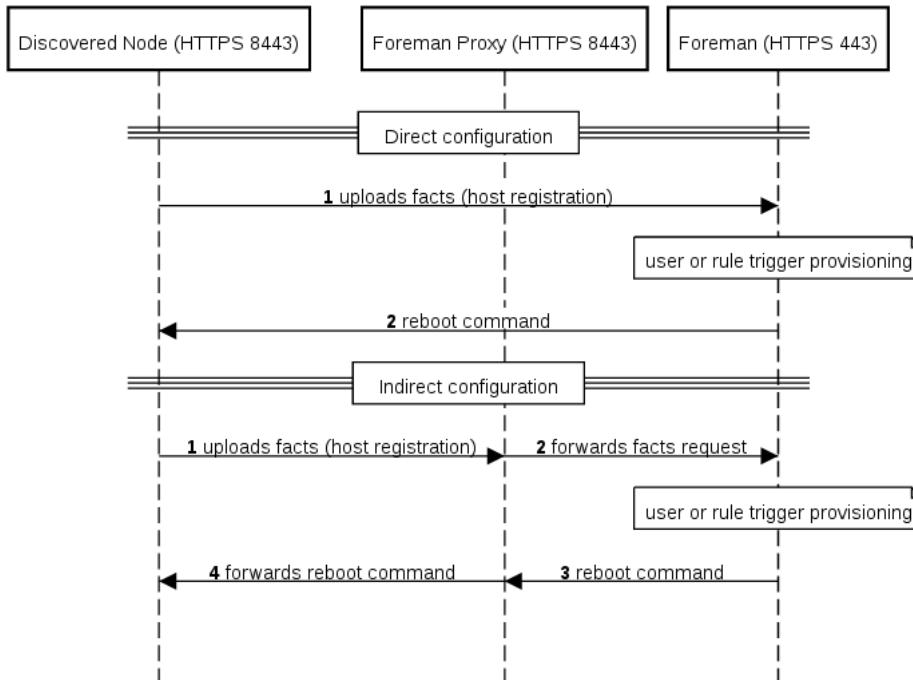


1. Foreman Discovery 4.1 Manual

This plugin enables Foreman to do automatic bare-metal discovery of unknown nodes on the provisioning network. New nodes self-register into Foreman and upload facts collected by Factor (serial id, network interfaces, memory, disks). The registered nodes show up on Discovered Hosts page and provisioning can be initiated either manually (via UI/CLI or API) or automatically via predefined Discovery Rules.

All communication can be optionally passed through Smart Proxy which has direct access both to the provisioning network and Foreman instance:

Foreman Discovery communication paths



1.1 Release Notes

For each Foreman version there is particular version of Foreman Discovery plugin:

Foreman version	Plugin version	Proxy version	Image version	CLI version
<= 1.2	1.0.2	N/A	N/A	N/A
= 1.3	1.1.0	N/A	0.1.1	N/A
= 1.4	1.2.0	N/A	0.3.0	N/A
= 1.5	1.3.0	N/A	0.5.0	N/A
= 1.6	1.4.0	N/A	0.6 or 2.0	N/A
= 1.7	2.0.0	0.1	2.0	0.0.1
= 1.8	3.0.0	1.0.2	2.1	0.0.1
= 1.9	4.0.0	1.0.2	2.1	0.0.2
= 1.10	4.1.1	1.0.3	3.0	0.0.2

1.1.1 Foreman Discovery plugin

4.1: Release notes

New features:

- PXE-less discovery via kexec mechanism

This version only works with Discovery Image 3.0+ but it is possible to reboot legacy discovered nodes into new version.

- #11915 (<http://projects.theforeman.org/issues/11915>) - MO files no longer missing for prod translations
- #12003 (<http://projects.theforeman.org/issues/12003>) - new API for refresh_facts action
- #12053 (<http://projects.theforeman.org/issues/12053>) - improved RHEL6 kexec template
- #11942 (<http://projects.theforeman.org/issues/11942>) - host conversion fixed on PostgreSQL
- #11978 (<http://projects.theforeman.org/issues/11978>) - fixed kexec auto provisioning
- #10300 (<http://projects.theforeman.org/issues/10300>) - adding new image power API support
- #11708 (<http://projects.theforeman.org/issues/11708>) - removed auto_provisioning RABL, improved unit tests
- #11313 (<http://projects.theforeman.org/issues/11313>) - Add the app/views path to RABL view path of foreman

4.0: Release notes

New features:

- Discovered host status indicators help to recognize new or inactive nodes.
- Multiple facts visible on the index page (discovery_fact_column).
- Improved logging helps to identify issues mostly in discovery phase.
- New Reboot all button useful for image upgrade of all nodes.
- Compatibility with Foreman 1.9.

We've fixed a lot of bugs, including issues with facter compatibility, rebooting when editing form, leftover tokens/attribute_sets and permissions. After upgrade, please review permissions of Discovery Manager and Reader roles as we have added more permissions and renamed one. No manual action is needed, but check if everything is setup properly.

- #10469 (<http://projects.theforeman.org/issues/10469>) - Discovery rule taxonomy is enforced
- #10192 (<http://projects.theforeman.org/issues/10192>) - Added model field to discovered
- #11209 (<http://projects.theforeman.org/issues/11209>) - Widget improvements and add status indicator
- #10898 (<http://projects.theforeman.org/issues/10898>) - Auto provisioning and rules permissions corrected
- #10195 (<http://projects.theforeman.org/issues/10195>) - Discovered node is no longer rebooted on error
- #10892 (<http://projects.theforeman.org/issues/10892>) - Multiple columns support for discovery_fact_column
- #9963 (<http://projects.theforeman.org/issues/9963>) - Removed unused permissions and role
- #11101 (<http://projects.theforeman.org/issues/11101>) - Relaxed deface dependency
- #10864 (<http://projects.theforeman.org/issues/10864>) - Removed N+1 queries
- #10828 (<http://projects.theforeman.org/issues/10828>) - Facts are logged when discovery fails
- #9607 (<http://projects.theforeman.org/issues/9607>) - Discovery search succeeds on IP, MAC and Subnet
- #10445 (<http://projects.theforeman.org/issues/10445>) - Subnet is associated through primary iface
- #9857 (<http://projects.theforeman.org/issues/9857>) - Facter extensions now compatible with 1.8+
- #9619 (<http://projects.theforeman.org/issues/9619>) - Improved API documentation
- #9291 (<http://projects.theforeman.org/issues/9291>) - Added Reboot All hosts button
- #10554 (<http://projects.theforeman.org/issues/10554>) - Removed the non UI routes for creating a discovered host
- #10555 (<http://projects.theforeman.org/issues/10555>) - Removed JSON response from discovered_hosts controller
- #10781 (<http://projects.theforeman.org/issues/10781>) - Fixed unit tests for new partition tables
- #9870 (<http://projects.theforeman.org/issues/9870>) - Friendlier error for auto-provisioning with no discovery rule
- #9597 (<http://provision.theforeman.org/issues/9597>) - Added test to check that token is not created for discovered host
- #9870 (<http://provision.theforeman.org/issues/9870>) - Added error response code when auto-provision fails
- #10375 (<http://projects.theforeman.org/issues/10375>) - Entering edit form doesn't delete attributes
- #9895 (<http://projects.theforeman.org/issues/9895>) - Fixed resource type in discovery_rules permissions
- #10365 (<http://projects.theforeman.org/issues/10365>) - Removed all tokens when there are no managed hosts
- #10005 (<http://projects.theforeman.org/issues/10005>) - Removed unused default scope and added tax tests
- #9881 (<http://projects.theforeman.org/issues/9881>) - Rules work when organizations and locations are disabled
- #9912 (<http://projects.theforeman.org/issues/9912>) - Fixed ambiguous column error by using table_name
- #9870 (<http://projects.theforeman.org/issues/9870>) - Auto-provisioning a host, when no rule is defined, raises proper error
- #9881 (<http://projects.theforeman.org/issues/9881>) - Added organization/location to discovery_rules
- #9546 (<http://projects.theforeman.org/issues/9546>) - Moved host_converter to proper directory for rails
- #9572 (<http://projects.theforeman.org/issues/9572>) - Improved API for CLI
- #9044 (<http://projects.theforeman.org/issues/9044>) - Refactored MAC test fix
- #9366 (<http://projects.theforeman.org/issues/9366>) - Host limit for discovery rule defaults to zero
- #9488 (<http://projects.theforeman.org/issues/9488>) - Migration to delete discovery_attribute_set from managed hosts

3.0: New features:

- bugfixes
- Foreman 1.8 compatibility

2.0: New features:

- automatic provisioning
- ZIP-based extensions
- indirect communication via foreman proxy plugin
- support for new discovery image 2.0
- UI enhancements
- new CLI hammer plugin
- reboot option

1.1.2 Smart Proxy Discovery plugin

1.0: Initial version.

1.1.3 Foreman Discovery Image

3.0.0: The new image now does not initialize all network cards with DHCP by default (use `fdi.initnet=all` to force this behavior). We have cleaned up network initialization little bit further, which solved two race conditions with `systemd`. Also, new option `cachefacts` was introduced to solve issues with hardware with late initialization. This can be overridden with kernel command line options. Also, `systemctl` is following system journal on the second console and smart proxy now logs into `journald` instead of separate file. Servers with IPMI on port #2 (Dell PowerEdge) is now reported correctly.

The biggest new feature is text-based interface with enhanced status information, ability to turn on/off ssh and better reporting/debugging. The ISO image can now be burned onto CDROM/USB stick and nodes can be discovered semi-automatically without PXE or DHCP. Possible scenarios:

- fully-automated discovery (traditional approach)
- networks with DHCP but without PXE
- networks without PXE/DHCP

Kernel command line options are accepted to do semi-automatic provisioning where some of forms can be pre-populated with values. Or things can be fully automated even on networks without PXE/DHCP by modifying `SYSLINUX` configuration on the discovery ISO. This essentially replaces Foreman Bootdisk plugin in scenarios where generic images (iPXE) do not work well (hardware compatibility).

Smart-proxy now listens on both HTTP/8448 and HTTPS/8443 ports and by default, new self-signed certificate is generated on each discovered node with common name set to IP address of the primary interface. Images prior 3.0 were using HTTP/8443 combination.

The image now supports UEFI systems and Apple Macintosh.

- #9488 (<http://projects.theforeman.org/issues/9488>) - PXE-less workflow with TUI interface
- #11678 (<http://projects.theforeman.org/issues/11678>) - Initial 5 fact uploads are not cached
- #9810 (<http://projects.theforeman.org/issues/9810>) - IPMI facter extensions multichannel support
- #11340 (<http://projects.theforeman.org/issues/11340>) - Improved build scripts
- #11241 (<http://projects.theforeman.org/issues/11241>) - Use NetworkManager dispatcher script instead of dhclient
- #11134 (<http://projects.theforeman.org/issues/11134>) - New fdm.inetnet option to init all interfaces
- #10831 (<http://projects.theforeman.org/issues/10831>) - New fdm.zipserver to override TFTP server for extensions
- Interactive TUI on the initial (status) screen
- Only Provisioning (PXE) NIC is initialized with DHCP by default
- System logs accessible via journald on tty2
- Removed BMC API with new Power API with kexec support
- Smart-proxy logs into systemd instead a separate file
- Dracut patch for old Fedoras removed

Contributors (in alphabetical order): Elmer Rivera, Greg Sutcliffe, Jan Ivar Beddari, Lukas Zapletal, Shlomi Zadok, Stephen Benjamin

2.1.1: Bugfixing release

2.1: New features:

- runtime extensions via http/ZIP
- Foreman Discovery Proxy support

2.0: New features:

- smaller in size (only 150 MB whole RAM disk)
- faster and easier to build (dropped ovirt-node support)
- easier downstream building (in koji)
- based on centos7 and fedora19
- systemd and network manager full support
- better troubleshooting (discovery-debug script)

1.1.4 Hammer CLI Foreman Discovery plugin

0.0.2: Bug fixing release.

0.0.1: Initial version.

2. Installation

There are three different components to install: Foreman Discovery plugin, Smart Proxy Discovery plugin (aka Smart Proxy Discovery) and Foreman Discovery Image.

2.1 Foreman Discovery plugin

Foreman Discovery plugin is the main component that needs to be installed.

2.1.1 Installer (recommended)

As of Foreman 1.8+, the foreman-installer is able to automatically install the package and proceed with all necessary steps. To do this, re-run the installer with the following option:

```
# foreman-installer --enable-foreman-plugin-discovery
```

Important note: Executing foreman-installer will re-deploy all foreman-related configuration files. In case some changes have been made, proceed with manual download described below.

2.1.2 Manual installation

To install Foreman Discovery plugin, do the following

```
# yum install tfm-rubygem-foreman_discovery
```

on Fedora and Red Hat systems or

```
# apt-get install ruby-foreman-discovery
```

on Debian and Ubuntu systems.

Restart of Foreman is needed when installing the package separately.

2.2 Smart Proxy Discovery plugin (optional)

Smart Proxy plugin can be used to discover hosts in networks Foreman can't directly reach itself. The plugin must be installed on all proxy hosts on the network where discovery feature is supposed to work.

To install the plugin, do

```
# yum install rubygem-smart_proxy_discovery
```

on Fedora and Red Hat systems or

```
# apt-get install ruby-smart-proxy-discovery
```

on Debian and Ubuntu systems.

Smart Proxy must be restarted after plugin installation and features must be refreshed for the given Proxy in the Foreman interface (*Infrastructure > Smart Proxies > a proxy > Refresh features*).

2.3 Foreman Discovery Image

The image is based on CentOS 7 and about 150 MB in size. There are two ways to install the image.

2.3.1 Download via installer (recommended)

As of Foreman 1.8+, the foreman-installer is able to automatically download latest stable image. For this, re-run the installer with the following option:

```
# foreman-installer \
  --foreman-plugin-discovery-source-url=http://downloads.theforeman.org/discovery/releases/3.0/
  --foreman-plugin-discovery-install-images=true
```

Check the version number in `foreman-plugin-discovery-source-url` option against the table above. Some image-plugin combinations are not compatible.

Tip: It is possible to install both Discovery plugin and image in one installer run by providing both the options.

Important note: Executing foreman-installer will re-deploy all foreman-related configuration files. In case some changes have been made, proceed with manual download described below.

2.3.2 Manual download

Images are available for direct download (<http://downloads.theforeman.org/discovery/>) To download the latest release to the expected location, do the following on Fedora and Red Hat systems:

```
# wget http://downloads.theforeman.org/discovery/releases/3.0/fdi-image-latest.tar \
  -O - | tar x --overwrite -C /var/lib/tftpboot/boot
```

Check the version number in the url against the table above. Some image-plugin combinations are not compatible.

On Debian systems, use `/srv/tftp/boot` instead of `/var/tftpboot/boot`.

2.3.3 Verify signatures and checksums

To verify individual ISO/tar files, import our Discovery Plugin key 7E81E7B0 (<http://keys.fedoraproject.org:11371/pks/lookup?search=0x7E81E7B0&op=get>) and verify individual files:

```
# wget 'http://keys.fedoraproject.org:11371/pks/lookup?search=0x7E81E7B0&op=get' -O- | gpg --import
# gpg --verify --multifile *.asc
gpg: Signature made Fri 06 Nov 2015 09:42:55 AM UTC using RSA key ID 7E81E7B0
gpg: Good signature from "Foreman Discovery <foreman-xxx@googlegroups.com>"
```

To verify extracted init RAM disk and linux kernel do:

```
# cat /var/lib/tftpboot/boot/fdi-image/SHA256SUM
beb3cfba7d9fb9d71481c0c8f... initrd0.img
f03bce150d2473a28697d2988... vmlinuz0
# sha256sum /var/lib/tftpboot/boot/fdi-image/*
```

On Debian systems, use `/srv/tftp/boot` instead of `/var/tftpboot/boot`.

2.3.4 Building own image

To build a discovery image, please visit the foreman-discovery-image git repository (<https://github.com/theforeman/foreman-discovery-image>) and find the README for further instructions.

2.4 Hammer CLI Foreman Discovery

2.4.1 Installation

Before installing make sure you have installed the hammer_cli_foreman plugin (<https://github.com/theforeman/hammer-cli/blob/master/doc/installation.md>),

On Red Hat compatible systems:

```
yum install tfm-rubygem-hammer_cli_foreman_discovery
```

On Debian/Ubuntu compatible systems:

```
apt-get install ruby-hammer-cli-foreman-discovery
```

Or install from rubygems:

```
gem install hammer_cli_foreman_discovery.
```

2.5 Upgrade

To upgrade Foreman Discovery follow the standard procedure of upgrading all the Foreman packages. Check Foreman Discovery global settings and PXELinux Default template.

After each upgrade, it is required to reboot all discovered hosts to load new discovery image which is compatible with the recent features added in the Foreman plugin. To do this, click on Reboot All button in the UI or do this via CLI.

3. Configuration

The following chapter covers configuration of all the components installed.

3.1 Foreman Discovery plugin

Foreman Discovery relies on intercepting the normal boot process for machines not registered in Foreman. To achieve this, the PXE *default.cfg* file needs to be altered to instruct new machines to boot the discovery image.

3.1.1 Default PXE template

In the Foreman UI, go to Provisioning Templates, edit *PXELinux global default* template and add the following after the "LABEL local" block of options:

```
LABEL discovery
MENU LABEL Foreman Discovery
MENU DEFAULT
KERNEL boot/fdi-image/vmlinuz0
APPEND initrd=boot/fdi-image/initrd0.img rootflags=loop root=live:/fdi.iso rootfstype=auto ro rd.l
ive.image acpi=force rd.luks=0 rd.md=0 rd.dm=0 rd.lvm=0 rd.bootif=0 rd.neednet=0 rd.debug=1 nomde
set proxy.url=https://FOREMAN_SERVER proxy.type=foreman
IPAPPEND 2
```

The *proxy.type* option can be either *proxy* or *foreman*. In the first case all communication goes through Smart Proxy, in the latter case the communication goes directly to Foreman (legacy mode). This is the default when not specified.

The *proxy.url* specifies URL of the Smart Proxy or Foreman depending on the previous setting. For backward compatibility, *foreman.url* is an alias for this setting.

In case Smart Proxy Discovery plugin is configured to forward communication, configure the APPEND line in the following way:

```
APPEND ... proxy.url=https://FOREMAN_PROXY_INSTANCE:8443 proxy.type=proxy
```

Once the APPEND line is modified properly, set the entry to be default via the ONTIMEOUT option:

```
ONTIMEOUT discovery
```

Alternatively, the Discovery image also searches for DNS SRV record named *_x-foreman._tcp*. If the DNS server is configured for this (see the example for ISC BIND below), then it is not required to provide *foreman.url* at all, but *proxy.type* is still required if the SRV record is configured to point to a proxy.

Here is an example of ISC BIND configuration statement to set *foreman* A record to be used with HTTPS protocol.

```
_x-foreman._tcp SRV 0 5 443 foreman
```

You can mix both approaches and override SRV record with the command line.

It is important to keep *IPAPPEND 2* option which is key configuration option to detect interface connected to provisioning network. Also never change or remove *root* option, otherwise image will not boot properly.

It is important to know that DNS servers from DHCP are taken into account for only for the interface that was specified via the BOOTIF option. This means when a system has multiple NICs, DNS will work for the correct interface - the one that was booted from.

3.1.2 Update PXE default template

Once the template has been updated, click the *Build PXE Default* button at the top of the Provisioning Templates page. This will instruct the TFTP proxy to rewrite the *pxelinux.cfg/default* file.

Repeat this step every time a change is made to the default template.

3.1.3 Organizations and Locations

If Locations and/or Organizations are enabled, Foreman assigns Organization and Location to discovered hosts according to the following rules from top to bottom:

- According to the *discovery_organization* or *discovery_location*, if present. These can be set under *Administer > Settings > Discovered*.
- If *foreman_organization* or *foreman_location* fact is present, set accordingly. Fact names which are looked up can be configured in *Administer > Settings > Puppet* section as *Organization/Location fact* setting.
- If Subnet was determined for particular discovered host, use the first Organization and Location associated with the Subnet.
- Select the first Organization / Location ordered by title (name the org/loc was created with). This is a workaround until we fix provisioning of discovered hosts without taxonomy (<http://projects.theforeman.org/issues/4426>).

Organization or Location can be changed via the "bulk actions" menu which appears once once or more discovered hosts are selected.

3.1.4 Global settings

There is also setting called `discovery_fact` which defaults to `discovery_bootif`. It specifies which incoming fact should be used to get the MAC address. By default, the PXELinux BOOTIF kernel command line option is used which gives the MAC address of the interface which was booted from.

Note that `auto_discovery` option is set to `false` by default. If you want to trigger provisioning automatically with rules, you need to turn this setting on. It's recommended to try provision manually first to test before proceeding.

It is possible to add any fact reported by Factor via `discovery_fact_column` global option onto the Discovered Hosts page table as a new column. To do that, set the value of this setting to name of a fact reported. To hide the new column, set to a blank value. To show multiple columns, separate fact names by comma.

3.1.5 Discovery image kernel options

Foreman discovery image currently recognizes the following list of configuration options:

- `proxy.url` - URL to Foreman or Proxy (if omitted DNS SRV lookup is made)
- `proxy.type` - endpoint type: "foreman" or "proxy"
- `fdi.zips` - extensions to download
- `fdi.zipserver` - override TFTP server reported by DHCP
- `fdi.initnet=all/bootif` - initialize all or pxe NICs (default) during startup
- `fdi.ssh` - configure ssh daemon after start (1 to enable)
- `fdi.rootpw` - configure ssh daemon password (plain string)
- `fdi.countdown` - initial countdown in seconds before registration attempt (15 by default)
- `fdi.uploadsleep` - seconds between facter runs (30 by default)
- `fdi.cachefacts` - number of fact uploads without caching (0 by default)
- `fdi.px*` - PXE-less workflow (described below)

By default, the image tries to send initial facts multiple times. Delay between facter runs can be changed with `fdi.uploadsleep`. Once facts are successfully sent to Foreman server, additional uploads are done only on hardware configuration change (e.g. disk drive is added). It is possible to force arbitrary amount of uploads via `fdi.cachefacts` option. Reasonable value is 2-3 and also increase `fdi.uploadsleep` to more than 60 seconds to prevent race conditions when provisioning hosts (host could be discovered second time when provisioning window is hit).

3.2 Smart Proxy Discovery plugin

Make sure `foreman_url` setting is present in the Smart Proxy configuration file.

```
# grep foreman_url /etc/foreman-proxy/settings.yml
:foreman_url: https://FOREMAN_HOST
```

This should be done automatically by our installer. It is a good idea to check if the host responds properly and there are no firewalls blocking the communication.

3.2.1 Subnet proxy setup

All subnets with discovered nodes need this specified in Foreman so it connects via the Smart Proxy. To do this, go to *Infrastructure > Smart Proxies* and verify if the desired proxy lists *Discovery* feature. If not, click on *Refresh features* button and it will appear immediately.

In *Infrastructure > Subnets* select the desired *Discovery Proxy* for each appropriate Subnet.

Make sure that lease pool (defined in ISC DHCP configuration) and reservation pool (defined in *Infrastructure > Subnets*) are disjoint. More on that topic in the installation scenarios (manuals/latest/index.html#3.2.3InstallationScenarios) section.

3.3 Hammer CLI Foreman Discovery plugin

The plugin must be enabled in `cli.modules.d/foreman_discovery.yml` (see Hammer config directories (<https://github.com/theforeman/hammer-cli/blob/master/doc/installation.md#locations>)) as follows:

```
:foreman_discovery:
  :enable_module: true
```

3.4 Permissions

The plugin will create a Role called *Discovery* when first started. This can be assigned to roles for non-admins to allow them to use the discovery plugin. Alternatively assign the `perform_discovery` permission to an existing Role.

4. Usage

Foreman Discovery plugin provides user interface, API and CLI.

4.1 Hardware discovery

Boot a machine in any provisioning network that was configured with the default PXE configuration above. It should register with Foreman and appear in *Hosts > Discovered Hosts*.

4.2 Manual provisioning

Select a discovered host and choose *Provision*. This will redirect to the normal Edit page for a Host, with the discovered data filled in where possible. Fill in the details as normal.

On save, Foreman modifies the host's PXELinux file on the TFTP server and reboots the discovered host, after which it boots into an installer for the chosen OS, and finally into the installed OS.

Delete a machine and reboot it to have it move back to the Discovery Pool.

4.3 Automatic provisioning

Starting with version 2.0, it is possible to predefine provisioning rules which will assign host group to provisioned hosts and trigger provisioning automatically. To do that, head over to *Configure > Discovery rules* and create such a rule:

- **Name** represents rule name shown to the users. It must not contain spaces or non-alphanumeric characters.
- **Search** statement is used to match discovered hosts for the particular rule. Use Scoped Search syntax to define it. Examples are shown below.
- **Host Group** is assigned to a matching host before starting provisioning process. It is very important the selected Host Group has all the required parameters set (domain, subnet, root password), otherwise provisioning process will fail.
- **Hostname** defines a pattern to assign human-readable hostnames to the matching hosts. When left empty, hostname is assigned in a form of `macMACADDRESS` by default. The same syntax as for provisioning templates is used. See below for more information and examples.
- **Hosts limit** enables to limit maximum amount of provisioned hosts per rule. If a limit was reached, the rule will not take effect until one or more hosts are deleted. Typical use case are rules per server rack or row when it is necessary to change provisioning parameters like hostname or host group per each entry.
- **Priority** puts the rules in order. Must be greater than zero and low numbers go first. Rules are always matched by priority given.
- **Enabled** flag is used for temporary shutdown of rules.
- **Organizations\Locations** discovered hosts can only auto provision from rules in the same organization/location as the discovered host. Host group taxonomy is enforced, therefore it is not possible to provision into a host group that is not associated with the discovery rule.

Once some rules are defined, the good practice is to discover a host and apply the rules using Auto discover button on the host.

By default, Foreman does not trigger auto discovery automatically. This must be explicitly turned on in *Administer > Settings > Discovered > discovery_auto*.

4.3.1 Search syntax

Easiest way of testing search patterns is in Discovered hosts list, because the search box gives the same results. Typical search fields are facts, they all start with "facts.". Auto completion can be used to browse the facts as well as discovered hosts detail screen. Typical search queries:

```
facts.architecture = x86_64
facts.bios_vendor ~ 'Dell*'
facts.macaddress = "aa:bb:cc:dd:ee:ff"
facts.macaddress_eth0 = "aa:bb:cc:dd:ee:ff"
facts.ipaddress_eth1 ~ "192.168.*"
```

As of Foreman 1.7, all the facts are currently simple strings, so it is not possible to do numeric comparisons.

Some important facts are extracted and converted to numbers. These are:

- `cpu_count` - number of CPUs
- `disk_count` - number of disks attached
- `disks_size` - total amount of disk space (in MiB)

Possible queries are:

```
cpu_count >= 8
disk_count < 10
disks_size > 1000000
```

See the searching section ([manuals/latest/index.html#4.1.5Searching](https://foreman.com/manuals/latest/index.html#4.1.5Searching)) for more information.

4.3.2 Hostname patterns

The target hostname template pattern has the same syntax as in Provisioning Templates (ERB). Domain will be appended automatically. A hostname based on MAC address will be used when left blank. In addition to `@host` attribute function `rand` for random integers is available. Examples:

```
application-server-<%= rand(99999) %>
load-balancer-<%= @host.facts['bios_vendor'] + '-' + rand(99999) %>
wwwsrv-<%= @host.hostgroup.name %>
minion-<%= @host.discovery_rule.name %>
db-server-<%= @host.ip.gsub('.', '-') + '-' + @host.hostgroup.subnet.name %>
```

When creating hostname patterns, make sure the resulting host names are **unique**. This is very important. Hostnames must not start with numbers. A good approach is to use unique information provided by `facter` (MAC address, BIOS or serial ID) or to randomize the hostname somehow.

4.4 Hammer CLI Foreman Discovery plugin

Confirm your setup by running `hammer -h` and check that the discovery command is listed.

```
$ hammer -h
...
Subcommands:
...
discovery                Discovery related actions.
...
...
```

The actions you can use with discovery will appear as follows:

```

$ hammer discovery -h
Usage:
  hammer discovery [OPTIONS] SUBCOMMAND [ARG] ...

Parameters:
SUBCOMMAND          subcommand
[ARG] ...           subcommand arguments

Subcommands:
auto-provision      Auto provision a host
delete              Delete a discovered host
facts               Show a discovered host
info                Show a discovered host
list                List all discovered hosts
provision           Provision a discovered host
reboot              Reboot a host
refresh-facts       Refresh the facts of a host

Options:
-h, --help          print help

```

For example to reboot a discovered_host:

```

$ hammer discovery reboot -h
Usage:
  hammer discovery reboot [OPTIONS]

Options:
--id ID              Name to search by
--name NAME          Name to search by
-h, --help           print help

$ hammer discovery reboot --id 130
Host reboot started

```

5. Extending the image

It is possible to extend the Foreman Discovery Image with custom facts, software or device drivers easily. There are two ways of doing that.

5.1 Runtime extensions

It's possible to provide a zip file containing extra code for the image to use. First, create a directory structure like:

```

.
├── autostart.d
│   └── 01_zip.sh
├── bin
│   └── ntpdate
├── facts
│   └── test.rb
├── lib
│   ├── libcrypto.so.1.0.0
│   └── ruby
│       └── test.rb

```

`autostart.d` contains scripts that will be executed in POSIX order by the image as it starts up, before the host is registered to Foreman. `bin` is added to `PATH`, you can place binaries here and use them in the autostart scripts. `facts` is added to `FACTERLIB` so that custom facts may be configured and sent to Foreman. `lib` is added to `LD_LIBRARY_PATH` and `lib/ruby` is added to `RUBYLIB`, so that binaries in `bin` can be executed properly.

Environment variables (`PATH`, `LD_LIBRARY_PATH`, `RUBYLIB` and `FACTERLIB`) are appended to. If you need to specify the path to something explicitly in your scripts, the zip contents are extracted to `/opt/extension` on the image.

An example structure is provided in `example_zip` in this repo. You can zip up your structure with `zip -r my_extension.zip .`

You can create multiple zip files, but be aware they will be extracted to the same place on the discovery image, so files in later zips will overwrite earlier ones if they have the same filename.

To inform the image of the extensions it should use, place your zip(s) on your TFTP server along with the discovery image, and then update your PXELinux `APPEND` line with `fdi.zips=<path-to-zip>, <path-to-zip>`, where the paths are relative to the TFTP root. So if you have two zips at `$TFTP/zip1.zip` and `$TFTP/boot/zip2.zip`, you would use `fdi.zips=zip1.zip, boot/zip2.zip`.

5.2 Image building with extensions

The very same ZIP file with the structure described above can be injected into your own image. This is useful for network drivers when there is a chicken and egg problem.

To build a discovery image with extensions, please visit the `foreman-discovery-image` git repository (<https://github.com/theforeman/foreman-discovery-image>) and find the `README` for further instructions.

5.3 PXE-less discovery

It is possible to use the Discovery image directly as a CDROM/DVDROM ISO that can be also transferred to hard drive or USB stick. Download the image from our site and transfer it onto a flash drive:

```

wget http://downloads.theforeman.org/discovery/releases/3.0/fdi-bootable-3.0.X.iso
dd if=fdi-bootable-3.0.X.iso of=/dev/sdx

```


In this case, automatic discovery process is not started automatically and the user can select the primary network interface interactively. On the next couple of screens, the user configures network credentials (IPv4 supported at the moment), the URL and type (Foreman or Smart Proxy), and fills custom user facts which are then uploaded to the server. These custom facts can be used with pre-defined discovery rules to let users automatically build hosts.

Image sends special "kexec" fact (flag) along standard and user facts which is a signal to Discovery plugin to initiate kexec reload instead of reboot. When the host is provisioned either manually or via discovery rule, instance loads installer via kexec utility.

This interactive mode is also available when image is PXE booted, user must cancel initial splash screen countdown with a key to get there.

Currently there is no authentication in the workflow (the same applies for normal discovery process), but it is possible to define own custom facts named "token" or "password" to secure PXE-less provisioning.

Before a system can be discovered without PXE there must be a discovery rule associated with a hostgroup having an OS associated with a "kexec" template kind. This template is used to pass in parameters for kexec call (kernel, initrd, append line). Discovery ships with templates for Red Hat and Debian distributions at the moment.

Make sure the associated provisioning template is configured with static networking, if the deployment does not provide DHCP services, otherwise Anaconda will fail to install the system after kernel reload. Also check if netmask, gateway and primary DNS server is present on the associated Subnet.

5.3.1 Unattended and semi-automatic mode

All interactive screens can be fully automated via kernel command line that can be either entered during SYSLINUX/Grub2 boot screen or burnt into the image (see below). Options are self-explanatory:

```
proxy.url=https://foreman:443
proxy.type=foreman
fdi.pxmac=aa:bb:cc:dd:ee:ff
fdi.pxip=192.168.122.42/24
fdi.pxgw=192.168.1.1
fdi.pxdns=192.168.1.1
fdi.pxfactname1=security-token
fdi.pxfactvalue1=987123
fdi.pxfactname2=deployment
fdi.pxfactvalue2=db-server
fdi.pxfactnameN=
fdi.pxfactvalueN=
```

When px.ip and/or px.gw are omitted, image tries to acquire network credentials over DHCP, which can be useful in PXE-less environments with DHCP server deployed. Therefore the absolute minimum set of options in this case is:

```
proxy.url=https://foreman:443
proxy.type=foreman
fdi.pxmac=aa:bb:cc:dd:ee:ff
```

All values must be confirmed in the text-user interface first, which gives semi-automatic behavior. To initiate the provisioning without any question (unattended mode or fully automatic), add the following option:

```
fdi.pxauto=1
```

When fdi.pxauto is not provided, Ok/Next buttons are always focused first on all screens therefore it is easier to walk through.

The pxmac option defines the provisioning interface to be used to acquire network credentials (optionally) and send facts. If omitted in unattended mode, the first NIC with link is picked up (in alphabetical order by network identifier in case of multiple items). In semi-automated mode, screen will appear to select the correct interface.

5.3.2 Remastering Discovery Image

Unattended and semi-automatic mode described above requires some kernel command line options to be present. To avoid typing them in in PXELinux/Grub2 loaders, it is possible to re-master the discovery image "burning" the options into the image.

Helper script called `discovery_remaster` (<https://github.com/theforeman/foreman-discovery-image/blob/master/aux/remaster/discovery-remaster>) is being shipped with the `foreman_discovery` plugin. It can be used to make a copy of the ISO image with additional kernel command line options. Usage is simple:

```
sudo discovery-remaster fdi-bootable-3.0.X.iso "fdi.pxip=192.168.100.68/24 fdi.pxgw=192.168.100.1
fdi.pxdns=192.168.100.1 proxy.url=https://192.168.100.1:8443 proxy.type=proxy fdi.pxfactname1=myf
act fdi.pxfactvalue1=somevalue fdi.pxmac=52:54:00:be:8e:8c fdi.pxauto=1"
```

The above command creates a copy in the same directory with date/time stamp in the name.

5.3.2 Example workflows

Provisioning without PXE/DHCP

- Setup auto-provisioning rules (optional)
- Transfer the ISO image onto an USB stick or CDROM and boot it.
- Select *Manual network setup*
- Select primary (provisioning) interface if the host has multiple NICs
- Enter IPv4 credentials
- Enter Foreman (Smart Proxy) credentials
- Provide custom facts and click on Confirm
- The host is discovered
- The host reloads kernel into installer either manually or via auto-provisioning

Provisioning with DHCP (no PXE)

- Transfer the ISO image onto a USB stick or CDROM and boot it.
- Select *Discover with DHCP*

- Select primary (provisioning) interface if the host has multiple NICs
- Enter Foreman (Smart Proxy) credentials
- Provide custom facts and click on Confirm
- The rest is same as in the example above (without DHCP)

Semi-automated workflow w/o PXE/DHCP

- Modify the discovery ISO with the following options:
 - `fdi.pxgw=192.168.1.1`
 - `fdi.pxdns=192.168.1.1`
 - `proxy.url=https://foreman:443`
 - `proxy.type=foreman`
 - `fdi.pxfactname1=deployment_type`
 - `fdi.pxfactvalue1=database_server`
- Transfer the ISO image onto a USB stick or CDROM and boot it.
- Select primary (provisioning) interface if the host has multiple NICs
- Enter IPv4 credentials (gateway and DNS is already populated)
- Select primary (provisioning) interface
- Enter Foreman (Smart Proxy) credentials
- Provide custom facts (the first is pre-populated) and click on Confirm
- The rest is same as in the examples above

Fully automated workflow w/o PXE/DHCP

- Modify the discovery ISO and set all required options including MAC and IP address
- Make sure the `fdi.pxauto` option is set to `1`
- Transfer the ISO image onto a USB stick or CDROM and boot it.
- Discovered node automatically uploads facts and reloads kernel into installer.

6 Help

Please follow our standard procedures and contacts ([support.html](#)). For problems with the image, send us the output of the following command for a running discovered node (see below how to get access to the shell):

```
# discovery - debug
```

6.1 Troubleshooting

If you find a bug, please file it in Redmine (<http://projects.theforeman.org/projects/discovery/issues/new>).

See the troubleshooting section ([manuals/latest/index.html#7.2GettingHelp](#)) in the Foreman manual for more info.

6.1.1 Troubleshooting plugins

If the booted machine fails to register with Foreman, then there are a number of common causes:

- Machine did not boot the correct image
 - Check your default PXE Linux template and try to re-deploy it
 - Check your `pxelinux.cfg/default` configuration file on the TFTP Proxy
- Machine booted image but failed to contact Foreman
 - Check the `proxy.url` and `proxy.type` options
 - Check DNS is working for that image, or use an IP in `proxy.url`
 - Check DHCP is handing IPs to the booted image correctly
- Discovered node does not reboot
 - Check the connection between Foreman and Proxy, Proxy and Node
 - For PXE-less discovery verify that `kexec` template was associated with the OS

6.1.2 Troubleshooting the image

First of all make sure your server (or VM) has more than 500 MB of memory because less memory can lead to various random kernel panic errors as the image needs to be extracted in-place (150 MB * 2).

The first virtual console is reserved for logs, all systemd logging is shown there. Particularly useful system logs are tagged with:

- `discover-host` - initial facts upload
- `foreman-discovery` - facts refresh, reboot remote commands
- `nm-prepare` - boot script which pre-configures NetworkManager
- `NetworkManager` - networking information

The root account and ssh access are disabled by default, but you can enable ssh and set root password using the following kernel command line options. Simply add these to the APPEND line of the pxe template:

```
fdi.ssh=1 fdi.rootpw=redhat
```

Use `tty2` console (or higher) to login onto a discovered host.

6.1.3 Blacklisting drivers

Since the image is based on CentOS 7, all kernel options are valid and should work normally, including `modprobe.blacklist` to blacklist a driver in init RAM disk.

6.1.4 Maximum length of command line

CentOS 7 distribution ships with `COMMAND_LINE_SIZE` option set to 2048. Therefore kernel command line must not be longer than that.

6.1.5 Initial restart problem

In typical Foreman workflow, hosts are set to always boot from the network via PXE. Unknown hosts boot into Discovery while known hosts boot into installer or boot from hard drive when in operation mode.

Some users use Discovery plugin on virtualized platforms. Hypervisors tend to treat the initial restart in a special way to allow comfortable OS installation. They usually attempt to change boot order to first hard drive instead of CD-ROM or PXE which was only used for the installation.

There is a known issue with libvirt which turns VM off instead of rebooting it which causes problems when using Discovery. To avoid this behavior and to align with recommended workflow, new VMs must be created with PXE booting only. In Virt Manager, click on Customize options and set the boot order explicitly. When using `virt-install`, do not provide `--pxe` option and set boot order explicitly with `--boot network` instead.

6.2 UEFI

Discovery Image version 3.0.0 and older does support booting via EFI thanks to Grub 2 which is present on the media alongside SYSLINUX. Remastered images preserve this capability as well.

6.3 Contributing

Follow the same process as Foreman ([contribute.html#SubmitPatches](https://github.com/foreman/foreman/blob/master/CONTRIBUTING.md#submitting-patches)) for contributing.

Conference season is here! Join us at FOSDEM & Config Management Camp, 2nd - 6th February 2019

The 2019 Foreman Community Survey is now open! Click here to participate!

(https://docs.google.com/forms/d/e/1FAIpQLSdBsoDwG5BDNOieVVGERTmPcZWvIaKua5D2dWOOKmiLZbPNHg/vusp=sf_link)

Foreman 1.20.1 has been released! Follow the quick start to install it. ([manuals/1.20/quickstart_guide.html](https://foreman.io/docs/manual/1.20/quickstart_guide.html))

Foreman 1.19.1 has been released! Follow the quick start to install it. ([manuals/1.19/quickstart_guide.html](https://foreman.io/docs/manual/1.19/quickstart_guide.html))

Foreman 1.18.3 has been released! Follow the quick start to install it. ([manuals/1.18/quickstart_guide.html](https://foreman.io/docs/manual/1.18/quickstart_guide.html))

[View our Privacy Policy \(/privacy.html\)](#)

This web site is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License (http://creativecommons.org/licenses/by-sa/3.0/deed.en_GB). Source available: github.com/foreman/foreman (<https://github.com/foreman/foreman>)

Thanks to our sponsors ([/sponsors.html](#)) for hosting this website and our project services.

