# Foreman Node Sharing Plugin

This plugin is deprecated and will no longer be maintained. It was replaced by purely Jenkins solution: Node Sharing Plugin

Allow multiple masters to share a node that is defined as a resource on a Foreman instance.

| Plugin Information |
|---|
| No information for the plugin 'foreman-node-sharing' is available. It may have been removed from distribution. |

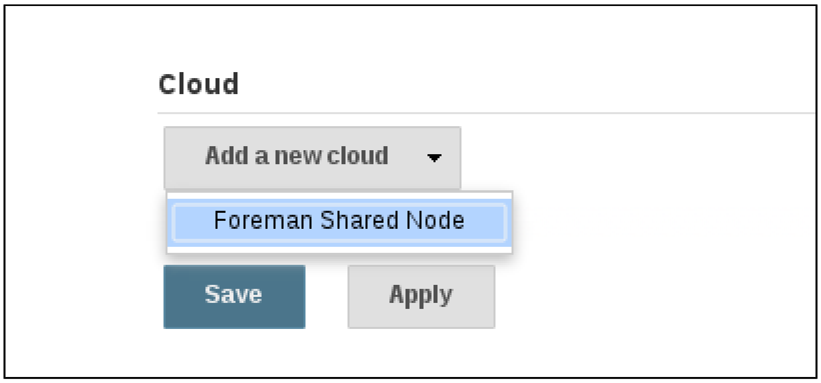
## Essentials

- This plugin provides a means to automatically coordinate the sharing of bare metal resources as nodes among multiple masters.
- It connects to single **Foreman** instance and multiple **Jenkins masters** reserve it communicating via the Foreman Reserve Plugin API.
    - Ensure the Foreman Reserve Plugin is installed. See https://github.com/david-caro/foreman_reserve
- Once configured, Jenkins is able to reserve hosts based on labels attached to them and the labels builds require.
    - It will behave just any other Jenkins Cloud.
- Nodes are connected over SSH using the same credential.
- To ensure fairness, hosts will be utilized for a single build only and returned back to pool for other masters to reserve.

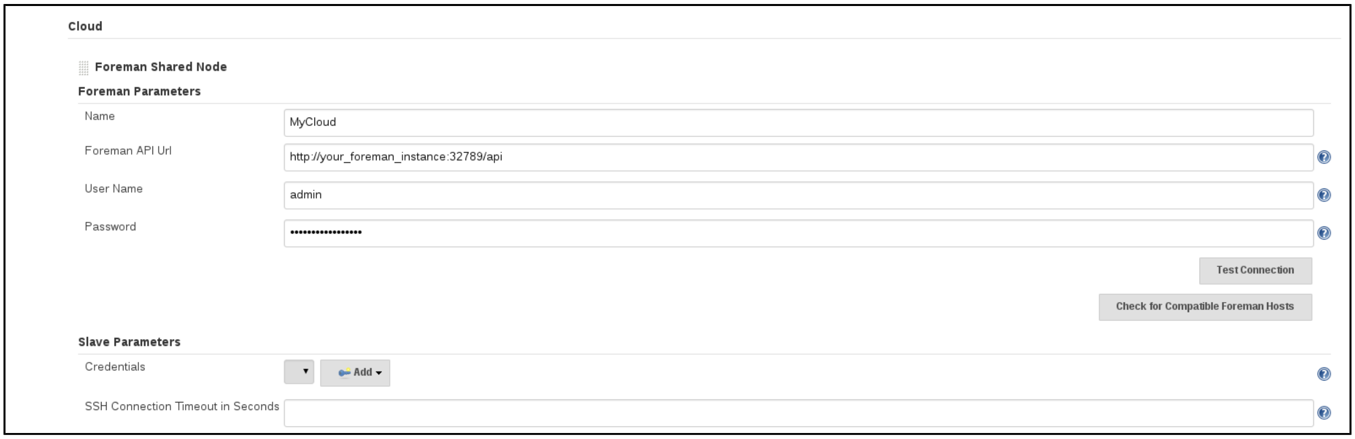
## Set Up

### Cloud in Jenkins(es)

- Once the plugin is installed, a cloud type called **Foreman Shared Node** is made available.
- Navigate to **Manage Jenkins** / **Configure Jenkins** / **Cloud**
- Click on **Add a new cloud**
- Select **Foreman Shared Node**



- Enter the Foreman connection details



Cloud

Foreman Shared Node

Foreman Parameters

Name: MyCloud

Foreman API Url: http://your_foreman_instance:32789/api

User Name: admin

Password: ••••••••••••••••

Slave Parameters

Credentials

SSH Connection Timeout in Seconds

- Click **Test Connection** to verify the Foreman connection details.
    - The plugin will report the version of Foreman.

- Click Save.

### Hosts in Foreman

#### Automatically

TODO: Document new inventory population tool.

There is a script that can be used to populate many resources in Foreman. See https://raw.githubusercontent.com/scoheb/acceptance-test-harness/f1f426b171538373d92e69afa278a0288bf9b017/src/test/resources/foreman_node_sharing_plugin/setup.sh

Usage: setup.sh URL HOSTNAME IP LABEL

#### Manually

- Login to the Foreman instance
- Navigate to **Hosts** / **New Host**
- Populate the following properties
  - Name
  - Domain
  - IP Address
- Add the following properties:
  - JENKINS_LABEL
  - JENKINS_SLAVE_REMOTEFS_ROOT
  - RESERVED
- Set the **JENKINS_LABEL** property to be the set of labels that this resources provides. Separate mulitple labels by a space.
- Set the **JENKINS_SLAVE_REMOTEFS_ROOT** property to be the value of the REMOTEFS_ROOT for this shared node.

## Internal workings

- Plugin uses
  - SSH Slaves plugin to connect to machines.
  - Cloud Statistics Plugin to report foreman reservation progress.
  - Resource Disposer Plugin to tear down resources reliably.

### Build sequence

- A job with label **myLabel** is triggered and placed in Jenkins queue.
- Jenkins Cloud API checks to see if a slave exists with label **myLabel**
- If not, all defined Clouds are checked to see which one can provision label **myLabel**
- The plugin is requested to provision and performs:
  - Foreman query to search for an **UN-RESERVED host** where JENKINS_LABEL matches **myLabel**
  - Foreman API call to reserve host
    - The Jenkins URL will be added as the **RESERVED** property value.
  - Foreman query to obtain Jenkins **RemoteFS Root** for the host
- Node is created and SSH Connection established using defined Cloud credentials.
- Build is performed on newly provisioned slave.
- The plugin immediately attempts to terminate the node and calls Foreman API to release the resource.

### Important Notes

- Freestyle, Maven and Matrix jobs are supported.
- In the event that the Foreman connection is lost while a build is running, the plugin will attempt to release the resource in the background
- Should you need to forcefully release a resource, simply set the **RESERVED** property in Foreman to be **false**
  - Only string value '**false**' put into **RESERVED** property makes a node free and available for further reservation, nothing else
- You may also use the **Audit** feature in Foreman for debugging purposes.

## Change Log

#### Version 1.2.9

- Removed PendingDelete behavior
- Bug fixed release

#### Version 1.2.2

- JENKINS-41693 Added timeouts to web client connection
- Javadoc fixes.
- Handle initialization properly when no hosts are reserved.

#### Version 1.2.1

- JENKINS-39148 Implemented basic cleanup on startup
- Show compatible hosts on dedicated page instead of configure page
- JENKINS-40030 Re-defined delays defined in hudson.slaves.NodeProvisioner.NodeProvisionerInvoker
- JENKINS-38899 ForemanSharedNodeCloud.provision(Label) iterates over the whole collection
- JENKINS-39140 Node termination is performed by periodic task
- Override AbstractCloudComputer.doDoDelete()

### Version 1.2.0

- Initial release