compose.
      \textbf{Note:} If installing Docker Compose on Windows, create a new environment variable called \texttt{COMPOSE_CONVERT_WINDOWS_PATHS} and set it to 1. By doing this, you enable path conversion from Windows-style to Unix-style in volume definitions. For more information, see the Docker Compose documentation.
   c) Initialize a swarm by running the following command:

   \begin{verbatim}
   docker swarm init
   \end{verbatim}

2. Download the Remediate Docker Compose file to the same directory as your license.
   \textbf{Note:} If you want to manually validate the signature, see \textbf{Verify Docker Compose file} for more information.
3. Run the following command (replacing \texttt{your-license.json} with your own license):

   \begin{verbatim}
   docker-compose run remediate start --license-file your-license.json
   \end{verbatim}

   The container images are pulled from the Google Cloud Platform.
4. To check that all the images have downloaded and that the containers have started running, run the following command:

```
docker ps
```

The list of running containers:

<table>
<thead>
<tr>
<th>CONTAINER ID</th>
<th>IMAGE</th>
<th>COMMAND</th>
<th>CREATED</th>
<th>STATUS</th>
<th>PORTS</th>
<th>NAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>36139cda23ca</td>
<td>gcr.io/puppet-discovery/pdp-go:&lt;version&gt;</td>
<td>app/entrypoint.sh&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
<td>8082/tcp, 8087/tcp, 5432/tcp</td>
<td>remediate_gopdp.1.w035ad01fvu0ynai6y641cq</td>
</tr>
<tr>
<td>43709401f394</td>
<td>gcr.io/puppet-discovery/pd-storage:&lt;version&gt;</td>
<td>&quot;storageEntryPoint.s...&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
<td>5432/tcp</td>
<td>remediate_storage.1.omn1rrd3rwhw03ngurgtcnxk</td>
</tr>
<tr>
<td>551b061acc98</td>
<td>gcr.io/puppet-discovery/licensing-api:&lt;version&gt;</td>
<td>entrypoint.sh&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
<td>8082/tcp, 8087/tcp, 5432/tcp</td>
<td>remediate_storage.1.omv1rrd3rwhw03ngurgtcnxk</td>
</tr>
<tr>
<td>43a3568e6b5a</td>
<td>vault:1.1.2</td>
<td>&quot;docker-entrypoint.s...&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
<td>8080/tcp</td>
<td>remediate_vault.1.k0jgyxx1451dn4pqadghv6j0o</td>
</tr>
<tr>
<td>11b90d8564ef</td>
<td>gcr.io/puppet-discovery/identity:&lt;version&gt;</td>
<td>&quot;storageEntryPoint.s...&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
<td>8080/tcp</td>
<td>remediate_identity.1.b1wkt4x4ps107949k14ctsgx</td>
</tr>
<tr>
<td>1235fd27cbdc</td>
<td>gcr.io/puppet-discovery/frontdoor:&lt;version&gt;</td>
<td>&quot;nginx -g 'daemon of...&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
<td>80/tcp</td>
<td>remediate_frontdoor.1.oib8jsr7u4z6wqvjz02mxcz</td>
</tr>
<tr>
<td>a3a6af5269</td>
<td>gcr.io/puppet-discovery/pdp-proxy:&lt;version&gt;</td>
<td>&quot;pdp-proxy-svc&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
<td>9200/tcp</td>
<td>remediate_exchange.1.lun509pnc2z4b4pf3rdj1ro0x</td>
</tr>
<tr>
<td>c60250b8a2eb</td>
<td>gcr.io/puppet-discovery/node-ui:&lt;version&gt;</td>
<td>&quot;user/src/app/entryp...&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
<td>80/tcp</td>
<td>remediate_node-ui.1.3u0te5wou7t4h2z0c46mmyo</td>
</tr>
<tr>
<td>f9af99dc9ca7</td>
<td>gcr.io/puppet-discovery/edge:&lt;version&gt;</td>
<td>edge-svc&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
<td>9997/tcp</td>
<td>remediate_exchange.1.koawcnjoe2tabwcb173619fu</td>
</tr>
<tr>
<td>149485b54fe8</td>
<td>gcr.io/puppet-discovery/controller:&lt;version&gt;</td>
<td>controller-svc&quot;</td>
<td>4 minutes ago</td>
<td>Up 4 minutes (healthy)</td>
<td>9999/tcp</td>
<td>remediate_controller.1.mww2fm9up41meisjor189hr4</td>
</tr>
<tr>
<td>f9flab4a029d</td>
<td>gcr.io/puppet-discovery/vr:&lt;version&gt;</td>
<td>vr-svc&quot;</td>
<td>4 minutes ago</td>
<td>Up 4 minutes (healthy)</td>
<td>80/tcp</td>
<td>remediate_vr.1.yjlliup91g4mac1bk1vww2nqq</td>
</tr>
</tbody>
</table>

5. To access Remediate on a local workstation, the URL is https://localhost:8443, or port 8443 on the host where you installed Remediate.

**Note:** When you first open the site, a warning message will be displayed that indicates the site certificate is untrusted. This is because Remediate uses a self-signed certificate and is expected behavior. Accept the certificate to continue.

6. Read and accept the software license agreement.

7. Sign in to Remediate.

For default usernames and passwords, see Managing user access.

**Related information**

**System requirements** on page 10
Before installing Puppet Remediate, check to ensure your system meets these requirements.

**Install Remediate on offline nodes** on page 14
If any of your swarm nodes are offline (do not have external connectivity), you must manually import the Puppet Remediate images in order to install the product.

© 2020 Puppet, Inc.
Install Remediate from a custom container registry on page 16
You can also use a custom Docker registry to install Puppet Remediate.

Verify Docker Compose file for online installs on page 18
With each Puppet Remediate release, a digital signature is created using the private key portion of an asymmetric key. You can manually validate the signature using the public key portion of the same asymmetric key.

Managing user access on page 21
As a Remediate administrator, you can create new user accounts and assign group-based access privileges to them.

Install Remediate on offline nodes
If any of your swarm nodes are offline (do not have external connectivity), you must manually import the Puppet Remediate images in order to install the product.

Before you begin
Prior to installing Remediate on any offline nodes, check that you have carried out the following prerequisite tasks:

• Make sure you meet the system requirements for installing and running Remediate, and that you have received your license.
• Ensure that Docker and Docker Compose are installed on the nodes where you want to install the Remediate images. If you are installing Docker Compose on Windows, ensure that you create a new Windows environment variable called `COMPOSE_CONVERT_WINDOWS_PATHS` and set it to 1. This enables path conversion from Windows-style to Unix-style in volume definitions.
• Ensure that your license is added to the nodes where you want to install the Remediate images.

To install Remediate on nodes that do not have network connectivity:

1. On a node with internet connectivity:
   a) Download the offline Remediate image bundle (https://storage.googleapis.com/remediate/stable/1.2.1/offline/images.tar.gz)

   **Note:** Skip this step if you are using your own custom Docker registry.

2. Optionally, you can verify the image bundle and offline `docker-compose.yml` files signatures:
   With each Puppet Remediate release, a digital signature is created using the private key portion of an asymmetric key. You can manually validate the signature using the public key portion of the same asymmetric key.
   a) **Download** the offline `docker-compose.yml` file signature and the image bundle signature, along with the public key to the same directory as your `docker-compose.yml` and license file.
   b) **Run** the following commands:

```
openssl dgst -sha256 -verify puppet-remediate-signing-key.pub -signature docker-compose_signature docker-compose.yml

And:
openssl dgst -sha256 -verify puppet-remediate-signing-key.pub -signature images_signature images.tar.gz
```

   If the signature is valid, you will get the following response for each command:

   `Verified Ok`

3. Copy the Remediate image bundle and offline `docker-compose.yml` file to the offline node where you want to install Remediate.
4. On the node where you want to install Remediate, initialize a swarm by running the following command:

   ```bash
docker swarm init
   ```

5. Run the Docker `load` command:

   ```bash
docker load -i images.tar.gz
   ```

6. Use the following command to start Remediate (replacing `your-license.json` with your own license):

   ```bash
docker-compose run remediate start -o --license-file ./your-license.json
   ```

7. To check that the containers have started running, run this command:

   ```bash
docker ps
   ```

   The list of running containers:

<table>
<thead>
<tr>
<th>CONTAINER ID</th>
<th>IMAGE</th>
<th>COMMAND</th>
<th>CREATED</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>36139cda23ca</td>
<td>gcr.io/puppet-discovery/pdp-go:&lt;version&gt;</td>
<td>&quot;app/entrypoint.sh&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
</tr>
<tr>
<td>43709401f394</td>
<td>gcr.io/puppet-discovery/pd-storage:&lt;version&gt;</td>
<td>&quot;storageEntryPoint.s...&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
</tr>
<tr>
<td>5432/tcp</td>
<td>remediate_storage.1.ormv1rre3rwhnwo3ngurtnxk</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>551b061acc98</td>
<td>gcr.io/puppet-discovery/licensing-api:&lt;version&gt;</td>
<td>&quot;entrypoint.sh&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
</tr>
<tr>
<td>43a3568e6b5a</td>
<td>vault:1.1.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8200/tcp</td>
<td>remediate_vault.1.k0jgyxxi451dn4pqadghqj0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11b90d8564ef</td>
<td>gcr.io/puppet-discovery/identity:&lt;version&gt;</td>
<td>&quot;opt/jboss/tools/do...&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
</tr>
<tr>
<td>1235fd27cbdc</td>
<td>gcr.io/puppet-discovery/frontdoor:&lt;version&gt;</td>
<td>&quot;nginx -g 'daemon of...&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
</tr>
<tr>
<td>43a3568e6b5a</td>
<td>vault:1.1.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8200/tcp</td>
<td>remediate_vault.1.k0jgyxxi451dn4pqadghqj0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a3a641af5269</td>
<td>gcr.io/puppet-discovery/pdp-proxy:&lt;version&gt;</td>
<td>&quot;pdp-proxy-svc&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
</tr>
<tr>
<td>c60250b8a2eb</td>
<td>gcr.io/puppet-discovery/node-ui:&lt;version&gt;</td>
<td>&quot;usr/src/app/entryp...&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
</tr>
<tr>
<td>f9af99dc9ca7</td>
<td>gcr.io/puppet-discovery/edge:&lt;version&gt;</td>
<td>&quot;edge-svc&quot;</td>
<td>3 minutes ago</td>
<td>Up 3 minutes (healthy)</td>
</tr>
<tr>
<td>149485b54fec</td>
<td>gcr.io/puppet-discovery/controller:&lt;version&gt;</td>
<td>&quot;controller-svc&quot;</td>
<td>4 minutes ago</td>
<td>Up 4 minutes (healthy)</td>
</tr>
<tr>
<td>f9f1ab4a029d</td>
<td>gcr.io/puppet-discovery/vr:&lt;version&gt;</td>
<td>&quot;vr-svc&quot;</td>
<td>4 minutes ago</td>
<td>Up 4 minutes (healthy)</td>
</tr>
</tbody>
</table>

8. To access Remediate on a local workstation, the URL is https://localhost:8443, or port 8443 on the host where you installed Remediate.

   **Note:** When you first open the site, a warning message will be displayed that indicates the site certificate is untrusted. This is because Remediate uses a self-signed certificate and is expected behavior. Accept the certificate to continue.

9. Read and accept the software license agreement.
10. Sign in to Remediate.
   For default usernames and passwords, see Managing user access.

Related information
System requirements on page 10
Before installing Puppet Remediate, check to ensure your system meets these requirements.

Install Remediate on online nodes on page 11
Install Puppet Remediate on a Linux or Windows machine that is connected to the internet.

Install Remediate from a custom container registry on page 16
You can also use a custom Docker registry to install Puppet Remediate.

Managing user access on page 21
As a Remediate administrator, you can create new user accounts and assign group-based access privileges to them.

Install Remediate from a custom container registry
You can also use a custom Docker registry to install Puppet Remediate.

Before you begin
Prior to installing Remediate using a custom Docker registry, check that you have carried out the following prerequisite tasks:

• Make sure you meet the system requirements for installing and running Remediate, and that you have received your license.
• Ensure that Docker and Docker Compose are installed on the nodes where you want to install the Remediate images. If you are installing Docker Compose on Windows, ensure that you create a new Windows environment variable called COMPOSE_CONVERT_WINDOWS_PATHS and set it to 1. This enables path conversion from Windows-style to Unix-style in volume definitions.
• Ensure that your license is added to the nodes where you want to install the Remediate images.
• If you use a custom Docker registry to store the Remediate Docker images, you must set the REGISTRY environment variable to point to the URL of your custom docker registry, e.g.:

```
export REGISTRY=my-custom-docker-registry.local.net
```

In this example, Remediate will contact my-custom-docker-registry.local.net to download the docker images it needs.

1. On the node where you want to install Remediate, initialize a swarm by running the following command:
   ```bash
docker swarm init
```

2. Download the Remediate Docker Compose file to the same directory as your license.
   Note: If you want to manually validate the signature, see Verify Docker Compose file for more information.

3. Run the following command (replacing your-license.json with your own license):
   ```bash
docker-compose run remediate start --license-file your-license.json
```

The container images are pulled from the custom registry location specified by your REGISTRY environment variable.

4. To check that all the images have downloaded and that the containers have started running, run the following command:
   ```bash
docker ps
```
5. To access Remediate on a local workstation, the URL is https://localhost:8443, or port 8443 on the host where you installed Remediate.

   Note: When you first open the site, a warning message will be displayed that indicates the site certificate is untrusted. This is because Remediate uses a self-signed certificate and is expected behavior. Accept the certificate to continue.

6. Read and accept the software license agreement.

7. Sign in to Remediate.

   For default usernames and passwords, see Managing user access.

Related information

System requirements on page 10
Before installing Puppet Remediate, check to ensure your system meets these requirements.

Install Remediate on online nodes on page 11
Install Puppet Remediate on a Linux or Windows machine that is connected to the internet.

Install Remediate on offline nodes on page 14
If any of your swarm nodes are offline (do not have external connectivity), you must manually import the Puppet Remediate images in order to install the product.

Verify Docker Compose file for online installs on page 18
With each Puppet Remediate release, a digital signature is created using the private key portion of an asymmetric key. You can manually validate the signature using the public key portion of the same asymmetric key.

Managing user access on page 21
As a Remediate administrator, you can create new user accounts and assign group-based access privileges to them.

Install Remediate in multi-network deployments

Puppet Remediate connects to security providers to discover hosts with vulnerabilities. To take action and fix the vulnerabilities, the system needs to connect directly to the hosts. If the host you want to fix is deployed in different network segments that are not directly accessible from where you installed Remediate, you can setup a multi-network deployment.

In a multi-network deployment, an edge service is deployed inside each network segment. Remediate instructs the edge to connect to the hosts when executing tasks, and then leverages Docker swarm to deploy an edge service on each swarm worker. Workers must have direct connectivity to the manager — the main node where you installed Remediate. For more information on workers and managers, see the Docker documentation.

For more information on installing Remediate on nodes that are not connected to the internet, see instructions on how to Install Remediate on offline nodes.

1. Set up a Docker swarm environment.
   a) On the node you want to be the manager, run:

   ```
   docker swarm init
   ```

   b) To add a manager to the swarm, run the following command and follow the instructions:

   ```
   docker swarm join-token manager
   ```

2. Using the key output from the above command as the token, run the following command on each of the workers:

   ```
   docker swarm join --token
   ```

   This command adds a worker to each network segment.

3. On the manager node, follow the normal installation instructions for Remediate. The swarm automatically deploys the edge on the workers.
4. After the installation is complete, verify the expected number of edges is running with the following command:

```bash
docker service ps remediate_remote-edge
```

### Related information

- **Install Remediate on offline nodes** on page 14
  - If any of your swarm nodes are offline (do not have external connectivity), you must manually import the Puppet Remediate images in order to install the product.

### Verify Docker Compose file for online installs

With each Puppet Remediate release, a digital signature is created using the private key portion of an asymmetric key. You can manually validate the signature using the public key portion of the same asymmetric key.

1. Download the signature file and the public key to the same directory as your `docker-compose.yml` and license file.

   **Note:** For instructions on downloading the `docker-compose.yml` and license files, see the instructions on how to **Install Remediate on online nodes**.

2. Run the following command:

   ```bash
   openssl dgst -sha256 -verify puppet-remediate-signing-key.pub -signature signature docker-compose.yml
   ```

   If the signature is valid, you will get the following response:

   ```
   Verified Ok
   ```

### Analytics data collection

Puppet Remediate automatically collects data about how you use the product. If you want to opt out of providing this data, you can do so, either during or after installing.

### What data does Remediate collect?

When Puppet Remediate starts, and restarts, it collects the following information:

- Browser name and version.
- Device type.
- Form submission events, but not form input data.
- Anonymized IP address.
- Geographic data inferred by the anonymized IP address.
- JavaScript exceptions.
- Link, button, and form element clicks.
- Operating system and version.
- Page views.
• Session durations and session details including:
  
  • number and title of screens viewed
  • screens arrived at
  • exit screens

**How does Puppet use the data?**

The data we collect is one of many methods we use for learning about our customers. For example, understanding how you navigate through the web interface can help us optimize the navigation so that you can complete your work faster. And, learning what kinds of data sources are used, helps us to prioritize new functionality like increasing how many resources we can discover for those sources.

**Opt out before installing**

Remediate does not collect any analytics data during the installation process. However, if you want to disable data collection before you start Remediate for the first time, run the following Docker Compose command from the folder where your `docker-compose.yml` file is located:

```bash
docker-compose run remediate config set-override analytics false
```

**Opt out after installing**

If you have already installed Remediate, and want to disable data collection, run the following Docker Compose command from the folder where your `docker-compose.yml` file is located:

```bash
docker-compose run remediate config set-override analytics false
```

Restart Remediate to complete the update:

```bash
docker-compose run remediate restart
```

**Uninstall Remediate**

You can uninstall Puppet Remediate easily with a couple of commands.

To uninstall Remediate:

1. Stop and remove the running Remediate containers and any associated volumes by issuing the following command from the folder where your `docker-compose.yml` file is located:

   ```bash
docker-compose run remediate reset
   ```

2. To delete the Remediate containers on your system, use the following Docker command:

   ```bash
docker system prune --filter label=com.docker.compose.service=remediate
   ```

**Upgrading Remediate**

Upgrading Puppet Remediate is easy can be done for both online and offline nodes.

- Upgrade Remediate on online nodes on page 20

Upgrading to a new version of Puppet Remediate on a node with internet access can be done with a few simple commands.
Upgrading to a new version of Puppet Remediate on a node with internet access can be done with a few simple commands.

To upgrade from Remediate 1.1.0 to Remediate 1.2.1:

1. Run the following command from the folder where your `docker-compose.yml` file is located:

   ```bash
docker-compose run remediate stop
   ```

2. Remove the `oauth_client.json` file:

   ```bash
docker secret rm oauth_client.json
   ```

3. Finally, run the Remediate update command:

   ```bash
docker-compose run remediate update
   ```

   **Remember:** After you update Remediate, any existing discovered data is lost. You must wait for the next scheduled data discovery run completes to see data in the UI. Alternatively, click Discover All on the Manage Sources page to start a discovery run manually.

Upgrading to new version of Puppet Remediate on offline machines can be done with a few simple commands.

To upgrade from Remediate 1.2.0 to Remediate 1.2.1 on an offline node:

1. On a node with internet connectivity:

   **Note:** Skip this step if you are using your own custom Docker registry.
2. Optionally, you can verify the image bundle and offline `docker-compose.yml` files signatures:
   With each Puppet Remediate release, a digital signature is created using the private key portion of an asymmetric key. You can manually validate the signature using the public key portion of the same asymmetric key.
   
a) Download the offline `docker-compose.yml` file signature and the image bundle signature, along with the public key to the same directory as your `docker-compose.yml` and license file.
   
b) Run the following commands:

   ```bash
   openssl dgst -sha256 -verify puppet-remediate-signing-key.pub -signature docker-compose_signature docker-compose.yml
   And:
   openssl dgst -sha256 -verify puppet-remediate-signing-key.pub -signature images_signature images.tar.gz
   ```

   If the signature is valid, you will get the following response for each command:

   ```plaintext
   Verified Ok
   ```

3. Copy the Remediate image bundle and offline `docker-compose.yml` file to the offline node where you want to install Remediate.

4. Run the following command from the folder where your `docker-compose.yml` file is located:

   ```bash
   docker-compose run remediate stop
   ```

5. Remove the `oauth_client.json` file:

   ```bash
   docker secret rm oauth_client.json
   ```

6. Use the following command to start Remediate (replacing `your-license.json` with your own license):

   ```bash
   docker-compose run remediate start -o --license-file ./your-license.json
   ```

Remember: After you update Remediate, any existing discovered data is lost. You must wait for the next scheduled data discovery run completes to see data in the UI. Alternatively, click Discover All on the Manage Sources page to start a discovery run manually.

## Managing user access

As a Remediate administrator, you can create new user accounts and assign group-based access privileges to them.

When you first install Remediate, you have one default superuser account: admin. Admin users have full access to all Remediate user interface functionality and can add and manage other users.
Remediate | Managing user access | 22

<table>
<thead>
<tr>
<th>Role</th>
<th>Username</th>
<th>Default password</th>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator</td>
<td>admin</td>
<td>@Admin</td>
<td>• Add and remove sources and credentials.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Upload scripts and modules.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Run tasks.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• View dashboards.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• View all node, package, container listing and details pages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Add, update, delete, log out other users</td>
</tr>
</tbody>
</table>

Manage user accounts

As an admin user, you can add, view, update or delete additional user accounts and assign them to groups.

The following groups are available to add to accounts:

<table>
<thead>
<tr>
<th>Permission Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add-credential</td>
<td>Add access credentials for a node.</td>
</tr>
<tr>
<td>add-source</td>
<td>Add a vulnerability scanner or infrastructure source.</td>
</tr>
<tr>
<td>add-task</td>
<td>Add a new remediation task.</td>
</tr>
<tr>
<td>remove-credential</td>
<td>Remove access credentials for a node.</td>
</tr>
<tr>
<td>remove-source</td>
<td>Remove a vulnerability scanner or infrastructure source.</td>
</tr>
<tr>
<td>remove-task</td>
<td>Remove a new remediation task.</td>
</tr>
<tr>
<td>run-task</td>
<td>Run a remediation task.</td>
</tr>
</tbody>
</table>

Each group represents a user privilege that can be granted to a user account. If you do not add any groups to a user account, that user will only have read-only access to the Remediate UI.

In addition to creating user accounts manually, you can also configure remediate to pull user information from your LDAP or Active Directory server.

- **Add new users** on page 23
  As a Remediate admin, you can add new user accounts and assign them group privileges.

- **Update a user** on page 23
  From time to time you might need to change a user’s password, or update the groups assigned to their account.

- **Delete a user** on page 24
  Remediate administrators can delete other user accounts, including admin accounts.

- **Force log out** on page 24
  Holders of the admin account can force other users to log out.

- **LDAP configuration** on page 24
  You can set up Remediate to use LDAP content to authenticate users.

- **Active Directory configuration** on page 25

© 2020 Puppet, Inc.
You can set up Remediate to use Active Directory to authenticate users.

- **LDAP mappers** on page 27
  LDAP mappers are listeners, which are triggered by the LDAP Provider at various points, and provide another extension point to LDAP integration.

- **Work with user groups in Remediate** on page 28
  Puppet Remediate provides a limited number of roles that allow you to control what users can and can’t do.

**Related information**
**Add new users** on page 23
As a Remediate admin, you can add new user accounts and assign them group privileges.

**Update a user** on page 23
From time to time you might need to change a user’s password, or update the groups assigned to their account.

**Add sources** on page 41
Add your vulnerability scanner to detect and fix vulnerabilities across your infrastructure. To discover nodes, packages, and containers running on your entire infrastructure, add multiple infrastructure sources.

**Force log out** on page 24
Holders of the admin account can force other users to log out.

**LDAP configuration** on page 24
You can set up Remediate to use LDAP content to authenticate users.

**Active Directory configuration** on page 25
You can set up Remediate to use Active Directory to authenticate users.

### Add new users
As a Remediate admin, you can add new user accounts and assign them group privileges.

**Important:** Until the first run wizard is completed, you should only log in as the default admin user.

To add a new user:

1. In the main Remediate UI, click **User admin** on the navigation sidebar.
   The user admin login page is displayed.
2. Log in with the administrator username and password.
3. Select **Users** in the sidebar, and click **Add user**.
4. On the **Add user** page:
   a) Enter a **Username**, and click **Save**.
      **Important:** Do not enter anything in the **Required User Actions** or **Email Verified** fields as it may prevent the user from logging in.
   b) On the **User** page for the new account, click the **Credentials** tab.
   c) Enter and confirm a password in the relevant fields, and click **Reset password** when you are done.
   d) On the **Groups** tab, select the groups you want to add to the new account, and click **Join**.

### Update a user
From time to time you might need to change a user’s password, or update the groups assigned to their account.

**Important:** Until the first run wizard is completed, you should only log in as the default admin user.

To update a user account:

1. In the main Remediate UI, click **User admin** on the navigation sidebar.
   The user admin login page is displayed.
2. Log in with the administrator username and password.
3. Select **Users** in the sidebar, and click **Edit** on the row assigned to the user account you want to edit.

4. Make changes on the relevant tabs in the **User** page for the account.

   **Important:** Do not enter anything in the **Required User Actions** or **Email Verified** fields as it may prevent the user from logging in.

---

**Delete a user**

Remediate administrators can delete other user accounts, including admin accounts.

To delete a user:

1. In the main Remediate UI, click **User admin** on the navigation sidebar.
   The user admin login page is displayed.
2. Log in with the administrator username and password.
3. Select **Users** in the sidebar, and click **Delete** on the row assigned to the user account you want to remove.
4. Click **Delete** on the pop-up dialog that appears to confirm.
   The user account is no longer displayed on the **Users** page.

---

**Force log out**

Holders of the admin account can force other users to log out.

To log a user out:

1. In the main Remediate UI, click **User admin** on the navigation sidebar.
   The user admin login page is displayed.
2. Log in with the administrator username and password.
3. Select **Users** in the sidebar, and click **Edit** on the row assigned to the user account you want to log out.
4. On the **Sessions** tab, click **Logout**.

---

**LDAP configuration**

You can set up Remediate to use LDAP content to authenticate users.

To configure Remediate to work with your LDAP server:

1. In the main Remediate UI, click **User admin** on the navigation sidebar.
   The user admin login page is displayed.
2. Log in with the administrator username and password.
3. On the sidebar, click **User Federation** and select **ldap** from the drop-down list.
4. In the **Required Settings** area:
   a) Ensure **Import Users** is set to **ON**. Failure to do so can cause issues when you try to sync users.
   b) Select the LDAP server type from the **vendor** drop-down list.
   
   **Note**: Some of the required parameters will be automatically filled out depending on the vendor option you selected. These parameters can be changed if they are unsuitable.
   
   Hover over the icon beside each field for more information about the required content for that field.
   c) Enter the **Username LDAP Attribute**.
      For many LDAP vendors this will be **uid**, but may be different in your case.
   d) Enter the **RDN LDAP Attribute**.
      This is the attribute that defines the relative distinguished name of the typical user distinguished name. Usually, it is the same as the username attribute.
   e) Enter the **UUID LDAP attribute**.
      For many LDAP vendors this is **entryUUID**, but may be different in your case.
   f) Enter values for all LDAP **User Object Classes**. Values must be comma-separated, for example:
      
      InetOrgPerson,organizationPerson
   
   g) Enter the **Connection URL** of your LDAP server.
      You can use the **Test connection** button to ensure that Remediate can connect to the server.
   h) Enter the **User DN**.
      The is the full distinguished name of the LDAP tree where your users are located. For example:
      
      ou=myusers,dc=myhost,dc=com
   i) Select the **Authentication Type**. There are two options:
      
      - **None** - anonymous LDAP authentication.
      - **Simple** - Bind credential + Bind password authentication.
   j) Enter the **Bind DN** field. This is the distinguished name of the LDAP admin account used to access the LDAP server. For example:
      
      uid=admin,dc=myhost,dc=com
   k) Enter the LDAP admin password in the **Bind Credential** field.
      You can use the **Test authentication** button to see if Remediate can access the LDAP server.

5. To configure the synchronization schedule between your LDAP or Active Directory server and Remediate:
   a) Enter the maximum number of users to be imported in any transaction in the **Batch Size** field. The default is **1000**.
   b) If you want to set up a full synchronization, set **Periodic Full Sync** to **ON**, and enter a value in seconds in the **Full Sync Period** field. The default is **604800** seconds (every 7 days).
   c) If you want to synchronize only new and changed users, set **Periodic Changed Users Sync** to **ON** and enter a value in seconds in the **Changed Users Sync Period** field. The default is **86400** seconds (every 24 hours).

6. When you are done, click **Save**.

**Related information**

**LDAP mappers** on page 27

 LDAP mappers are listeners, which are triggered by the LDAP Provider at various points, and provide another extension point to LDAP integration.

**Active Directory configuration**

You can set up Remediate to use Active Directory to authenticate users.

© 2020 Puppet, Inc.
To configure Remediate to use your Active Directory server:

1. In the main Remediate UI, click **User admin** on the navigation sidebar.  
   The user admin login page is displayed.
2. Log in with the administrator username and password.
3. On the sidebar, click **User Federation** and select **ldap** from the drop-down list.
4. In the **Required Settings** area:
   a) Select **Active Directory** from the **vendor** drop-down list. 
      
      **Note**: Some of the required parameters will be automatically filled. These parameters can be changed if they are unsuitable.
      
      Hover over the icon beside each field for more information about the required content for that field.
   b) Enter the **Username LDAP Attribute**.
   c) Enter the **RDN LDAP Attribute**.
      This is the attribute that defines the relative distinguished name of the typical user distinguished name. Usually, it is the same as the username attribute. However, you might also use `cn` for this attribute if you used `sAMAccountName` for the username attribute.
   d) Enter the **UUID LDAP attribute**.
      For Active Directory, this is usually `objectGUID`.
   e) Enter values for all LDAP **User Object Classes**. Values must be comma-separated, for example:
      ```
      InetOrgPerson,organizationPerson
      ```
   f) Enter the **Connection URL** of your LDAP server.
      For example:
      ```
      ldap://ds.example.com:389
      ```
      Or:
      ```
      ldaps://ds.example.com:636
      ```
      You can use the **Test connection** button to ensure that Remediate can connect to the server.
   g) Enter the **User DN**.
      This is the full distinguished name of the LDAP tree where your users are located. For example:
      ```
      ou=Users,dc=remediate,dc=io
      ```
   h) Select the **Authentication Type**. There are two options:
      - **None** - anonymous LDAP authentication.
      - **Simple** - Bind credential + Bind password authentication.
   i) Enter the **Bind DN** field. This is the distinguished name of the LDAP admin account used to access the LDAP server. For example:
      ```
      cn=admin,dc=remediate,dc=io
      ```
   j) Enter the LDAP admin password in the **Bind Credential** field.
      You can use the **Test authentication** button to see if Remediate can access the LDAP server.
5. To configure the synchronization schedule between your LDAP or Active Directory server and Remediate:
   a) Enter the maximum number of users to be imported in any transaction in the **Batch Size** field. The default is 1000.
   b) If you want to set up a full synchronization, set **Periodic Full Sync** to **ON**, and enter a value in seconds in the **Full Sync Period** field. The default is 604800 seconds (every 7 days).
   c) If you want to synchronize only new and changed users, set **Periodic Changed Users Sync** to **ON** and enter a value in seconds in the **Changed Users Sync Period** field. The default is 86400 seconds (every 24 hours).

6. When you are done, click **Save**.

**Related information**

**LDAP mappers** on page 27

LDAP mappers are listeners, which are triggered by the LDAP Provider at various points, and provide another extension point to LDAP integration.

### LDAP mappers

LDAP mappers are listeners, which are triggered by the LDAP Provider at various points, and provide another extension point to LDAP integration.

LDAP mappers are triggered when a user logs in via LDAP and needs to be imported, or when a user is queried from the user admin interface. When you create an LDAP provider, a default set of built-in mappers for this provider is made available. You are free to change this set and create a new mapper, or update/delete existing ones.

<table>
<thead>
<tr>
<th>Mapper Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Attribute Mapper</td>
<td>This mapper allows you to specify which LDAP attribute is mapped to which user attribute. So, for example, you can configure that LDAP attribute <code>mail</code> to the attribute <code>email</code> in the Remediate user database. For this mapper implementation, there is always a one-to-one mapping (one LDAP attribute is mapped to one Remediate user attribute)</td>
</tr>
<tr>
<td>FullName Mapper</td>
<td>This mapper allows you to specify that the full name of the user, which is saved in some LDAP attribute (usually <code>cn</code>), is to be mapped to <code>firstName</code> and <code>lastname</code> attributes in the Remediate user database. Using <code>cn</code> to contain the full name of a user is a common case for some LDAP deployments.</td>
</tr>
<tr>
<td>Role Mapper</td>
<td>This mapper is not applicable for the current release.</td>
</tr>
<tr>
<td>Hardcoded Role Mapper</td>
<td>This mapper will grant a specified role to each user linked with LDAP.</td>
</tr>
<tr>
<td>Group Mapper</td>
<td>This mapper allows you to configure group mappings from LDAP into Remediate user group mappings. A group mapper can be used to map LDAP groups from a particular branch of an LDAP tree into groups in Remediate. It will also propagate user-group mappings from LDAP into user-group mappings in Remediate.</td>
</tr>
<tr>
<td>MSAD User Account Mapper</td>
<td>This mapper is specific to Microsoft Active Directory (MSAD). It can tightly integrate the MSAD user account state into the Remediate user account state (account enabled, password is expired etc) using the <code>userAccountControl</code> and <code>pwdLastSet</code> LDAP attributes (both of which are specific to MSAD and are not LDAP standard).</td>
</tr>
<tr>
<td>Certificate Mapper</td>
<td>This mapper is used specifically for mapping X.509 certificates. It will generally be used in conjunction with X.509 authentication and <strong>Full certificate in PEM format</strong> as an identity source. It behaves the same way as the User Attribute Mapper, but allows you to filter for an LDAP attribute which stores a certificate in either PEM or DER format. It is generally advised to enable <strong>Always Read Value From LDAP</strong> with this mapper.</td>
</tr>
</tbody>
</table>
Adding LDAP mappers

By default, there are User Attribute mappers that map basic Remediate user attributes like `username`, `firstname`, `lastname`, and `email` to corresponding LDAP attributes. You are free to extend these and provide additional attribute mappings.

To add an LDAP mapper:

1. In the main Remediate UI, click User admin on the navigation sidebar.
   The user admin login page is displayed.
2. Log in with the administrator username and password.
3. On the sidebar, click User Federation, and then click the row assigned to the LDAP/AD connection you want to update.
4. Click the Mappers tab, and then Create.
5. Give the mapper a Name, and select the Mapper Type from the drop-down list.
6. Enter the required information in the relevant fields for your mapper type.
   **Tip:** Hover over the icon beside each field for more information about the required content for that field.
7. Click Save when you are done.
   **Tip:** To edit your mapper settings, click on the row assigned to it on the Mappers tab.

Work with user groups in Remediate

Puppet Remediate provides a limited number of roles that allow you to control what users can and can’t do.

The following permission groups are available to add to accounts:

<table>
<thead>
<tr>
<th>Permission Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add-credential</td>
<td>Add access credentials for a node.</td>
</tr>
<tr>
<td>add-source</td>
<td>Add a vulnerability scanner or infrastructure source.</td>
</tr>
<tr>
<td>add-task</td>
<td>Add a new remediation task.</td>
</tr>
<tr>
<td>remove-credential</td>
<td>Remove access credentials for a node.</td>
</tr>
<tr>
<td>remove-source</td>
<td>Remove a vulnerability scanner or infrastructure source.</td>
</tr>
<tr>
<td>remove-task</td>
<td>Remove a new remediation task.</td>
</tr>
<tr>
<td>run-task</td>
<td>Run a remediation task.</td>
</tr>
</tbody>
</table>

When using LDAP, you may want to grant a certain user group full permissions but restrict other user groups to more limited permissions.

This following example describes a sample scenario. It shows how to grant everyone in the LDAP group called `secops` full permissions, while letting all other users just run tasks.

To achieve this, we'll configure 2 LDAP providers with default mappers, and assign a different priority to each provider.

1. In the main Remediate UI, click User admin on the navigation sidebar.
   The user admin login page is displayed.
2. Log in with the administrator username and password.
3. On the sidebar, click User Federation and select ldap from the drop-down list.
4. Create an LDAP Provider for the `secops` user group:
   a) Set the **Priority** to 0. This is the highest priority.
   b) Specify a **Custom User LDAP Filter** that only includes members of `secops`. As in the following example:

<table>
<thead>
<tr>
<th>Provider ID</th>
<th>e91fdd55-6365-4868-bc0a-10727af274fe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>ON</td>
</tr>
<tr>
<td>Console Display Name</td>
<td>ldap</td>
</tr>
<tr>
<td>Priority</td>
<td>0</td>
</tr>
<tr>
<td>Import Users</td>
<td>ON</td>
</tr>
<tr>
<td>Edit Mode</td>
<td></td>
</tr>
<tr>
<td>Sync Registrations</td>
<td>OFF</td>
</tr>
<tr>
<td>* Vendor</td>
<td>Other</td>
</tr>
<tr>
<td>* Username LDAP attribute</td>
<td>uid</td>
</tr>
<tr>
<td>* RDN LDAP attribute</td>
<td>uid</td>
</tr>
<tr>
<td>* UUID LDAP attribute</td>
<td>entryUUID</td>
</tr>
<tr>
<td>* User Object Classes</td>
<td>inetOrgPerson, organizationalPerson</td>
</tr>
<tr>
<td>* Connection URL</td>
<td>ldaps://ldap.puppetlabs.com:636</td>
</tr>
<tr>
<td>* Users DN</td>
<td>ou=users,dc=puppetlabs,dc=com</td>
</tr>
<tr>
<td>* Bind Type</td>
<td>simple</td>
</tr>
<tr>
<td>* Bind DN</td>
<td>cn=foobar,ou=service,ou=users,dc=puppetlabs,dc=com</td>
</tr>
<tr>
<td>* Bind Credential</td>
<td>**********</td>
</tr>
<tr>
<td>Custom User LDAP Filter</td>
<td>(memberOf=cn=secops,ou=groups,dc=puppetlabs,dc=com)</td>
</tr>
</tbody>
</table>

For a full step-by-step guide to creating an LDAP provider, see [LDAP configuration](#) in the related information section at the bottom of this page.
5. Click **Save** when you are done.

6. Next, create default mappers for each default permissions group you want to assign to the *secops* user group. This example shows a mapper for the *add-source* permission: group:

For a full step-by-step guide to creating a default mapper, see **LDAP mappers** in the related information section at the bottom of this page.
7. Create an LDAP Provider for all other users:
   
a) Set the **Priority** to 10. This number must be lower than 0 so that users are evaluated initially against the first LDAP provider, and then only this provider if they are not members of the *secops* LDAP group.
Related information
LDAP configuration on page 24
You can set up Remediate to use LDAP content to authenticate users.
LDAP mappers on page 27
LDAP mappers are listeners, which are triggered by the LDAP Provider at various points, and provide another extension point to LDAP integration.

Remediate Vulnerability Dashboard

The Remediate Vulnerability Dashboard provides you with a wealth of information on the health of your system. The Remediate dashboard contains a wide variety of indicators about what is happening on your system. It is composed of three main components:

- The Metrics bar
- Vulnerabilities overview tab
- Nodes overview tab

Dashboard Metrics bar on page 32
The Metrics bar at the top of the Vulnerabilities dashboard provides a breakdown of key metrics regarding affected nodes on your network.

- Dashboard Vulnerability Overview tab on page 33
The Vulnerability Overview tab on the Puppet Remediate Vulnerabilities dashboard displays information on the number, criticality and extent to which the nodes on your network are vulnerable.

- Dashboard Node Overview tab on page 33
The Nodes overview tab on the Vulnerabilities Dashboard displays information on the nodes affected by vulnerabilities on your network.

Dashboard Metrics bar

The Metrics bar at the top of the Vulnerabilities dashboard provides a breakdown of key metrics regarding affected nodes on your network.

The Metrics bar at the top of the Vulnerabilities dashboard provides a breakdown of key metrics regarding affected nodes, including:

- **Nodes**: The total number of nodes currently discovered by Puppet Remediate. Click on this area to go to the Nodes page.

  **Note**: The result shown in this area may differ from that displayed in the Vulnerable Nodes area if your scanner duplicates nodes owing to IP address reuse. The Nodes area ignores duplicates in this instance.

- **Vulnerable Nodes**: The total number of discovered nodes currently assessed as vulnerable. Clicking on this area takes you to the Most vulnerable nodes page.

- The total number of vulnerabilities across all discovered nodes. Click this area to visit the Total Vulnerabilities page.

  **Note**: The result shown in this area may differ from that displayed in the Vulnerable Nodes area if your scanner duplicates nodes owing to IP address reuse. The Nodes area ignores duplicates in this instance.

- The risk score of the most severe vulnerabilities according to your vulnerability scanner. Click this area to go to the Vulnerability detail page for this vulnerability, where you’ll see what nodes are affected, and from where you can run remediation tasks.

- The number of vulnerabilities on the node with the most vulnerabilities. Clicking this area takes you to the Node detail page for your most vulnerable node. Again, from this page you can initiate remediate tasks to clean vulnerabilities from the node.
• The **Metrics bar** also contains a search box you can use to find out more detail on a vulnerability if you have its CVE number. Searching here take you to the **Vulnerability detail** page where you can see nodes affected by the selected vulnerability, and also run tasks to remediate it.

A full overview of the information provided by the **Vulnerabilities** dashboard **Metrics bar** is contained in the following video.

### Dashboard Vulnerability Overview tab

The **Vulnerability Overview** tab on the Puppet Remediate **Vulnerabilities dashboard** displays information on the number, criticality and extent to which the nodes on your network are vulnerable.

The **Key Statistics** area gives you an idea of the overall health of your network. Here you’ll see the total number of vulnerabilities affecting nodes within your network, as well as the total number of vulnerabilities designated as medium risk, high risk or critical.

**Note:** The result shown in this area may differ from that displayed in the **Vulnerable Nodes** area if your scanner duplicates nodes owing to IP address reuse. The **Nodes** area ignores duplicates in this instance.

The **Criticality Breakdown** chart displays the percentage of vulnerabilities in each criticality division. Mouse over each section in the chart to see more information on the number of vulnerabilities in that category affecting nodes in your network. Click on a section to view all vulnerabilities in this criticality division in the **Vulnerabilities** table.

The **Top 5 common vulnerabilities** chart displays the number of nodes compromised by the most widespread vulnerabilities on your network. Mouse over each bar for the name of the vulnerability in question. Click a bar to see more information on the selected vulnerability in the **Vulnerabilities** table.

The **Vulnerabilities table** itself provides top-level information on individual vulnerabilities:

- Click the export icon to export the **Vulnerabilities table** data in CSV format.
- Filter the content by vulnerability severity by using filter option menu and create filter button at the top of the table. If you need to filter by vulnerability name or analysis content, click **Create Filter** and select the appropriate column, choose the required operator, and add the value you want to search for.
- Sort by clicking on the appropriate column header. You can also choose which columns are displayed and which are hidden from the **Columns** drop-down menu.

Each row of the **Vulnerabilities table** provides:

- The name of the vulnerability.
- The risk score assigned by the vulnerability scanner.
- An analysis of the vulnerability provided by your vulnerability scanner. This is a description of the vulnerability threat, and the possible consequences that can occur if the vulnerability is successfully exploited.
- The number of nodes affected by the particular vulnerability.
- This table also shows the Puppet risk score. This is the vulnerability scanner risk score multiplied by the number of nodes affected by the vulnerability. The Puppet risk score lets you see straight away which vulnerabilities need to be addressed first.

Click the name of the vulnerability to go to the **Vulnerability detail** page. Here, you’ll find fuller information on the vulnerability, the nodes it affects, and advice about how to the threat. You can also launch remediation tasks from this page.

A full breakdown of the information provided by the **Vulnerabilities Overview** tab is contained in the following video:

### Dashboard Node Overview tab

The **Nodes overview** tab on the **Vulnerabilities Dashboard** displays information on the nodes affected by vulnerabilities on your network.

The **Key Statistics** area on this tab, provides information on:
- The number of accessible nodes with and without vulnerabilities.
- The total number of inaccessible nodes. In other words, the total number of nodes that Puppet Remediate does not have access credentials for.
- The number of nodes that have vulnerabilities that are deemed to be critical.

The **Vulnerable Nodes by OS** chart shows the nodes on your network affected by vulnerabilities listed by the top 5 operating systems affected. Click on a bar in this diagram to list all nodes using the selected operating system in the **Nodes table** below.

The **Nodes with vulnerabilities** diagram breaks down the percentage of nodes on your network that are affected by vulnerabilities by accessibility. Mouse over each section in the ring to see more information on the number of nodes with vulnerabilities in each accessibility type. Click on a section to list nodes by accessibility type in the **Nodes table** below.

Each row of the **Nodes table** provides:
- The resource name of the node on the network.
- The number of vulnerabilities affecting the node.
- IP address.
- Operating system name and version.
- The version of Puppet running on the node.
- The infrastructure source type.
- The node's uptime.
- The date and time a scan was last initiated on the node.

Click the export icon to export the **Nodes table** data in CSV format. Filter by node accessibility by using the filter option menu. If you need to filter by column, click **Create Filter** and select the appropriate column, choose the required operator, and add the value you want to search for.

Sort by clicking on the appropriate column header. You can also choose which columns are displayed and which are hidden from the **Columns** drop-down menu.

Clicking the **resource name** entry for a node takes you to the **Attribute** tab on that node’s **Node details** page with more information on the node. Clicking the number of vulnerabilities for the node, brings you to the **Vulnerabilities** tab on the **Node details** page. From here, you can run tasks to remediate the vulnerabilities affecting the selected node.

### Configuring Remediate

After installing Puppet Remediate, configure and administer the application by using the command line interface (CLI).

#### Command line options

The CLI provides additional administrative and troubleshooting options beyond the user interface.

```bash
docker-compose run remediate [command]
```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>View and modify the configuration.</td>
</tr>
<tr>
<td></td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>docker-compose run remediate config [command]</code></td>
</tr>
<tr>
<td></td>
<td>The following commands are used with the <code>config</code> command:</td>
</tr>
<tr>
<td></td>
<td>• <code>get-compose-file</code></td>
</tr>
<tr>
<td></td>
<td>• <code>set-override</code></td>
</tr>
<tr>
<td></td>
<td>• <code>unset-override</code></td>
</tr>
<tr>
<td></td>
<td>• <code>view</code></td>
</tr>
<tr>
<td>config get-compose-file</td>
<td>Gets the docker compose file.</td>
</tr>
<tr>
<td></td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>docker-compose run remediate config get-compose-file [flags]</code></td>
</tr>
<tr>
<td></td>
<td>In addition to the available global flags, you can use the following flag</td>
</tr>
<tr>
<td></td>
<td>with the <code>config get-compose-file</code> command:</td>
</tr>
<tr>
<td></td>
<td>• <code>--filename string</code> - The filename to output the docker compose file to.</td>
</tr>
<tr>
<td></td>
<td>(default: <code>remediate-docker-compose.yml</code>).</td>
</tr>
<tr>
<td>config set-override</td>
<td>Set configuration override for Puppet Remediate (requires restart).</td>
</tr>
<tr>
<td></td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>docker-compose run remediate config set-override [key] [value] [flags]</code></td>
</tr>
<tr>
<td></td>
<td>See <strong>Configuration overrides</strong> below for more information about how to use</td>
</tr>
<tr>
<td></td>
<td>this command.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>config unset-override</td>
<td>Unset configuration override for Puppet Remediate (requires restart).</td>
</tr>
<tr>
<td></td>
<td>Usage: docker-compose run remediate config unset-override [key]</td>
</tr>
<tr>
<td></td>
<td>See <strong>Configuration overrides</strong> below for more information about how to use this command.</td>
</tr>
<tr>
<td>config view</td>
<td>View configuration for Puppet Remediate.</td>
</tr>
<tr>
<td></td>
<td>Usage: docker-compose run remediate config view [flags]</td>
</tr>
<tr>
<td>export</td>
<td>Export the Remediate images to a <em>tar</em> file.</td>
</tr>
<tr>
<td></td>
<td>Usage: docker-compose run remediate export [flags]</td>
</tr>
<tr>
<td>generate-docs</td>
<td>Generates the CLI documentation.</td>
</tr>
<tr>
<td></td>
<td>Markdown files are created at the specified path detailing the CLI commands.</td>
</tr>
<tr>
<td></td>
<td>Usage: docker-compose run remediate generate-docs [path] [flags]</td>
</tr>
<tr>
<td>help</td>
<td>Displays the help commands.</td>
</tr>
<tr>
<td></td>
<td>Usage: docker-compose run remediate help [command] [flags]</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **import** | Import the Remediate images from a *tar* file.  
Usage:  
```
docker-compose run remediate import [flags]
```
In addition to the available global flags, you can use the following flag with the *import* command:  
- **--config** string  
The config file (default is `$HOME/.installer.yaml`) (default: `.installer.yml`)

| **logs** | Displays the application service logs.  
Usage:  
```
docker-compose run remediate logs [flags]
```
In addition to the available global flags, you can use the following flags with the *logs* command:  
- **--service** string  
- **--since** string  
  - Display logs for only this service.  
  - Show logs since timestamp (e.g. 2013-01-02T13:23:37) or relative (e.g. 42m for 42 minutes) (default "4h").

  **Note:** By default the *logs* command returns logs for the last 8 hours only. Use the **--since** flag to return logs for a time period different from the default.

| **mayday** | Creates a mayday tarball containing debug information for troubleshooting with the Puppet support team.  
Usage:  
```
docker-compose run remediate mayday [flags]
```
In addition to the available global flags, you can use the following flag with the *mayday* command:  
- **--since** string  
  - Show logs since timestamp (e.g. 2013-01-02T13:23:37). You can also use an integer with m or h. For example. 42m for 42 minutes, or 1h for 1 hour. The default is "4h".

© 2020 Puppet, Inc.
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reset</td>
<td>Deletes sources, credentials, user passwords, and resets the application to an uninstalled state.</td>
</tr>
<tr>
<td></td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td>docker-compose run remediate reset [flags]</td>
</tr>
<tr>
<td></td>
<td>In addition to the available global flags, you can use the following flag with the reset command:</td>
</tr>
<tr>
<td></td>
<td>• --quietly - Hides the splash screen.</td>
</tr>
<tr>
<td>restart</td>
<td>Restarts the application.</td>
</tr>
<tr>
<td></td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td>docker-compose run remediate restart [flags]</td>
</tr>
<tr>
<td></td>
<td>In addition to the available global flags, you can use the following flag with the restart command:</td>
</tr>
<tr>
<td></td>
<td>• --quietly - Hides the splash screen.</td>
</tr>
<tr>
<td>set-license</td>
<td>Sets a new license.</td>
</tr>
<tr>
<td></td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td>docker-compose run remediate set-license [license-file] [flags]</td>
</tr>
<tr>
<td></td>
<td>In addition to the available global flags, you can use the following flag with the set-license command:</td>
</tr>
<tr>
<td></td>
<td>• --quietly - Hides the splash screen.</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>start</strong></td>
<td>Initializes and starts the application.</td>
</tr>
<tr>
<td>Usage:</td>
<td>docker-compose run remediate start [flags]</td>
</tr>
<tr>
<td>In addition to the available global flags, you can use the following flags with the start command:</td>
<td></td>
</tr>
<tr>
<td>• --license-file string - License file (default is $PWD/ license.json)</td>
<td></td>
</tr>
<tr>
<td>• --quietly - Hides the splash screen.</td>
<td></td>
</tr>
<tr>
<td>• -o, --offline - start without internet access</td>
<td></td>
</tr>
<tr>
<td><strong>stop</strong></td>
<td>Stops the application.</td>
</tr>
<tr>
<td>Usage:</td>
<td>docker-compose run remediate stop [flags]</td>
</tr>
<tr>
<td>In addition to the available global flags, you can use the following flag with the stop command:</td>
<td></td>
</tr>
<tr>
<td>• --quietly - Hides the splash screen.</td>
<td></td>
</tr>
<tr>
<td><strong>update</strong></td>
<td>Updates to the latest version.</td>
</tr>
<tr>
<td>Usage:</td>
<td>docker-compose run remediate update [flags]</td>
</tr>
<tr>
<td>In addition to the available global flags, you can use the following flags with the update command:</td>
<td></td>
</tr>
<tr>
<td>• --quietly - Hides the splash screen.</td>
<td></td>
</tr>
<tr>
<td>• --version string - Version to update to (default &quot;latest&quot;).</td>
<td></td>
</tr>
<tr>
<td>The update command also be used with the list sub-command or with its own local flags.</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>update list</td>
<td>Shows a list of available versions.</td>
</tr>
<tr>
<td></td>
<td>Usage:</td>
</tr>
<tr>
<td></td>
<td>docker-compose run remediate update list [flags]</td>
</tr>
</tbody>
</table>

**Global flags**

In addition to the flags that can be used with individual commands, the following flags can be used with all commands:

<table>
<thead>
<tr>
<th>Flag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--config string</td>
<td>Config file (default is $HOME/.installer.yaml) (default &quot;.installer.yml&quot;)</td>
</tr>
<tr>
<td>-h, --help</td>
<td>Help for remediate</td>
</tr>
<tr>
<td></td>
<td>Use remediate [command] --help to get more information about a command.</td>
</tr>
</tbody>
</table>

**Configuration overrides**

After installing Remediate, optimize the application for your environment by configuring and tuning settings as needed.

To view the current configuration:

```
docker-compose run remediate config view
```

To override a configuration key:

```
docker-compose run remediate config set-override [key] [value]
```

To reset a configuration key:

```
docker-compose run remediate config unset-override [key]
```

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td>controller.discoveryinterval</td>
<td>The interval between each discovery run.</td>
<td>every 4h</td>
</tr>
</tbody>
</table>
### Key Description Default value

**controller.loglevel** The logging levels for the controller container.  
INFO (default)  
DEBUG  
WARN  
ERROR  
FATAL  
PANIC  

**edge.discoworkers** The maximum number of discovery processes that run in parallel.  
512  

**edge.taskworkers** The maximum number of tasks that run in parallel.  
32  

**edge.workertimeout** The time out value, in minutes, for each job.  
25  

**edge.loglevel** The logging levels for the edge container.  
WARN (default)  
INFO  
DEBUG  
ERROR  
FATAL  
PANIC  

---

## Add sources

Add your vulnerability scanner to detect and fix vulnerabilities across your infrastructure. To discover nodes, packages, and containers running on your entire infrastructure, add multiple infrastructure sources.

1. On the sidebar, click **Manage sources**.
2. Click **Add sources**, and then select a source:

   - **Vulnerability scanner:**
     - Qualys
     - Rapid7
     - Tenable.io
     - Tenable.sc
   - **Infrastructure source:**
     - Amazon Web Services
     - Google Cloud Platform
     - Microsoft Azure
     - OpenStack
     - VMware vSphere
     - IP addresses

3. Enter your authentication credentials for the source (see below for details).

   **Tip:** If you are adding a vulnerability scanner, click **Test Credentials** to check that Remediate has the correct access credentials for the scanner before you kick off a discovery.

4. Click **Discover**.

   **Note:** The quantity of data that is contained in your source affects the amount of time it takes for information to be displayed in Remediate. It might take several hours for content to be displayed if your source uses a very large data set.

**Related information**

**Run tasks** on page 52
Run ad hoc tasks on target nodes to upgrade packages, restart services, execute shell commands, or perform any other type of single-action executions on your nodes.

**Discovering and managing resources** on page 58
The **Resources** dashboard provides a high-level summary view of your infrastructure, and consists of a number of dashboard cards to drill down from for detailed information about nodes, packages, and containers.

---

**Vulnerability scanners**

Puppet Remediate integrates with Tenable, Qualys and Rapid7.

**Note:** Ask your security team for the permissions to import vulnerability scan data.

**Qualys**

Add the details for your Qualys Vulnerability Manager account.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique and descriptive name to identify this vulnerability scanner.</td>
</tr>
<tr>
<td>API server URL</td>
<td>The HTTPS URL and port number to the platform where your Qualys account is located.</td>
</tr>
<tr>
<td>Username</td>
<td>Your Qualys username to authenticate with.</td>
</tr>
</tbody>
</table>

**Note:** Qualys CE is not API compatible and therefore is not supported by Remediate. For more information, see the Qualys CE user guide.
### Remediate | Add sources | 43

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>Your Qualys password to authenticate with.</td>
</tr>
</tbody>
</table>

#### Rapid7

Add the details for your Rapid7 Nexpose (on-prem) or InsightVM (cloud) account.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique and descriptive name to identify this vulnerability scanner.</td>
</tr>
<tr>
<td>InsightVM URL</td>
<td>The HTTPS URL and port number to your InsightVM or Nexpose instance.</td>
</tr>
<tr>
<td>Username</td>
<td>Your Rapid7 username to authenticate with.</td>
</tr>
<tr>
<td>Password</td>
<td>Your Rapid7 password to authenticate with.</td>
</tr>
<tr>
<td>Enable SSL certification verification</td>
<td>To verify the signature on the SSL certificate returned by Rapid7 using your CA cert, select this option. Save your CA cert in the <code>puppet-remediate/data/vr/ca_certs</code> directory.</td>
</tr>
</tbody>
</table>

#### Tenable.io

Add the details for your Tenable.io (cloud) account.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique and descriptive name to identify this vulnerability scanner.</td>
</tr>
<tr>
<td>Access key</td>
<td>Your Tenable.io access key to authenticate with the Tenable.io API.</td>
</tr>
<tr>
<td>Secret key</td>
<td>Your Tenable.io secret key to authenticate with the Tenable.io API.</td>
</tr>
</tbody>
</table>

**Note:** You must use the Administrator role in Tenable.io to export data using the Tenable.io API.

#### Tenable.sc

Add the details for your Tenable.sc account.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique and descriptive name to identify this vulnerability scanner.</td>
</tr>
<tr>
<td>URL</td>
<td>The URL of your Tenable.sc instance.</td>
</tr>
<tr>
<td>Username</td>
<td>Your Tenable.sc account username.</td>
</tr>
<tr>
<td>Password</td>
<td>Your Tenable.sc account password.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Enable SSL certificate verification</td>
<td>Select this checkbox if you want to verify the SSL certificate returned by Tenable.sc. Remember that you must add your own CA certificate in the <code>puppet-discovery/data/vr/ca-certs</code> directory.</td>
</tr>
</tbody>
</table>

### Infrastructure sources

Discover node instances on the following infrastructure sources.

#### Amazon Web Services (AWS)

Add the AWS authentication credentials to discover the EC2 instances running on your AWS account.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique and descriptive name to identify this source.</td>
</tr>
<tr>
<td>Access key</td>
<td>The access key ID that you generated in the AWS Management Console.</td>
</tr>
<tr>
<td>Secret key</td>
<td>The secret access key that corresponds to your access key ID.</td>
</tr>
</tbody>
</table>

#### Google Cloud Platform (GCP)

Add the GCP authentication credentials to discover the node instances running on each of your accounts. The client email, the private key, the private key ID, and the project ID values are located in the service account key file (.json) you saved after generating your GCP credentials in the GCP console.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique and descriptive name to identify this source.</td>
</tr>
<tr>
<td>Client email</td>
<td>The service account email associated with your GCP account.</td>
</tr>
<tr>
<td>Private key</td>
<td>The private key you generated in the GCP console.</td>
</tr>
<tr>
<td>Private key ID</td>
<td>The private ID that corresponds to your private key.</td>
</tr>
<tr>
<td>Project ID</td>
<td>The GCP project that corresponds to your service account.</td>
</tr>
</tbody>
</table>

#### Microsoft Azure

Add the Microsoft Azure authentication credentials to discover the node instances on each of your Microsoft Azure accounts.

Make sure to log into the Azure portal and register your application. Name it **PuppetRemediate** and select the **Web app / API** application type. You must also assign the **Reader** permission to the application.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique and descriptive name to identify this source.</td>
</tr>
<tr>
<td>Subscription ID</td>
<td>The subscription ID that identifies your Azure services subscription.</td>
</tr>
<tr>
<td>Tenant ID</td>
<td>The AAD tenant ID (also known as the directory ID).</td>
</tr>
</tbody>
</table>
Remediate | Add sources | 45

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application ID</td>
<td>The Azure application ID (also known as the client ID).</td>
</tr>
<tr>
<td>Client ID</td>
<td>The client key (also known as the authentication key) generated for your application within the AAD.</td>
</tr>
</tbody>
</table>

**OpenStack**

Add the OpenStack authentication credentials to discover the node instances running on each of your OpenStack accounts.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique and descriptive name to identify this source.</td>
</tr>
<tr>
<td>Endpoint</td>
<td>The authentication URL for the identity (Keystone) service.</td>
</tr>
<tr>
<td>Username</td>
<td>Your OpenStack username to authenticate with.</td>
</tr>
<tr>
<td>Password</td>
<td>Your OpenStack password to authenticate with.</td>
</tr>
<tr>
<td>Domain name</td>
<td>The authentication domain name used to connect to OpenStack.</td>
</tr>
<tr>
<td>Tenant ID</td>
<td>The tenant ID, also known as the project ID, used for OpenStack.</td>
</tr>
</tbody>
</table>

**VMware vSphere**

Add the VMware vSphere authentication credentials to discover the node instances running on each of your VMware vSphere accounts.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique and descriptive name to identify this source.</td>
</tr>
<tr>
<td>vCenter server</td>
<td>The FQDN of the vCenter server.</td>
</tr>
<tr>
<td>vCenter username</td>
<td>The VMware vSphere username used to authenticate to the vCenter server.</td>
</tr>
<tr>
<td>vCenter password</td>
<td>The VMware vSphere password required to authenticate to the vCenter server.</td>
</tr>
</tbody>
</table>

**Network nodes**

Discover nodes by specifying an IP address range, a CIDR block, or by uploading a comma-separated IP address list.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A unique and descriptive name to identify this source.</td>
</tr>
</tbody>
</table>
Parameter | Description
--- | ---
Type | The IP address input type:
• IP range: Using the From and To fields, enter the IP address range.
• CIDR: Enter the base IP address and the subnet mask to determine the network portion of the address.
• CSV: Upload a comma-separated list of:
  • IP addresses.
  • Hyphen-separated IP ranges.
  • CIDR notations.

Adding node credentials

Add credentials to authenticate with nodes and fix vulnerabilities or gain insights into discovered resources. Each credential is encrypted and stored securely in the vault.

The two authentication methods are Secure Shell (SSH) authentication with Linux nodes on port 22, and Windows Remote Management (WinRM) authentication with Windows nodes on ports 5986 and 5985.

**CAUTION:** Using each node credential you provide, Remediate attempts to authenticate with each discovered node until a successful authentication is achieved. This process repeats every four hours, using previously successful credentials first. Depending on the configuration of your network management and security sensors, Remediate activities might trigger alerts or an active response.

- **Secure Shell (SSH)** on page 46
  The two types of SSH authentication are username and password, using negotiated encryption, and private key files, using asymmetric encryption.
  - **Windows Remote Management (WinRM)** on page 47
    To authenticate with Windows nodes, Puppet Remediate uses NTLM authentication over HTTPS on port 5986. When enabled, Remediate falls back to using NTLM authentication over HTTP on port 5985, if the default authentication fails.

**Secure Shell (SSH)**

The two types of SSH authentication are username and password, using negotiated encryption, and private key files, using asymmetric encryption.

**Tip:** As a dual-factor authentication with nodes, it’s recommended to use SSH private key files and to include a username and passphrase for each file. Using an SSH private key file, rather than an SSH username and password, is considered more secure against potential compromises on remote nodes because the password is not sent over the network.

When using SSH authentication to discover resources running on Linux nodes, there are a number of prerequisites:

- To install the Puppet agent on nodes, your SSH credentials must be for the root account.
- To discover containers on nodes, your SSH credentials must be for the root account or an account that is a member of the Docker group.
Add SSH private key files

Upload an SSH private key file to discover resources, and to run tasks on your Linux hosts.

1. On the sidebar, click Manage credentials.
2. Click Add credentials and then click SSH private key file.
3. Click Browse, select your files, and then click Open.
4. Click Configure keys.
5. In the Name field enter a unique and descriptive name.
6. Assign an individual scope, or both, to the credential:
   - Discover resources on nodes: This credential scope is valid only for discovering resources on your Linux nodes.
   - Remediate vulnerabilities: This credential is valid only for running tasks on your Linux nodes. When this individual scope is selected, no attempts are made to discover resources.
   - Escalate privileges to root: When required to run tasks on nodes, sudo escalate non-root account privileges to root. Privilege escalation occurs if the first attempt to run a task fails when using non-root privileges.
7. In the Username field, enter your SSH username.
8. In the Passphrase field, enter your SSH passphrase, or leave it blank if your key is not encrypted.
9. Click Add keys.

Add SSH username and password

Add an SSH username and password to discover resources, and to run tasks on your Linux hosts.

1. On the sidebar, click Manage credentials.
2. Click SSH credential.
3. In the Name field, enter a unique and descriptive name.
4. Assign an individual scope, or both, to the credential:
   - Discover resources on nodes: This credential scope is valid only for discovering resources on your Linux nodes.
   - Remediate vulnerabilities: This credential is valid only for running tasks on your Linux nodes. When this individual scope is selected, no attempts are made to discover resources.
   - Escalate privileges to root: When required to run tasks on nodes, sudo escalate non-root account privileges to root. Privilege escalation occurs if the first attempt to run a task fails when using non-root privileges
5. In the Username field, enter your SSH username.
6. In the Password field, enter your SSH password, and then click Add credential.

Windows Remote Management (WinRM)

To authenticate with Windows nodes, Puppet Remediate uses NTLM authentication over HTTPS on port 5986. When enabled, Remediate falls back to using NTLM authentication over HTTP on port 5985, if the default authentication fails.

To discover resources on your Windows hosts, you must enable WinRM access on each host by running the following commands:

```
winrm quickconfig
y
winrm set winrm/config/winrs '(@{MaxMemoryPerShellMB="1024"})'
```
CAUTION:

To enable the HTTP fallback option (see step 6 below), include the `winrm set winrm/config/service '@{AllowUnencrypted="true"}'` command. This means that all Puppet Remediate commands and responses, not including credentials, are sent in plaintext over the network.

Note:

Your Windows user account must be a member of either the local administrator group or the `WinRMRemoteWMIUsers_` group. For more information, see the Default Group Access section in the Microsoft Windows Remote Management documentation.

Add WinRM credentials

1. On the sidebar, click Manage credentials.
2. Click WinRM credential.
3. In the Name field, enter a unique and descriptive name.
4. Assign an individual scope, or both, to the credential:
   - Discover resources on nodes: This credential scope is valid only for discovering resources on your Windows nodes.
   - Remediate vulnerabilities: This credential is valid only for running tasks on your Windows nodes. When this individual scope is selected, no attempts are made to discover resources.
5. Select HTTP fallback to permit using authentication over HTTP, if the default authentication over HTTPS fails.
6. Click Add credential.

Prioritizing vulnerabilities

To improve your infrastructure security, it is crucial to analyze the vulnerabilities detected during a vulnerability scan. By monitoring the number of vulnerabilities and affected nodes, the risk score, and the analysis of each vulnerability, you can prioritize its remediation and manage your security resources efficiently.

To help you examine the severity of the vulnerabilities in the context of each node, the Vulnerabilities dashboard provides a summary view of the most critical ones detected on your infrastructure.

Tip: By default, the dashboard automatically refreshes every 15 minutes. To change to a manual refresh, click Manage sources > Automatically refresh dashboard. The manual refresh dashboard option appears at the top of the dashboard.

Vulnerable nodes needing immediate attention

It’s a top priority to fix a critical vulnerability in a node that’s considered extremely important. However, remediating the same vulnerability might not be a top priority if it’s present on a node of medium or low importance.

There are several ways you can explore vulnerable nodes from the Vulnerabilities dashboard:

- From the metrics bar at the top of the Vulnerabilities dashboard:
  - The Vulnerable Nodes area of the metrics bar displays how many nodes on your network are vulnerable. Click this area to go to the Most vulnerable nodes page. The Most vulnerable nodes page lists and orders nodes ranked by the number of vulnerabilities detected on each node. You can further filter by resource name, IP address, operating system, and OS version, or any combination of these. Select an individual node to go its Most vulnerable nodes page where you can run tasks to remediate the vulnerabilities affecting it.
  - The Most Vulnerable Node area of the metrics bar displays the number of vulnerabilities affecting the most vulnerable node on your system. Click this area to go to the Most vulnerable nodes page for this particular node. You can run task directly from this page to remediate the vulnerabilities affecting this node.
• From the **Nodes** tab:
  • Sort the **Nodes** table by **Vulnerability count** to list the nodes with most vulnerabilities.

**Vulnerabilities posing the highest risk to my infrastructure**

A risk is a potential for loss, damage, or destruction of a node as a result of a threat exploiting a vulnerability. The risk score assigned to each vulnerability indicates the potential danger it poses to each node. It is based on the impact and possibility of exploit.

There are several ways you can explore high risk vulnerabilities from the **Vulnerabilities** dashboard:

• On the metrics bar at the top of the **Vulnerabilities** dashboard:
  • The **Highest Risk Score** area displays the risk score (as defined by your scanner) for the most serious vulnerability affecting your network. Click this area to go to the **Vulnerability detail** page for this vulnerability, where you can information on the number of nodes with the vulnerability, an analysis of the threat, and information on the steps needed to remediate the problem (if provided by your scanner).

• On the **Vulnerabilities** tab:
  • The **Criticality breakdown** chart displays the percentage of vulnerabilities in each criticality division affecting nodes on your network. Mouse over each section in the chart to see more information on the number of vulnerabilities in that category affecting nodes in your network. Click on a section to view all vulnerabilities in this criticality division listed in the **Vulnerabilities** table.
  • Sort the **Vulnerabilities** table by **Risk score** to list vulnerabilities by highest risk score.

The **Highest risk vulnerabilities** page lists and orders vulnerabilities ranked by the assigned risk score and by the number of nodes the vulnerability was detected on.

**Important:** The risk score is assigned by your vulnerability scanner.

**Vulnerabilities affecting a large percentage of my infrastructure**

On the **Vulnerabilities** tab of the Vulnerabilities dashboard, the **Top 5 common vulnerabilities** chart lists the top 5 vulnerabilities detected on nodes running on your infrastructure. Each vulnerability on this chart is ranked by the number of nodes it was detected on. Click a bar in this area to see all the nodes affected by the selected vulnerability in the **Vulnerabilities** table.

The **Hosts affected** column in the **Vulnerabilities** table lists vulnerabilities ranked by the number of nodes the vulnerability is detected on. Sort this column to order vulnerabilities by the number of nodes they affect.

• Use the **Vulnerability details** on page 49
  Use the **Vulnerability details** page to identify which nodes are affected by the vulnerability, and using the analysis and remediation details determine which task to run on the nodes to help fix the specific vulnerability.

• **Filtering and exporting data** on page 50
  Create custom filters and customize table views to view vulnerability data most important to you, or for backup purposes, export data to a CSV file.

**Viewing vulnerability details**

Use the **Vulnerability details** page to identify which nodes are affected by the vulnerability, and using the analysis and remediation details determine which task to run on the nodes to help fix the specific vulnerability.

**Metrics**

Displayed on each vulnerability details page are metrics relating to the vulnerability:

• Risk score - the risk score value assigned to the vulnerability by your vulnerability scanner.
• Nodes affected - the number of nodes the vulnerability was detected on.
• Infrastructure affected - the percentage of your entire infrastructure the vulnerability affects.
**CVE details**

If provided by your vulnerability scanner, the following CVE details are displayed for each vulnerability.

<table>
<thead>
<tr>
<th>CVE detail</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE ID</td>
<td>The CVE ID is an unique identifier for a security vulnerability and is the number sequence of a CVE entry. For example, CVE-2019-0601 includes the CVE prefix, the year that the CVE ID was assigned or made public, and the sequence of numbers.</td>
</tr>
<tr>
<td>CVSS base score</td>
<td>Depending on how easy it is to exploit the vulnerability and how damaging it can be, each vulnerability is assigned a base score in the Common Vulnerability Scoring System (CVSS) which ranges from zero to ten.</td>
</tr>
<tr>
<td>CVSS V3 base score</td>
<td></td>
</tr>
<tr>
<td>CVSS temporal score</td>
<td>The temporal score is calculated using metrics on how likely the vulnerability will be exploited, on how easy the vulnerability is to fix, and on how confidently it can be said that a vulnerability exists.</td>
</tr>
<tr>
<td>CVSS V3 temporal score</td>
<td></td>
</tr>
</tbody>
</table>

**Analysis**

The analysis provided by your vulnerability scanner is a description of the vulnerability threat, and the possible consequences that can occur if the vulnerability is successfully exploited.

**Remediation**

The remediation details provided by your vulnerability scanner is a brief summary of how you can fix the vulnerability. For information on how to fix a vulnerability, see Remediating Vulnerabilities.

**Filtering and exporting data**

Create custom filters and customize table views to view vulnerability data most important to you, or for backup purposes, export data to a CSV file.

**Create custom filters**

You can create a list of nodes or vulnerabilities for further investigation by creating a custom filter with multiple criteria.

1. On any listing or details page, click **Create filter**.
2. Select a **Field**.
3. Select an **Operator**:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Filter for resources where the value of the field:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>is an exact match to the value you specify.</td>
</tr>
<tr>
<td>Not equal to</td>
<td>is anything except an exact match to the value you specify.</td>
</tr>
<tr>
<td>Contains</td>
<td>contains the value you specify.</td>
</tr>
<tr>
<td>Does not contain</td>
<td>does not contain the value you specify.</td>
</tr>
</tbody>
</table>

4. Enter a **Value**.
5. Click **Add filter**.
6. To add criteria to the filter, click **Add another filter**.
7. As needed, repeat these steps to add additional filters.
8. To display the filtered list, click **Apply all**.

**Customize table views**

Customize each table view by showing or hiding table columns on any of the node or vulnerability pages.

1. On any listings or details page, click **Columns +/-**.
   By default, all available table columns are selected.
2. To hide a table column, click the column header to remove the selection indicator (✓).
3. To confirm your selections, click **Apply**.

**Export data**

To keep a backup of data relating to nodes or vulnerabilities, download a CSV file that contains the same information as the page you are currently viewing, including all filter selections.

To download resource data as a .csv file, click **Export**. The filename is `<ui_page>_<timestamp>.csv`, for example `pd_export_highest_risk_vulnerabilities_2019-06-17_1054.csv`.

**Remediating vulnerabilities**

To help fix vulnerabilities, run ad hoc tasks on target Linux and Windows nodes to install Puppet agents, manage packages or system services, or to execute shell commands. Upload your scripts to be converted into tasks, or upload Puppet module containing a task.

- **Upload scripts** on page 51
  Uploaded Linux and Windows scripts are converted into tasks to run on vulnerable nodes and help fix vulnerabilities.
- **Upload modules** on page 52
  To help fix vulnerabilities on nodes, upload modules published on the Puppet Forge. The module must contain a task.
- **Run tasks** on page 52
  Run ad hoc tasks on target nodes to upgrade packages, restart services, execute shell commands, or perform any other type of single-action executions on your nodes.
- **Remediate your top vulnerabilities** on page 56
  The **Top vulnerabilities** table on the **Vulnerabilities** page lists the vulnerabilities that you need to address most urgently. The ranking on this table is based on a combination of risk score and number of network nodes affected.
- **Remediate high risk vulnerabilities** on page 57
  The **Highest risk vulnerabilities** chart on the Remediate dashboard gives you visibility of the most serious vulnerabilities affecting your system.
- **Using tasks tutorial** on page 57
  The built-in tasks are simple to set up in Remediate. You can even create your own custom tasks.

**Upload scripts**

Uploaded Linux and Windows scripts are converted into tasks to run on vulnerable nodes and help fix vulnerabilities.

You can upload any script that can be executed from PowerShell on Windows, or a shell on Linux. You must ensure the scripting language is installed on the nodes where you want to run your task as Remediate does not check for this, and you will get an error if the relevant language interpreter is not present.

1. On the sidebar, click **Manage tasks**, and then click **Add tasks**.
2. On the **Upload file** page, select **Add Linux script** or **Add Windows script**.
3. Select your script, and then click **Open**.
   
   Tip: If you upload both a Linux and Windows script, they are converted and combined into one task that can run on either operating system.

4. Click **Add details**.

5. On the **Add details** page, enter a name and description for the task.

6. Click **Save task**.

   The **Settings** page appears, and your task is now available and is listed in the tasks table.

**Related information**

**Run tasks** on page 52

Run ad hoc tasks on target nodes to upgrade packages, restart services, execute shell commands, or perform any other type of single-action executions on your nodes.

---

**Upload modules**

To help fix vulnerabilities on nodes, upload modules published on the Puppet Forge. The module must contain a task.

Modules are self-contained bundles of code and data. Each module manages a specific task in your infrastructure, such as installing and configuring a piece of software. For more information, see **Module fundamentals**.

1. Download a module from the **Forge**.

2. On the sidebar, click **Manage tasks**, and then click **Add tasks**.

3. Click **Upload a module**, choose the module you just downloaded from the Forge.

   **Important**: The name of your module needs to follow the default download format.

4. Click **Add details**.

5. Click **Save task**.

   The **Settings** page appears, and your tasks from the module is now available and is listed in the tasks table.

   **Important**: Your module can’t exceed 1Mb in size.

**Related information**

**Run tasks** on page 52

Run ad hoc tasks on target nodes to upgrade packages, restart services, execute shell commands, or perform any other type of single-action executions on your nodes.

---

**Run tasks**

Run ad hoc tasks on target nodes to upgrade packages, restart services, execute shell commands, or perform any other type of single-action executions on your nodes.

Tasks are similar to scripts, but they are kept in modules and can have metadata. For more information, see **Bolt tasks**.

When selecting to run a task from either a details or listing page, some of the selected nodes might not be eligible to run the task on. An eligible node must be accessible using the SSH or WinRM credentials you added, or if running the install Puppet agent task, an eligible node doesn't have the agent installed.

1. On the sidebar, click **Run tasks**, and select the task:
   
   - Install Puppet agent.
   - Run a shell command.
   - Manage package.
   - Manage service.

2. Enter the values for the task parameters (see below).

3. Click **Select nodes**.
4. Select the nodes to run the task on, and click **Select credentials**.
5. Select the credentials for accessing the hosts, and click **Review task summary**.
6. Review the tasks details, your credentials, the list of target nodes, and click **Run task**.

    **Tip:** To view the status of the task run, on the left hand pane, click **Recent events**.

## Installing Puppet agents

Install a Puppet Enterprise agent to regularly pull configuration catalogs from a Puppet master, and apply them to your target Linux or Windows nodes. The agent maintains the node configuration you want.

Although Puppet Remediate is not integrated with Puppet or Puppet Enterprise, and you do not need to have Puppet or Puppet Enterprise installed to use Remediate, you can use Remediate to install a Puppet Enterprise agent to work with a Puppet master.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacert_content</td>
<td>The master CA certificate content (optional). If not specified, the master's identity is not verified during the agent installation.</td>
</tr>
<tr>
<td>certname</td>
<td>The unique certificate name for the Puppet agent (optional).</td>
</tr>
<tr>
<td>custom_attribute</td>
<td>The custom attribute setting added to puppet.conf and included in the custom_attributes section of csr_attributes.yaml.</td>
</tr>
<tr>
<td>Important:</td>
<td>Values must be entered as an array.</td>
</tr>
<tr>
<td>dns_alt_names</td>
<td>The alternative DNS names for generating the agent certificate.</td>
</tr>
<tr>
<td>environment</td>
<td>The environment to install with the Puppet agent (optional).</td>
</tr>
<tr>
<td>extension_request</td>
<td>The extension attribute setting added to puppet.conf and included in the extension_requests section of csr_attributes.yaml.</td>
</tr>
<tr>
<td>Important:</td>
<td>Values must be entered as an array.</td>
</tr>
<tr>
<td>master</td>
<td>The required hostname for the Puppet master. The FQDN must be fully resolvable by the node on which you're installing the agent.</td>
</tr>
</tbody>
</table>

## Running shell commands

Execute an arbitrary shell command on discovered nodes without installing an agent.

**Note:** If you are using Remediate on Linux, the Remediate user must be added without a password to the `/etc/sudoers` file and configured to not require a `tty`. For example:

```
Defaults:myuser !requiretty
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>command</td>
<td>The command to execute on the target nodes.</td>
</tr>
<tr>
<td></td>
<td>Linux example:</td>
</tr>
</tbody>
</table>
|           | ```bash
          echo "Hello, World ${USER}" > /hello.txt
``` |
|           | Windows example: |
|           | ```bash
          echo "Hello, World $env:UserName" > C:\hello.txt
``` |
| failonfail| By default, the task fails when the command returns a non-zero. To disable this default setting, select the checkbox. |
| interleave| By default, content from stdout and stderr is interleaved. To disable this default setting, select the checkbox. |

**Managing packages**

Install, upgrade, or uninstall packages on discovered nodes without installing an agent.

**Remember:**

To run the manage package task on target hosts, the following package management systems are required:

- APT or YUM for Linux hosts.
- Chocolatey for Windows hosts.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **action** | The action to be applied to the package:  
  - **install** the package. To install a specific version of the package, specify the value in the **version** parameter. If installing the package for the first time, the package repository on each target node must have the package stored.  
  - **uninstall** the package. To uninstall a specific version of the package, specify the value in the **version** parameter.  
  - **upgrade** the version of the package. This is particularly useful for upgrading vulnerable packages to secure versions. To upgrade to a specific version, choose **install**, and specify the value in the **version** parameter. |
| **name** | The name of the package. |
| **provider** | The name of the provider to use for managing or inspecting the package. |
| **version** | The version, and if applicable, the release value of the package. A version number range or a semver pattern are not permitted. For example, to install the `bash-4.1.2-29.el6.x86_64.rpm` package, enter `4.1.2-29.el6`.  
  **Tip:** To install or upgrade to the latest version of a package, leave the **version** parameter blank. |

**Managing system services**

Manage system services on discovered hosts without installing an agent.
Remediate your top vulnerabilities

The Top vulnerabilities table on the Vulnerabilities page lists the vulnerabilities that you need to address most urgently. The ranking on this table is based on a combination of risk score and number of network nodes affected.

To remediate a top vulnerability:

1. On the Remediate Vulnerabilities dashboard Vulnerability overview tab, click the relevant bar in the Top 5 common vulnerabilities chart.
   Information on the selected vulnerability appears in the Vulnerabilities table.
   Tip: Alternatively, sort the Vulnerabilities table by Hosts affected to list the vulnerabilities affecting the greatest number of nodes in the Vulnerabilities table.

2. Select the vulnerability you want to remediate in the Vulnerabilities table.

3. On the Vulnerability detail page, review the analysis and remediation information that is displayed for the selected vulnerability.
   • The Analysis section provides you with information on the nature of the vulnerability and the threat it poses.
   • The Remediation section gives practical information on the remediation task you need to carry out to block the threat (where provided by your vulnerability scanner).

4. In the Nodes affected table, select the nodes to which you want the remediation task to apply.
   Note:
   If Remediate does not have the credentials to apply a task to a node, it is not selectable in the Nodes affected table.

5. Click Run Task and select the appropriate task type from the drop-down list.

6. On the Configure task page, configure the task as required. Instructions on remediation for the selected vulnerability are visible on this page. Click Review Nodes when you are done.

7. On the Review nodes page, ensure that all the nodes to which you want to apply the task are selected. When ready, click Select credentials.

8. On the Select credentials page, select the credentials that allow you to run the task on the selected nodes, and click Review task summary.
9. On the **Review and run task** page, verify that the task summary information is correct, and click **Run task**.

   A confirmation message appears at the top of the page, confirming that the task type that is now running and how many nodes it affects.

   **Note:** The changes made by the task if successful will only be reflected here after your next security scan, so don’t worry if you see no updates at this point.

**Remediate high risk vulnerabilities**

The **Highest risk vulnerabilities** chart on the Remediate dashboard gives you visibility of the most serious vulnerabilities affecting your system.

To remediate a high risk vulnerability:

1. On the Remediate Vulnerabilities dashboard **Vulnerability overview** tab, click the **Critical** (or **High** if **Critical** does not exist) segment in the **Criticality breakdown** chart.

   **Tip:** Alternatively, select the appropriate criticality level from the filter drop-down menu at the top of the **Vulnerabilities** table. You can also sort the **Vulnerabilities** table by **Risk score** to discover the highest risk vulnerabilities.

   A list of critical vulnerabilities appears in the **Vulnerabilities** table.

2. Select the vulnerability you want to remediate in the **Vulnerabilities** table.

3. On the **Vulnerability detail** page, review the analysis and remediation information that is displayed for the selected vulnerability.

   - The **Analysis** section provides you with information on the nature of the vulnerability and the threat it poses.
   - The **Remediation** section gives practical information on the remediation task you need to carry out to block the threat (where provided by your vulnerability scanner).

4. In the **Nodes affected** table, select the nodes to which you want the remediation task to apply.

   **Note:**

   If Remediate does not have the credentials to apply a task to a node, it is not selectable in the **Nodes affected** table.

5. Click **Run Task** and select the appropriate task type from the drop-down list.

6. On the **Configure task** page, configure the task as required. Instructions on remediation for the selected vulnerability are visible on this page. Click **Review Nodes** when you are done.

7. On the **Review nodes** page, ensure that all the nodes to which you want to apply the task are selected. When ready, click **Select credentials**.

8. On the **Select credentials** page, select the credentials that allow you to run the task on the selected nodes, and click **Review task summary**.

9. On the **Review and run task** page, verify that the task summary information is correct, and click **Run task**.

   A confirmation message appears at the top of the page, confirming that the task type that is now running and how many nodes it affects.

   **Note:** The changes made by the task if successful will only be reflected here after your next security scan, so don’t worry if you see no updates at this point.

**Using tasks tutorial**

The built-in tasks are simple to set up in Remediate. You can even create your own custom tasks.

After identifying a vulnerability, or set of vulnerabilities, you need to remediate, you can target the afflicted nodes with a Task. A Task is a script that the target node is able to execute, wrapped in metadata Remediate can consume.

Learn how to remediate a vulnerability using both built-in and custom tasks in this short tutorial.
Discovering and managing resources

The **Resources** dashboard provides a high-level summary view of your infrastructure, and consists of a number of dashboard cards to drill down from for detailed information about nodes, packages, and containers.

**Remember:** To discover resources running on each node, you must provide SSH or WinRM credentials. See [Adding node credentials](#).

- **Discovering resources** on page 58
  Discover node resources running on your infrastructure source account. Add your node credentials and discover node attributes, the system services, the users and groups belonging to each node, along with packages, tags, and containers.

- **Viewing resource details** on page 60
  Puppet Remediate groups attributes associated with each discovered resource into a number of different facets to give you even more insights. A facet represents a set of related attributes, each one independently maintained, that describe a certain aspect of a discovered node.

- **Filtering and exporting data** on page 68
  Create custom filters and customize table views to view resource data most important to you, or for backup purposes, export data to a CSV file.

**Related information**

**Add sources** on page 41
Add your vulnerability scanner to detect and fix vulnerabilities across your infrastructure. To discover nodes, packages, and containers running on your entire infrastructure, add multiple infrastructure sources.

---

Discovering resources

Discover node resources running on your infrastructure source account. Add your node credentials and discover node attributes, the system services, the users and groups belonging to each node, along with packages, tags, and containers.

**What resources can I discover?**

Depending on the level of credentials you enter, this table lists the depth of resources that you can discover.

<table>
<thead>
<tr>
<th>Sources and credentials</th>
<th>Discovered resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Infrastructure source.</td>
<td>• Nodes, including node attributes.</td>
</tr>
<tr>
<td>• Infrastructure source.</td>
<td>• Nodes, including node attributes.</td>
</tr>
<tr>
<td>• SSH or WinRM credentials.</td>
<td>• Services</td>
</tr>
<tr>
<td></td>
<td>• Users</td>
</tr>
<tr>
<td></td>
<td>• Groups</td>
</tr>
<tr>
<td></td>
<td>• Tags</td>
</tr>
<tr>
<td></td>
<td>• Packages, including package attributes.</td>
</tr>
<tr>
<td></td>
<td>• Container images, including container attributes.</td>
</tr>
</tbody>
</table>
**Manually triggering a scan**

In addition to the scheduled scanning of resources, you can also manually trigger a rescan of all or selected resources. On the **Manage Sources** page, click **Discover All** to rescan all sources listed in the **Sources** table.

Alternatively, to trigger a rescan for an individual resource, click the icon on the row assigned to it in the **Sources** table.

**Deleting a resource**

On the **Manage Sources** page, click the icon on the row assigned to the resource you want to delete in the **Sources** table.

⚠️ **CAUTION:** After you delete a resource by this method, it is not automatically rediscovered by the scheduled discovery. You must manually add it again. For information on how to add a resource, see **Add sources**.

**Nodes**

A number of nodes summary pages give you information about node instances across your entire infrastructure, and each page queries the data platform for specific attributes belonging to each node. For more information on nodes, see **Node attributes**.

**Tip:** Click a card to drill down and view the detailed list.

<table>
<thead>
<tr>
<th>Node summary page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nodes</td>
<td>Click the <strong>Nodes</strong> dashboard card to see a list of on-premises and cloud nodes running on your infrastructure.</td>
</tr>
<tr>
<td>Packages</td>
<td>The total number of packages installed on discovered hosts.</td>
</tr>
<tr>
<td>Inaccessible nodes</td>
<td>The total number of discovered nodes running on your infrastructure that could not be accessed due to entering invalid credentials or due to configuration issues.</td>
</tr>
<tr>
<td>AWS nodes</td>
<td>Click the <strong>AWS nodes</strong> dashboard card to see a list of EC2 instances running on your AWS account.</td>
</tr>
<tr>
<td>VMware nodes</td>
<td>The total number of nodes running on your VMware vSphere account.</td>
</tr>
<tr>
<td>OpenStack nodes</td>
<td>The total number of nodes running on your OpenStack account.</td>
</tr>
<tr>
<td>Microsoft Azure nodes</td>
<td>The total number of nodes running on your Microsoft Azure account.</td>
</tr>
<tr>
<td>Google Cloud Platform nodes</td>
<td>The total number of nodes running on your GCP account.</td>
</tr>
<tr>
<td>Nodes with Puppet</td>
<td>The percentage and total number of discovered nodes with or without Puppet installed, as well as unknown hosts.</td>
</tr>
<tr>
<td>Uptime less than 24 hours</td>
<td>The percentage and total number of nodes with an uptime with less than, or greater than, 24 hours.</td>
</tr>
<tr>
<td>Top operating systems</td>
<td>The top six operating systems installed on discovered hosts.</td>
</tr>
<tr>
<td>Linux distributions</td>
<td>The percentage and total number of Linux nodes categorized by the distribution installed.</td>
</tr>
</tbody>
</table>
### Packages

Click the Packages dashboard card to see a list of packages in use across your infrastructure by name, version, and manager, as well as the number of instances of each package. For more information about packages, see Package attributes.

### Containers

Click the Containers, or the Top containers by image, dashboard card to see a list of container instances running on your infrastructure. For more information about containers, see Container attributes.

### Related information

Add sources on page 41
Add your vulnerability scanner to detect and fix vulnerabilities across your infrastructure. To discover nodes, packages, and containers running on your entire infrastructure, add multiple infrastructure sources.

### Viewing resource details

Puppet Remediate groups attributes associated with each discovered resource into a number of different facets to give you even more insights. A facet represents a set of related attributes, each one independently maintained, that describe a certain aspect of a discovered node.

### Node attributes

Puppet Remediate considers each node as a network accessible resource, whether it's physical or virtual, and discovers attributes that define the host's state and properties.

### Amazon Web Services

Discover EC2 instances running on your AWS account, including instance attributes, packages, and containers.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Facet::attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMI Name</td>
<td>aws_ec2Instance::name</td>
<td>The name of the Amazon Machine Image (AMI).</td>
</tr>
<tr>
<td>Availability zone</td>
<td>aws_ec2Instance::Placement</td>
<td>The availability zone of the instance.</td>
</tr>
<tr>
<td>Creation Date</td>
<td>aws_ec2Instance::CreationDate</td>
<td>The date and time the resource was created.</td>
</tr>
<tr>
<td>Description</td>
<td>aws_ec2Instance::Description</td>
<td>The description of the image.</td>
</tr>
<tr>
<td>Attribute</td>
<td>Facet::attribute</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Image type</td>
<td>aws_ec2Instance::imageType</td>
<td>The type of image:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• machine</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• kernel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ramdisk</td>
</tr>
<tr>
<td>Instance type</td>
<td>aws_ec2Instance::InstanceType</td>
<td>The instance type.</td>
</tr>
<tr>
<td>Key name</td>
<td>aws_ec2Instance::KeyName</td>
<td>The name of the key pair.</td>
</tr>
<tr>
<td>Launch time</td>
<td>aws_ec2Instance::LaunchTime</td>
<td>The date and time the resource was launched.</td>
</tr>
<tr>
<td>Name</td>
<td>host::name</td>
<td>The name of the host.</td>
</tr>
<tr>
<td>Operating system</td>
<td>computeHost::os</td>
<td>The operating system running on the instance.</td>
</tr>
<tr>
<td>OS version</td>
<td>computeHost::osVersion</td>
<td>The version of the operating system running on the instance.</td>
</tr>
<tr>
<td>Owner ID</td>
<td>aws_ec2Instance::OwnerId</td>
<td>The account ID of the image owner.</td>
</tr>
<tr>
<td>Private DNS Name</td>
<td>aws_ec2Instance::PrivateDnsName</td>
<td>The private DNS name.</td>
</tr>
<tr>
<td>Private IP address</td>
<td>aws_ec2Instance::PrivateIp</td>
<td>The private IP version 4 address.</td>
</tr>
<tr>
<td>Public DNS Name</td>
<td>aws_ec2Instance::PublicDnsName</td>
<td>Fully qualified public DNS hostname.</td>
</tr>
<tr>
<td>Public IP address</td>
<td>aws_ec2Instance::PublicIp</td>
<td>The public IP version 4 address.</td>
</tr>
<tr>
<td>Region</td>
<td>cloudResource::region</td>
<td>The region where the instance exists.</td>
</tr>
<tr>
<td>Security groups</td>
<td>aws_ec2Instance::SecurityGroups</td>
<td>The security group associated with the instance.</td>
</tr>
<tr>
<td>State</td>
<td>aws_ec2Instance::MonitoringState</td>
<td>Indicates whether monitoring is enabled:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• disabling</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• enabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• pending</td>
</tr>
<tr>
<td>Status</td>
<td>cloudResource_status</td>
<td>The current status of the instance.</td>
</tr>
<tr>
<td>State transition reason</td>
<td>aws_ec2Instance::StateTransitionReason</td>
<td>Describes the state change.</td>
</tr>
<tr>
<td>Subnet ID</td>
<td>aws_ec2Instance::SubnetId</td>
<td>The ID of the subnet.</td>
</tr>
<tr>
<td>Tags</td>
<td>tags::name</td>
<td>The tag assigned to the AWS resource.</td>
</tr>
<tr>
<td></td>
<td>tags::value</td>
<td></td>
</tr>
</tbody>
</table>
### Attribute | Facet::attribute | Description
--- | --- | ---
Uptime | computeHost::uptime | The uptime for the instance, in seconds.
VM Image | vm::image | The image identification number.
VM key name | vm::keyName | The unique identifier for the file in storage.
VM virtualization type | vm::virtualizationType | The virtualization type:
• pv
• hvm
VPC ID | aws_ec2Instance::VpcId | The identification for the Virtual Private Cloud (VPC).

---

**Microsoft Azure**

Discover compute instances running on your Azure account, including instance attributes, packages, and containers.

### Attribute | Facet::attribute | Description
--- | --- | ---
Created at | azure_computeInstance::Created | The date and time the resource was created.
Data disks | azure_computeInstance::DataDisks | The parameters used to add the data disk to the virtual machine.
ID | azure_computeInstance::ID | The resource ID.
Image | azure_computeInstance::Image | The virtual machine image.
Instance type | azure_computeInstance::InstanceType | The type of instance.
IP address | host::privateIPv4 | The private IP version 4 address.
Key name | azure_computeInstance::KeyName | The keypair name.
Last scan | | The Last scan of the host represented by the data platform timestamp in UTC.
Location | azure_computeInstance::Location | The location of the resource.
MAC address | azure_computeInstance::MacAddress | The MAC address assigned to the resource.
Name | host::name | The name of the host.
Operating system | computeHost::os | The operating system running on the host.
OS disk | azure_computeInstance::OSDisk | The name of the operating system used by the virtual machine.
OS version | computeHost::osVersion | The version of the operating system running on the host.
Puppet version | computeHost::puppetVersion | The version of Puppet installed.
Resource group | azure_computeInstance::ResourceGroup | The name of the resource group.
Size | azure_computeInstance::VMSize | The size type of the virtual machine.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Facet::attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>azure_computeInstance::Status</td>
<td>The status of the virtual machine.</td>
</tr>
<tr>
<td>Subscription ID</td>
<td>azure_computeInstance::SubscriptionID</td>
<td>The subscription ID.</td>
</tr>
<tr>
<td>Tags</td>
<td>azure_computeInstance::Tags</td>
<td>A list of tags relevant to the content of the image.</td>
</tr>
<tr>
<td>Uptime</td>
<td>computeHost::Uptime</td>
<td>The uptime for the host, in seconds.</td>
</tr>
</tbody>
</table>

### OpenStack

Discover hosts running on your OpenStack account, including host attributes, packages, and containers.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Facet::attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server ID</td>
<td>os_compute::ServerID</td>
<td>The UUID of the server.</td>
</tr>
<tr>
<td>Name</td>
<td>host::hostname</td>
<td></td>
</tr>
<tr>
<td>IP address</td>
<td>host::privateIPv4</td>
<td>The private IP version 4 address.</td>
</tr>
<tr>
<td>Operating system</td>
<td>computeHost::os</td>
<td>The operating system running on the host.</td>
</tr>
<tr>
<td>OS version</td>
<td>computeHost::osVersion</td>
<td>The version of the operating system running on the host.</td>
</tr>
<tr>
<td>Uptime</td>
<td>computeHost::uptime</td>
<td>The uptime for the host, in seconds.</td>
</tr>
<tr>
<td>Last scan</td>
<td></td>
<td>The Last scan of the host represented by the data platform timestamp in UTC.</td>
</tr>
<tr>
<td>Puppet version</td>
<td>computeHost::puppetVersion</td>
<td>The version of Puppet installed.</td>
</tr>
<tr>
<td>Created on</td>
<td>os_compute::ServerCreated</td>
<td>The date and time the resource was created.</td>
</tr>
<tr>
<td>Server Name</td>
<td>os_compute::ServerName</td>
<td>The server name.</td>
</tr>
<tr>
<td>Flavor</td>
<td>os_compute::FlavorName</td>
<td>The display name of the flavor.</td>
</tr>
<tr>
<td>Image name</td>
<td>os_compute::ImageName</td>
<td>The display name of the image.</td>
</tr>
<tr>
<td>Status</td>
<td>os_compute::ServerStatus</td>
<td>The current server state.</td>
</tr>
<tr>
<td>Security groups</td>
<td>os_compute::security_groups</td>
<td>The security group names.</td>
</tr>
<tr>
<td>VCPUs</td>
<td>os_compute::vcpus</td>
<td>The number of virtual CPUs in use.</td>
</tr>
<tr>
<td>RAM (MB)</td>
<td>os_compute::ram</td>
<td>The RAM (MB) on the virtual machine.</td>
</tr>
<tr>
<td>Disk (GB)</td>
<td>os_compute::disk</td>
<td>The disk size (GB) on the virtual machine.</td>
</tr>
<tr>
<td>Key name</td>
<td>os_compute::key_name</td>
<td>The keypair name.</td>
</tr>
<tr>
<td>IP addresses</td>
<td>os_compute::ServerAddresses</td>
<td>The IP addresses assigned to the virtual machine.</td>
</tr>
<tr>
<td>Metadata</td>
<td>os_compute::ServerMetadata</td>
<td>Custom server metadata at server launch time.</td>
</tr>
</tbody>
</table>
### Google Cloud Platform

Discover compute engine instances running on your GCP account, including instance attributes, packages, and containers.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createTime</td>
<td>The date and time the resource was created.</td>
</tr>
<tr>
<td>hostname</td>
<td>The name of the host.</td>
</tr>
<tr>
<td>id</td>
<td>The ID of the instance.</td>
</tr>
<tr>
<td>image</td>
<td>The image identification number used by your cloud source.</td>
</tr>
<tr>
<td>name</td>
<td>The resource name.</td>
</tr>
<tr>
<td>os</td>
<td>The operating system running on the host.</td>
</tr>
<tr>
<td>osKernel</td>
<td>The type of kernel running in the operating system: Linux or Windows.</td>
</tr>
<tr>
<td>osVersion</td>
<td>The version of the operating system running on the host.</td>
</tr>
<tr>
<td>preemptible</td>
<td>Indicates whether it is an instance that can be temporarily interrupted: true or false.</td>
</tr>
<tr>
<td>privateIPv4</td>
<td>The private IP version 4 address.</td>
</tr>
<tr>
<td>provider</td>
<td>The name of the cloud provider: Google Cloud Platform.</td>
</tr>
<tr>
<td>publicIPv4</td>
<td>The IP version 4 address.</td>
</tr>
<tr>
<td>puppetInstalled</td>
<td>The indicator on whether Puppet is installed.</td>
</tr>
<tr>
<td>puppetVersion</td>
<td>The version of Puppet installed.</td>
</tr>
<tr>
<td>region</td>
<td>The region where the host exists.</td>
</tr>
<tr>
<td>status</td>
<td>The current status of the host.</td>
</tr>
<tr>
<td>tags</td>
<td>The tags, containing metadata, assigned to the instance.</td>
</tr>
<tr>
<td>uptime</td>
<td>The uptime for the host, in seconds.</td>
</tr>
</tbody>
</table>

### VMware vSphere

Discover ESXi nodes running on your VMware vSphere account, including node attributes, packages, and containers.

<table>
<thead>
<tr>
<th>Field</th>
<th>Facet::attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation</td>
<td>vmware_computeInstance::Config_Annotation</td>
<td>Description for the virtual machine.</td>
</tr>
<tr>
<td>Disk</td>
<td>vmware_computeInstance::Disk</td>
<td>Guest information about disks; used space and total space available.</td>
</tr>
<tr>
<td>Field</td>
<td>Facet::attribute</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Fault tolerance state | vmware_computeInstance::Runtime_FaultToleranceState | Defines a set of states for a fault tolerant virtual machine:  
• disabled  
• enabled  
• needSecondary  
• notConfigured  
• running  
• starting |
| Guest full name       | vmware_computeInstance::GuestFullName    | The full name of the guest operating system for the virtual machine.                                                                     |
| Guest hostname        | vmware_computeInstance::HostName         | The FQDN of the guest host.                                                                                                               |
| Guest heartbeat status| vmware_computeInstance::QuickStats       | The current status of the guest operating system:  
• gray - VMware tools not installed or running.  
• red - No heartbeat.  
• yellow - Intermittent heartbeat.  
• green - Guest operating system responding normally. |
| Guest state           | vmware_computeInstance::GuestState       | Operation mode of the guest operating system:  
• running  
• shuttingdown  
• resetting  
• standby  
• notrunning  
• unknown  
<p>| ID                    | vmware_computeInstance::ID               | The guest operating system identifier.                                                                                              |
| IP address            | host::privateIPv4                        | The private IP version 4 address.                                                                                                       |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Facet::attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP stack</td>
<td>vmware_computeInstance::IpStack</td>
<td>Guest information about the IP networking stack:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DomainName</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• IpAddress</td>
</tr>
<tr>
<td>Last scan</td>
<td></td>
<td>The last scan of the host represented by the data platform timestamp in UTC.</td>
</tr>
<tr>
<td>Max memory usage (MB)</td>
<td>vmware_computeInstance::Runtime_MaxMemoryUsage</td>
<td>The maximum memory usage based on the memory configuration of the virtual machine, as well as limits configured on the virtual machine, or any parent resource pool.</td>
</tr>
<tr>
<td>Memory size (MB)</td>
<td>vmware_computeInstance::Config_MemorySizeMB</td>
<td>The memory size of the virtual machine, in megabytes.</td>
</tr>
<tr>
<td>Name</td>
<td>host::name</td>
<td>The name of the host.</td>
</tr>
<tr>
<td>Net</td>
<td>vmware_computeInstance::Net</td>
<td>Guest information about network adapters:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Network</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MacAddress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• IpAddress</td>
</tr>
<tr>
<td>Number of CPU's</td>
<td>vmware_computeInstance::Config_NumCpu</td>
<td>The number of processors in the virtual machine.</td>
</tr>
<tr>
<td>Number of virtual disks</td>
<td>vmware_computeInstance::Config_NumVirtualDisks</td>
<td>The number of virtual disks attached to the virtual machine.</td>
</tr>
<tr>
<td>Operating system</td>
<td>computeHost::os</td>
<td>The operating system running on the host.</td>
</tr>
<tr>
<td>OS version</td>
<td>computeHost::osVersion</td>
<td>The version of the operating system running on the host.</td>
</tr>
<tr>
<td>Overall status</td>
<td>vmware_computeInstance::OverallStatus</td>
<td>The overall alarm status on this host.</td>
</tr>
<tr>
<td>Power state</td>
<td>vmware_computeInstance::Runtime_PowerState</td>
<td>Defines a set of states for a virtual machine:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• poweredOff</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• poweredOn</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• suspended</td>
</tr>
</tbody>
</table>
### Tools running status
Field: `vmware_computeInstance::ToolsRunningStatus`
- **Description**: The current running status of VMware tools in the guest operating system.

### Tools version
Field: `vmware_computeInstance::ToolsVersion`
- **Description**: The VMware tools version.

### Uptime
Field: `computeHost::uptime`
- **Description**: The uptime for the instance, in seconds.

## Network nodes
Discover nodes running on your network, including node attributes, packages, and containers.

### Attribute | Description
---|---
dnsName | The private DNS name.
hostname | The name of the host.
names | The resource name.
os | The operating system running on the host.
osKernel | The type of kernel running in the operating system: Linux or Windows.
osVersion | The version of the operating system running on the host.
privateIPv4 | The private IP version 4 address.
privateIPv6 | The private IP version 6 address.
publicDnsName | Fully qualified public DNS hostname.
publicIPv4 | The IP version 4 address.
puppetInstalled | The indicator on whether Puppet is installed.
puppetVersion | The version of Puppet installed.
uptime | The uptime for the host, in seconds.

## Package attributes
Discover various types of information about a package instance, its attributes, along with a list of hosts and containers it’s installed on.

Discover attributes that describe the characteristics of the package. The UI queries the data platform to discover the name, version, and packageManager attributes.

### Attribute | Description
---|---
name | The package name.
version | The package version.
packageManager | The name of the package manager:
- apk
- Chocolatey
- dpkg
- rpm

© 2020 Puppet, Inc.
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>msi</td>
<td>Discover various types of information about a container image, its attributes, packages in use, and its label information. Discover attributes that describe the characteristics of the container. The UI queries the data platform for the discovered container facet.</td>
</tr>
</tbody>
</table>

Discover the container label names and values:

- **Label** - the label name, derived from the `name` attribute.
- **Value** - the label value, derived from the `value` attribute.

**Filtering and exporting data**

Create custom filters and customize table views to view resource data most important to you, or for backup purposes, export data to a CSV file.

**Create custom filters**

You can create a list of nodes, packages, or containers for further investigation by creating a custom filter with multiple criteria.

1. On any listing or details page, click **Create filter**.
2. Select a **Field**.
3. Select an **Operator**:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Filter for resources where the value of the field:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>is an exact match to the value you specify.</td>
</tr>
<tr>
<td>Not equal to</td>
<td>is anything except an exact match to the value you specify.</td>
</tr>
<tr>
<td>Contains</td>
<td>contains the value you specify.</td>
</tr>
<tr>
<td>Does not contain</td>
<td>does not contain the value you specify.</td>
</tr>
</tbody>
</table>

4. Enter a **Value**.
5. Click **Add filter**.
6. To add criteria to the filter, click **Add another filter**.
7. As needed, repeat these steps to add additional filters.
8. To display the filtered list, click **Apply all**.

**Customize table views**

Customize each table view by showing or hiding table columns on any of the node, package, or container pages.

1. On any listings or details page, click **Columns +/-**.
   
   By default, all available table columns are selected.

2. To hide a table column, click the column header to remove the selection indicator (✓).

3. To confirm your selections, click **Apply**.

**Export data**

To keep a backup of data relating to nodes, packages, or containers, download a CSV file that contains the same information as the page you are currently viewing, including all filter selections.

To download resource data as a .csv file, click **Export**. The filename is `<ui_page>_<timestamp>.csv`, for example `pd_export_nodes_2019-06-17_1054.csv`.

**Review recent events**

View a list of recent events and drill down to see useful information about each one.

1. On the sidebar, click **Recent events**.

   The **Recent events** page provides a quick glance at the status of each event, organized by the type of event, and sorted by the date and time of when each event succeeded or failed.

2. To view the details of an event, click the event name.

   The **Event details** page provides detailed information about the specific event, organized by the status of the event, and sorted by the date and time of when the event completed.

**Discovery events**

The **Recent events** page provides the following information about each discovery run which, by default, occurs every 4 hours.

<table>
<thead>
<tr>
<th>Event detail</th>
<th>Description</th>
</tr>
</thead>
</table>
| Total jobs   | Total number of jobs relating to the discovery event. Each discovery event has two jobs:  
• Discover the node.  
• Discover resources on the node. |
| Failed       | Total number of failed jobs. |
| Status       | The event status. |
| Started      | Date and time the discovery run started. |
### Event detail

<table>
<thead>
<tr>
<th>Event detail</th>
<th>Description</th>
</tr>
</thead>
</table>
| Status       | The job status:  
  • Succeeded  
  • Failed  
  • Pending  
  • Running  |
| Source       | Your configured source: vulnerability scanner or infrastructure source.  
| Job details  | The number of facets discovered. Each facet represents a set of related attributes, each one independently maintained, that describe a certain aspect of a discovered resource; node, package, or container.  
| Completion time | The time the discovery run completed. |

### Tasks events

The Task event page provides the following information about the task that was ran on nodes.

<table>
<thead>
<tr>
<th>Event detail</th>
<th>Description</th>
</tr>
</thead>
</table>
| Total jobs   | Total number of jobs relating to the task event. Each task event has two jobs:  
  • Run task on node.  
  • Discover resources on the node.  
| Failed       | Total number of failed jobs.  
| Status       | The event status.  
| Started      | Date and time the event started.  
| Status       | The job status:  
  • Succeeded  
  • Failed  
  • Pending  
  • Running  |
| Source       | Your configured source: vulnerability scanner or infrastructure source. |
### Event detail

| Job details | The command output from running the task and the number of facets discovered. Each facet represents a set of related attributes, each one independently maintained, that describe a certain aspect of a discovered resource; node, package, or container. |
| Completion time | The time the task completed. |

### Integration status

The **Latest scan results** page displays information on the latest discovery scan carried out by your scanner.

Click the **Integration status** link in the navigation sidebar to access the **Latest scan results**. The **Integration status** link is accompanied by an icon indicating if the latest scan is in progress, completed or returned an error.

The **Latest scan results** page provides the following information:

<table>
<thead>
<tr>
<th>Information</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vulnerability scanner status</td>
<td>Either in <strong>progress</strong>, <strong>completed</strong>, or returned an <strong>error</strong>.</td>
</tr>
<tr>
<td>Vulnerability scanner</td>
<td>The name of the scanner used for the scan.</td>
</tr>
<tr>
<td>Latest communication time</td>
<td>The time of the latest status response on the scan.</td>
</tr>
<tr>
<td>Completion time of previous update</td>
<td>The time when the last scan was completed.</td>
</tr>
<tr>
<td>Vulnerability scanner error</td>
<td>Error message from the vulnerability scanner, if any.</td>
</tr>
</tbody>
</table>

### Troubleshooting

Use this section to troubleshoot issues with your Puppet Remediate installation.

**Forgotten password**

We do not currently support resetting individual passwords. If you forget the admin password, you must reset the entire system and re-enter any provider and credential information. To reset, run the following command:

```
docker-compose run remediate reset
```

**How to generate and send logs to support**

To generate service logs, create a tarball by running the mayday command:

```
docker-compose run remediate mayday
```

The tarball contains debug information which the Puppet support team uses for troubleshooting. You can send this to support via a Zendesk ticket or email.
For general help

For a list of helpful CLI commands, run the following:

```
docker-compose run remediate --help
```