Installing Puppet Server:

Supported operating systems:

Puppet provides official packages that install Puppet Server and all of its prerequisites for the following platforms:

Operating Systems	**Version**
Red Hat Enterprise Linux	7, 8, 9
Debian	9 (Stretch), 10 (Buster), 11 (Bullseye), 12 (Bookworm) amd64
Ubuntu	16.04 (Xenial, amd64 only), 18.04 (Bionic), 20.04 (Focal), 22.04
SLES	12 SP1, 15 (x86_64)

Puppet Server listens on port 8140 by default.

If needed change the port Puppet Server uses in
`/etc/puppetlabs/puppetserver/conf.d/puppetserver.conf.`

Enable the Puppet platform repository : Link to get repository

Note : - change hostname of server ex. Puppetmaster.local, something.local

1. **Logged in as root, download the package and run the dpkg tool in install mode:**

```
wget https://apt.puppet.com/puppet8-release-focal.deb
sudo dpkg -i puppet8-release-focal.deb
sudo apt-get update
```

2. **Install the Puppet Server package and start puppetserver:**

```
apt-get install puppetserver
sudo systemctl start puppetserver
```

3. **Configure Puppet Server (optional):**

- Edit the configuration file:

```
sudo nano /etc/puppetlabs/puppetserver/conf.d/puppetserver.conf
```

- Modify the JAVA_ARGS to allocate more memory, if needed:

```
JAVA_ARGS="-Xms2g -Xmx2g"
```

- Restart puppetserver:

```
sudo systemctl restart puppetserver
sudo systemctl enable puppetserver
```

4. **Install the agent:**

   - Install and start service:
     ```
     sudo apt-get install puppet-agent
     sudo /opt/puppetlabs/bin/puppet resource service puppet
     ensure=running enable=true
     ```

   - Configure server setting:
     ```
     puppet config set server something.local --section main
     systemctl restart puppet
     ```

   - Check puppetserver service and certificate:
     ```
     systemctl status puppet
     puppetserver ca list --all
     ```

   Note : add host entry in /etc/hosts [ip something.local] or DNS Entry

5. **To set up automatic certificate signing in Puppet :**

   The autosign.conf file contains a list of hostnames (or regular expressions) for which Puppet will automatically sign certificates.

- Create autosign.conf file:
  ```
  sudo nano /etc/puppetlabs/puppet/autosign.conf
  ```
- Add hostnames, domain names, or regular expressions:
  ex.
  ```
  *.example.com -> ram.example.com, sham.example.com
  agent*.example.com -> agent1.example.com,agent2.exam..
  * -> for all vms
  ```
- Modify Puppet Master Configuration:
  ```
  sudo nano /etc/puppetlabs/puppet/puppet.conf
  ```
- Add following setting under [main]:
  ```
  [main]
  autosign = /etc/puppetlabs/puppet/autosign.conf
  ```
- Restart the Puppet Server.

Integrate Puppet Server with Foreman:

Supported Platforms:

- **Red Hat Enterprise Linux 9**
  - Architectures: x86_64 only
  - Apply all SELinux-related errata.
- **CentOS Stream 9**
  - Architectures: x86_64 only
  - **Note:**
    - The RPM packages are built on Red Hat Enterprise Linux 9, but tested to work also on CentOS Stream 9
    - EPEL 9 is incompatible, enabling will cause package dependency issue.
- **Ubuntu 22.04 (Jammy)**
- **Debian 11 (Bullseye)**
- **Debian 12 (Bookworm)**

Note: below mentioned configurations are for Ubuntu 22.04 (Jammy) for other refer doc Foreman Manual

1. **Enable Puppet's repository:**

- `sudo apt-get -y install ca-certificates`
- `cd /tmp && wget https://apt.puppet.com/puppet8-release-jammy.deb`
- `sudo apt-get install /tmp/puppet8-release-jammy.deb`

2. **Enable the Foreman repositories:**

- `sudo wget https://deb.theforeman.org/foreman.asc -O /etc/apt/trusted.gpg.d/foreman.asc`
- `echo "deb http://deb.theforeman.org/ jammy 3.13" | sudo tee /etc/apt/sources.list.d/foreman.list`
- `echo "deb http://deb.theforeman.org/ plugins 3.13" | sudo tee -a /etc/apt/sources.list.d/foreman.list`

3. **Downloading the installer:**

- `sudo apt-get update && sudo apt-get -y install foreman-installer`

4. **Running the installer:**

- `sudo foreman-installer`

After it completes, the installer will print Initial credentials and some details about where to find Foreman and the Smart Proxy.

Installing Puppet Agent:

Supported operating systems:

puppet-agent packages are available for the platforms listed in the table:

Operating system	Tested versions	Untested versions
Debian	10, 11, 11 (ARM), 12 (x86_64, ARM)	
Fedora	36 (x86_64), 40 (x86_64)	
macOS	11 Big Sur (64-bit packages only), 12 Monterey (64-bit packages only), 12 (M1), 13 Ventura (x86_64, ARM), 14 (ARM), 14 (Intel)	
Microsoft Windows	10 Enterprise, 11 Enterprise (x86_64)	8, 10
Microsoft Windows Server	2012R2, 2016, 2019, 2022	2012
Red Hat Enterprise Linux, including: • Amazon Linux v1 (using RHEL 6 packages) • Amazon Linux v2 x86_64 (using RHEL 7 packages)	6 (i386, x86_64), 7 (x86_64), 8 (x86_64, aarch64, ppc64le), 9 (x86_64, ARM64, ppc64le)	
Amazon Linux	2 (AARCH64), 2023 (x86_64, AARCH64)	
SUSE Linux Enterprise Server	12 (x86_64), 15 (x86_64)	
AlmaLinux	8 (x86_64), 9 (x86_64, AARCH64)	
Rocky Linux	8 (x86_64), 9 (x86_64, AARCH64)	
Oracle Linux	6 (x86_64, i386), 7 (x86_64), 8 (x86_64, aarch64, ppc64le)	
Scientific Linux	6 (x86_64, i386), 7 (x86_64)	
Ubuntu	18.04, 18.04 AARCH, 20.04, 20.04 AARCH, 22.04 (x86_64, ARM64), 24.04 (x86_64, ARM)	

For Linux VMs:

1. **Logged in as root, download the package and run the dpkg tool in install mode:**

```
wget https://apt.puppet.com/puppet8-release-focal.deb
sudo dpkg -i puppet8-release-focal.deb
sudo apt-get update
```

2. **Install and start service:**

```
   sudo apt-get install puppet-agent
   sudo /opt/puppetlabs/bin/puppet resource service puppet
   ensure=running enable=true
```

3. **Configure setting:**

```
    puppet config set server something.local --section main
   Puppet config set runinterval 30m --section main
   systemctl restart puppet
```

4 . **Add Host Entry (**optional if DNS Entry for server is made**):**

```
    ip something.local
```

For Windows Vms:

1. **Install Chocolatey:** Installation Document for Chocolatey
2. **Install puppet agent:** (use PowerShell as Administrator)

```
choco install puppet-agent -y
```

3. **Configure setting:** go to file C:\ProgramData\PuppetLabs\puppet\etc\puppet.conf and add

```
 [main]
server = something.local
runinterval = 30m
manage_internal_file_permissions=true
```

4. **Restart puppet service:**

```
  Restart-Service -Name puppet
```

5. **Add Host Entry (**optional if DNS Entry for server is made**):**

```
    ip something.local
```

NOTE: You can clone this VMs and change its hostname and restart and your VM will auto register if auto-assign is enabled on puppet server.

Types of Resources:

In Puppet, resources are used to manage system configuration, and different types of resources are available to manage various aspects of a system.

1. **anchor**
   - Used to create a "marker" in the catalog. It can be used to ensure that certain resources are applied in a specific order.
2. **augeas**
   - Used to manage configuration files with a special syntax, called **Augeas**, that provides a structured way of interacting with configuration files. It allows you to edit configuration files based on their internal structure.
3. **cron**
   - Manages cron jobs. You can use this resource to ensure that specific cron jobs are present, modified, or removed on a system.
4. **exec**
   - Executes arbitrary commands or scripts on the system. The exec resource is flexible, allowing you to run shell commands and scripts based on specific conditions (like onlyif or unless).
5. **file**
   - Manages files and directories. This is a very important resource in Puppet that allows you to ensure that specific files exist, have the correct permissions, ownership, and content.
6. **file_line**
   - Manages individual lines in a file. This is useful for adding, removing, or modifying a single line of text in a file, which is commonly used for configuration changes.
7. **filebucket**
   - A resource type used for managing file backups (or file buckets). This resource is typically used to store versions of files for backup and restore purposes.
8. **group**
   - Manages groups on the system. This resource is used to ensure that specific groups exist, with appropriate names, IDs, and memberships.
9. **host**
   - Manages entries in the system's /etc/hosts file. This resource ensures that specific hostnames and IP addresses are present in the hosts file.
10. **mount**
    - Manages mounts (e.g., network shares, or local filesystems). This resource ensures that a file system is mounted or unmounted at a specified location.
11. **notify**
    - Used to send notifications to the user or log files. The notify resource allows you to trigger messages or logging when certain resources are applied.

12. package

- Manages software packages. This resource is used to install, upgrade, or remove software packages using the system's package manager (like apt, yum, or dpkg).

13. resources

- Not a specific resource type but a method to manage or interact with other resources dynamically. For example, you can use resources to collect and manage Puppet resources.

14. schedule

- Manages schedules for recurring tasks or resources. This can be used to define when resources should run on the system.

15. scheduled_task

- Specifically used to manage scheduled tasks on Windows systems. This is equivalent to cron jobs on Linux but for Windows.

16. selboolean

- Manages SELinux booleans. SELinux is a security module for Linux, and this resource is used to configure and manage SELinux boolean values to control security policies.

17. selmodule

- Manages SELinux policy modules. This is used to install, configure, and manage SELinux modules to enforce custom security policies.

18. service

- Manages services on the system (like starting, stopping, or restarting services). This resource is used to ensure that specific services are running or stopped.

19. ssh_authorized_key

- Manages SSH authorized keys. This resource ensures that a specific public SSH key is present in the authorized_keys file for a user, enabling SSH access.

20. sshkey

- Manages SSH keys (for creating, installing, and managing SSH keys). This resource is used to generate and distribute SSH keys on systems.

21. stage

- Stages are used to organize and control the order of resource application in Puppet's catalog. A stage helps to group related resources for better control over execution.

22. tidy

- Manages the cleanup of files and directories. This resource is used to remove old or unused files and manage the cleanup of directories to maintain system cleanliness.

23. user

- Manages user accounts on the system. You can use this resource to create, modify, or remove users and set their attributes like home directory, shell, and password.

24. yumrepo

- Manages **YUM** repositories on Red Hat-based systems. This resource ensures that a YUM repository is present and configured correctly for package management.

25. zfs

- Manages **ZFS** file systems. ZFS is a file system and logical volume manager, primarily used in Solaris and OpenZFS environments. This resource is used to manage ZFS pools, file systems, and volumes.

26. zone

- Manages **Solaris Zones**, which are a virtualization feature in the Solaris operating system. This resource helps in managing and configuring zones in a Solaris system.

27. zpool

- Manages **ZFS storage pools** (zpool) on ZFS-enabled systems. A zpool is a storage pool of physical devices used by ZFS. This resource ensures that ZFS pools are created, modified, or removed.

For a detailed explanation of Puppet resources, their uses click here

Syntax:

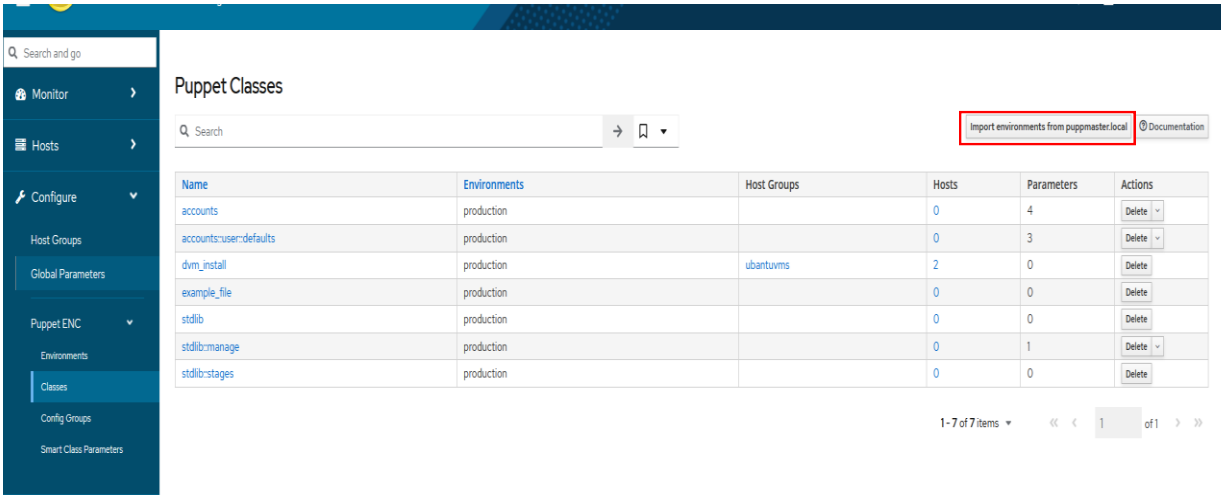
```
<TYPE> { '<TITLE>':
          <ATTRIBUTE> => <VALUE>,
 }
```

Create class containing all resources:

example:

```
class dvm_install {

# Ensure the target directory exists before extracting the file

  file { "/mnt/network_share":

    ensure => "directory",

    recurse=> "true",

    mode   => "0755",

    before => Exec["update-all-packages"],  # Ensure the directory is
created before packages are installed

  }

  file { '/home/puppet':

    ensure => 'directory',

    mode   => '0755',
    before => Exec["update-all-packages"],  # Ensure the directory is created before packages are installed
  }
  file { "/home/puppet/Downloads":
    ensure => "directory",
    recurse=> "true",
    mode   => "0755",
    before => Exec["update-all-packages"],  # Ensure the directory is created before packages are installed
  }
  # Update all packages to the latest
  exec { "update-all-packages":
    command => "apt update && apt upgrade -y",  # Using apt for Ubuntu
    before  => Package["p7zip-full", "p7zip-rar"],  # Ensure update happens before these packages are installed
    path    => ['/usr/bin', '/usr/sbin',],
  }
  # Install p7zip and p7zip-plugins
  package { ["p7zip-full", "p7zip-rar"]:  # Use correct package names for Ubuntu
    ensure => "installed",
    before => Package["cifs-utils"],
  }
  # Install cifs-utils to mount the network share
  package { "cifs-utils":
    ensure => "latest",
    before => Mount["/mnt/network_share"],
  }
  mount { "/mnt/network_share":
    ensure  => "mounted",
    device  => "//172.25.0.11/Releases/HyWorks/HyDesk/Linux_VDI/Patches/Ubuntu_Assistive_Packages",  # Corrected path
    fstype  => "cifs",
    options => "username=nas,password=nas",
    before  => Exec["copy_file"],
 }
 # Use exec to copy the file from the mounted network share to the local
system
  exec { "copy_file":
    command => "cp
/mnt/network_share/Linux_Assistive_Package_3.5.0.75_Ubuntu.7z
/home/puppet/Downloads/Linux_Assistive_Package_3.5.0.75_Ubuntu.7z",
    path    => ['/usr/bin', '/usr/sbin', '/bin'],
    unless  => "test -f
/home/puppet/Downloads/Linux_Assistive_Package_3.5.0.75_Ubuntu.7z",  # Skip
if file already exists
    before  => Exec["extract-7z-file"],  # Ensure the file is copied before
extraction
  }
  # Extract the 7z file
  exec { "extract-7z-file":
    command => "7z x
/home/puppet/Downloads/Linux_Assistive_Package_3.5.0.75_Ubuntu.7z -
o/home/puppet/Downloads/",
    creates =>
"/home/puppet/Downloads/Linux_Assistive_Package_3.5.0.75_Ubuntu",
    path    => ['/usr/bin', '/usr/sbin',],
    before  => Exec["run-installer"],
  }
  # Run the installer in quiet mode
  exec { "run-installer":
    command =>
"/home/puppet/Downloads/Linux_Assistive_Package_3.5.0.75_Ubuntu/execute_ins
taller.sh 1 AD SHD 1",
    creates =>
"/home/puppet/Downloads/Linux_Assistive_Package_3.5.0.75_Ubuntu/installed",
    path    => ['/usr/bin', '/usr/sbin',],
  }
}
```

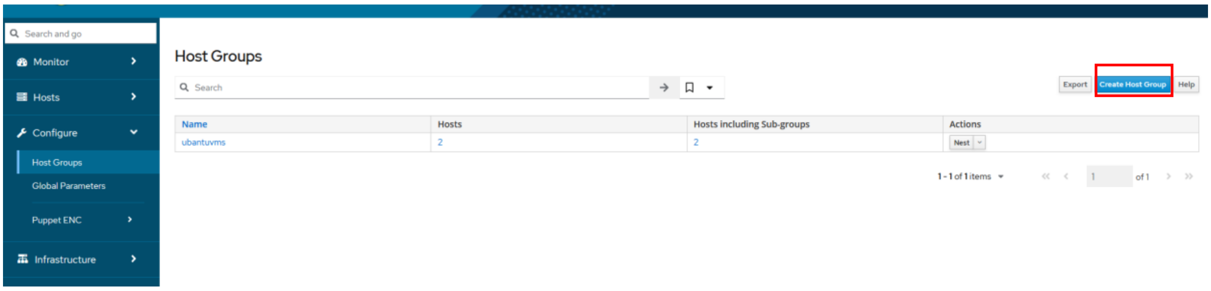
- save this class in file with name dvm_Install.pp at location /etc/puppetlabs/code/environments/production/manifests/
- Now import this class in foreman



Puppet Classes

Name	Environments	Host Groups	Hosts	Parameters	Actions
accounts	production		0	4	Delete
accounts::user::defaults	production		0	3	Delete
dvm_install	production	ubantuvms	2	0	Delete
example_file	production		0	0	Delete
stdlib	production		0	0	Delete
stdlib::manage	production		0	1	Delete
stdlib::stages	production		0	0	Delete

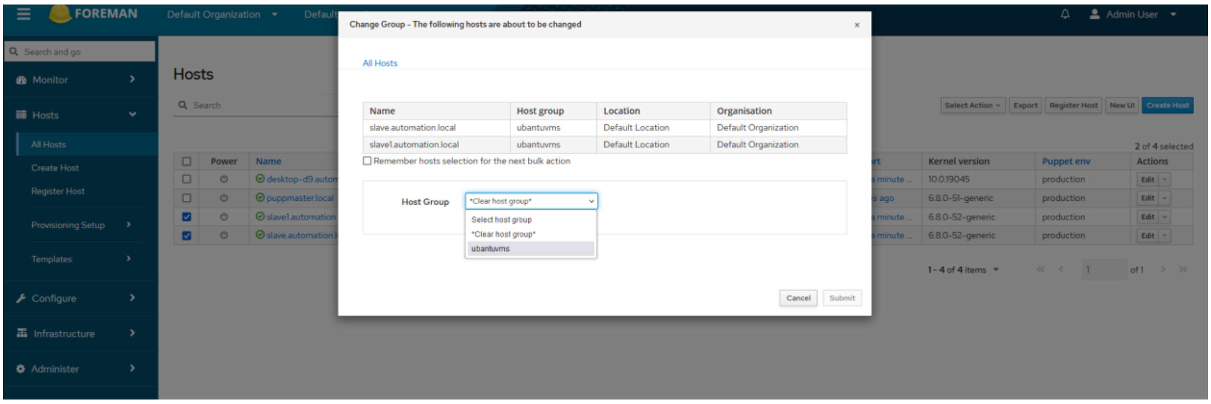
- Create hosts group



Host Groups

Name	Hosts	Hosts including Sub-groups	Actions
ubantuvms	2	2	Nest

- Select all hosts and add in to Group



Change Group - The following hosts are about to be changed

All Hosts

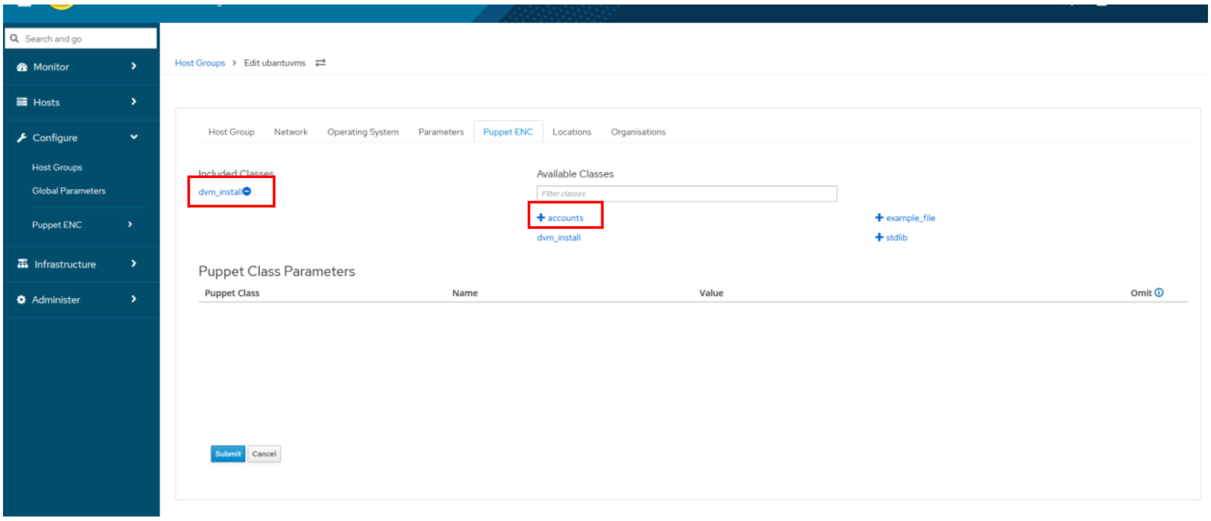
Name	Host group	Location	Organisation
slave.automation.local	ubantuvms	Default Location	Default Organization
slave1.automation.local	ubantuvms	Default Location	Default Organization

□ Remember hosts selection for the next bulk action

Host Group: *Clear host group*
Select host group
Clear host group
ubantuvms

Cancel Submit

- Now edit host group and select class you want to execute



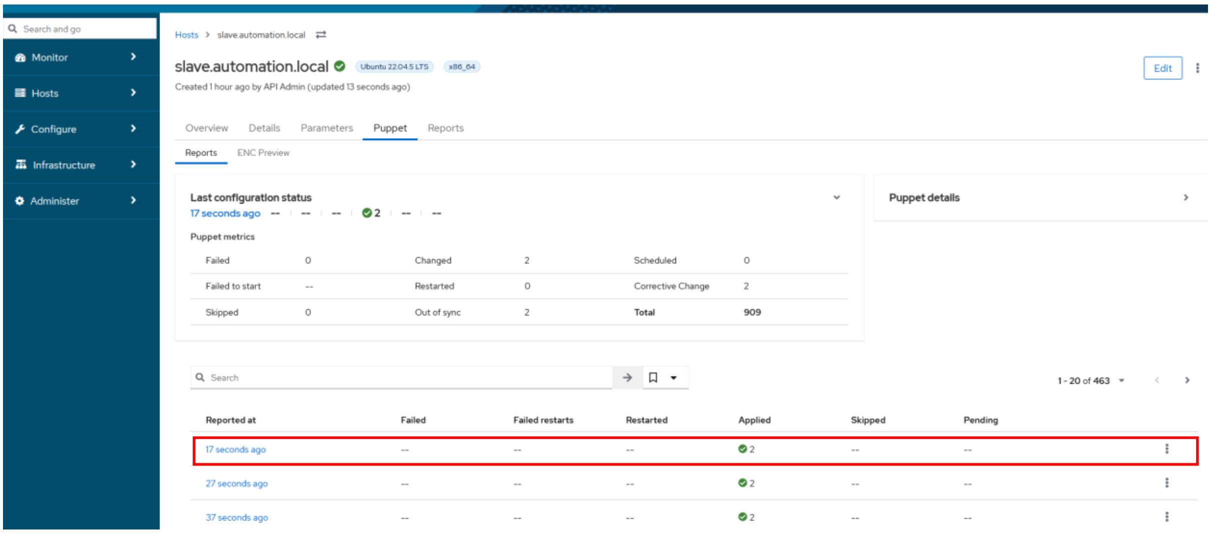
Included Classes
dvm_install

Available Classes
accounts
dvm_install
example_file
stdlib

Puppet Class Parameters

Puppet Class	Name	Value	Omit

- Now Watch execution report for host



slave.automation.local Ubuntu 22.04.5 LTS x86_64

Created 1 hour ago by API Admin (updated 13 seconds ago)

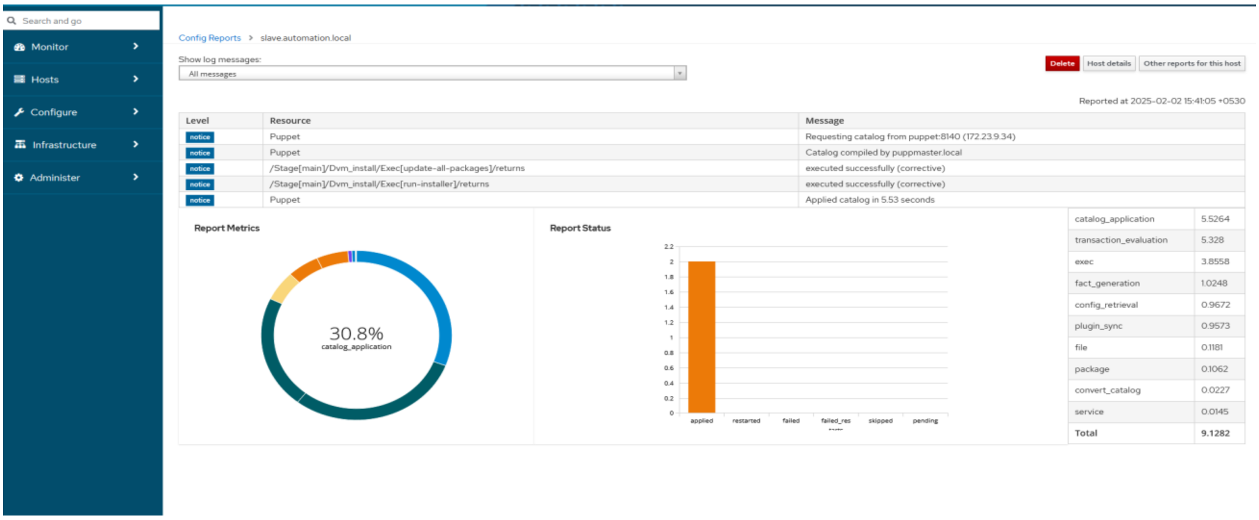
Last configuration status

17 seconds ago -- -- -- ✓ 2 -- --

Puppet metrics

Failed	0	Changed	2	Scheduled	0
Failed to start	--	Restarted	0	Corrective Change	2
Skipped	0	Out of sync	2	**Total**	**909**

Reported at	Failed	Failed restarts	Restarted	Applied	Skipped	Pending
17 seconds ago	--	--	--	✓ 2	--	--
27 seconds ago	--	--	--	✓ 2	--	--
37 seconds ago	--	--	--	✓ 2	--	--



Show log messages: All messages

Reported at 2025-02-02 15:41:05 +0530

Level	Resource	Message
notice	Puppet	Requesting catalog from puppet:8140 (172.23.9.34)
notice	Puppet	Catalog compiled by puppmaster.local
notice	/Stage[main]/Dvm_install/Exec[update-all-packages]/returns	executed successfully (corrective)
notice	/Stage[main]/Dvm_install/Exec[run-installer]/returns	executed successfully (corrective)
notice	Puppet	Applied catalog in 5.53 seconds

catalog_application	5.5264
transaction_evaluation	5.328
exec	3.8558
fact_generation	1.0248
config_retrieval	0.9672
plugin_sync	0.9573
file	0.1181
package	0.1062
convert_catalog	0.0227
service	0.0145
Total	9.1282