The Foreman

Doina Cristina Duma, cristina.aiftimiei<at>cnaf.infn.it
Diego Michelotto, diego.michelotto<at>cnaf.infn.it
INFN-CNAF

Corso Ansible/Foreman/Puppet, Bari, 5 - 9 Giugno 2018

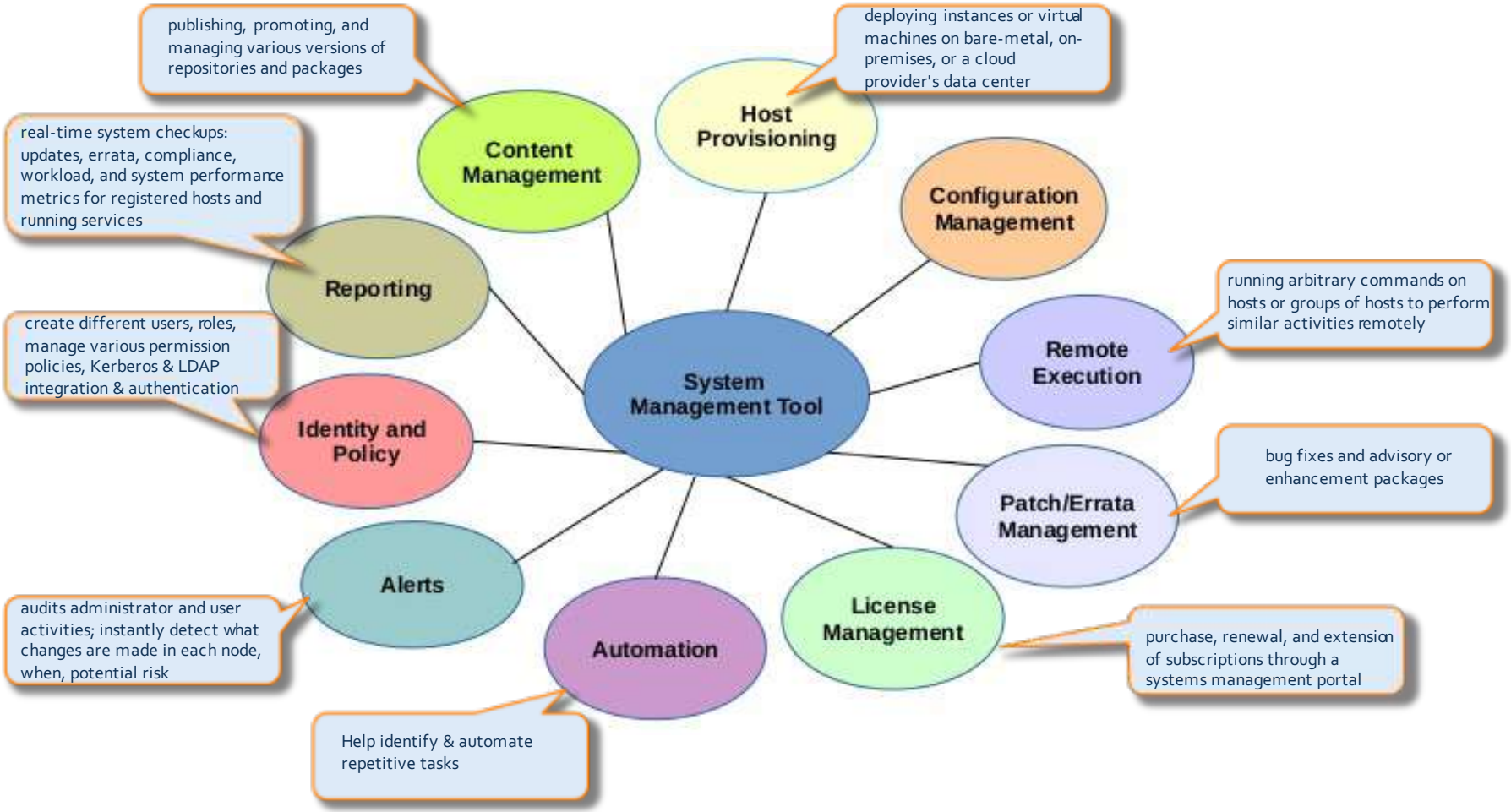
Outline

- **The Foreman**
  - Introduction
- **Architecture**
  - Smart Proxies, Config. Manager, Compute Resources
- Provisioning
- Configuration
- Monitoring
- References

Is your system management tool robust enough?

Image by : opensource.com

Effective system management tool



The Foreman's realm

- A Typical System Life-Cycle

Pre/ Installation

Initial Configuration

Updates

Drift Management

Audits

Foreman is

+ A complete lifecycle management tool for physical and virtual servers.

Provisioning

Provision on bare-metal & public or private clouds all from one place with one simple process.

Configuration

A complete configuration management solution including an ENC for Puppet, built-in support for parameterized classes and hierarchical parameter storage.

Monitoring

Inventory and activity reporting based on Puppet reports and facts.
Including configuration status and inventory distribution and trends.

Introduction to Foreman
- what is all about -

Facts

- Project started in 2009
- Licensed under the GPLv3
- Development pushed by Red Hat
- Very active & helpful community

Overview

- Tool for provisioning of VMs & bare metal
- Provides config management & monitoring integration
- Rails & JavaScript application
- Exposes a web interface, REST API & CLI

Strong & Weak points

Strong Suite

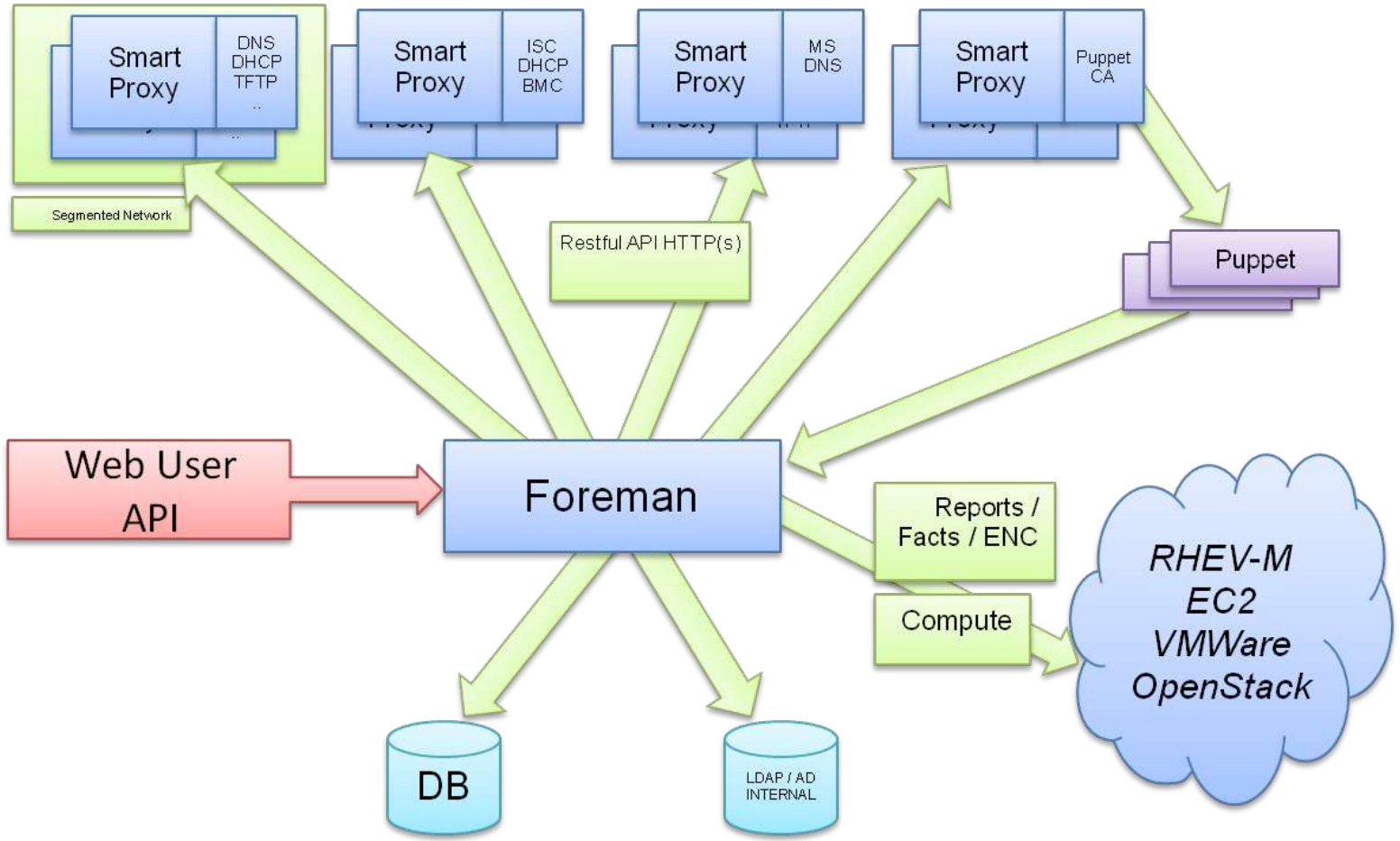
- Very flexible
- Offers tons of features
- Active development & open community
- Modular setup, start small then expand
- Can serve as a source of truth (CMDB)
- Can be used as an ENC
- Proper ACL implementation
- Enterprise Support available (Red Hat Satellite 6)

Weak Spots

- Somewhat steep learning curve
- Can be quite tricky to debug an issue
- API has room for improvement
- Offers sometimes too many possible ways to implement a task

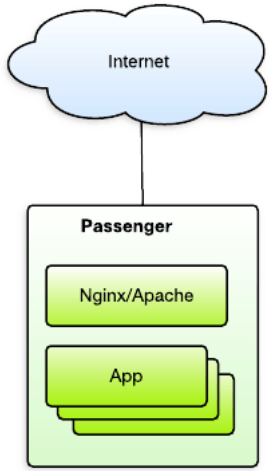
Architecture
- Overview of the different components -

Bird's Eye view

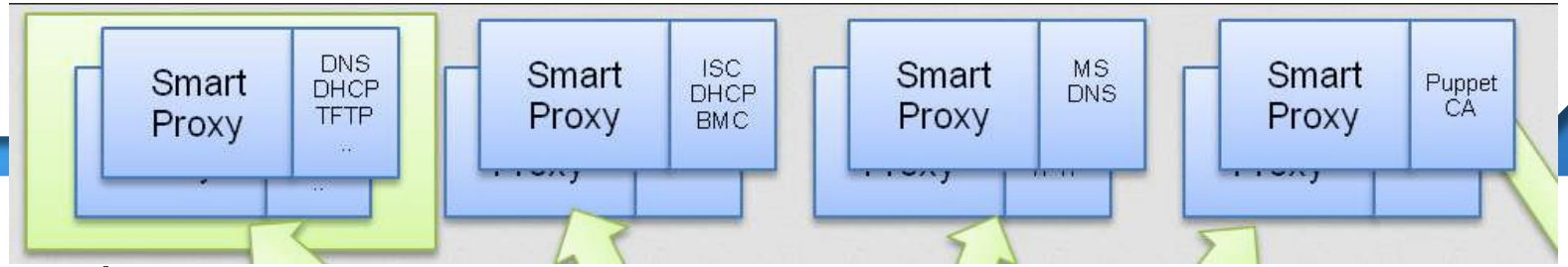


Foreman

- Heart of the whole stack
  - Central instance that is responsible for providing the Web based GUI, node configurations, initial host configuration files
- Stores all resources & information
- Rails stack, use Passenger + nginx / Apache to run it
- Stores most data in a DB (SQLite, MySQL or PostgreSQL)
- Local or LDAP users for authentication

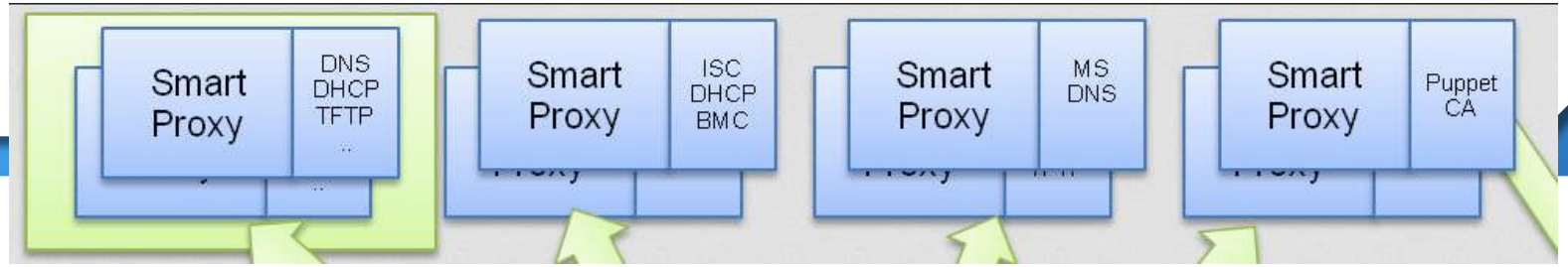


Smart Proxies



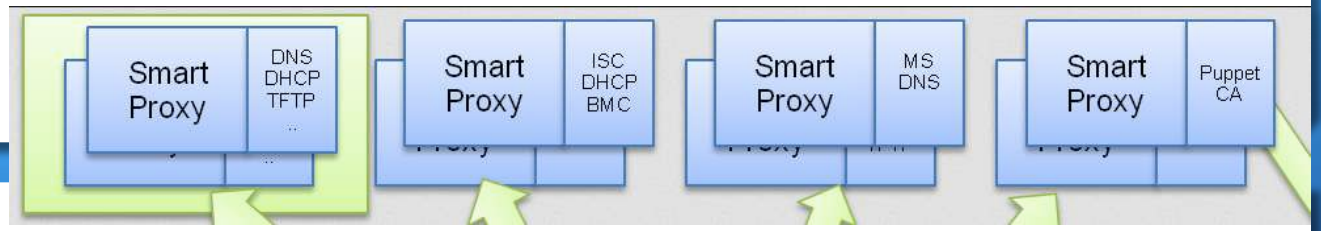
- Small autonomous HTTP application
- Exposes a REST API to provide different services
- Allows Foreman to control components in isolated networks
  - located on or near a machine (reduces latencies) that performs a specific function and helps foreman ***orchestrate the process of commissioning a new host***
- Also called ***foreman-proxy***

Smart Proxies



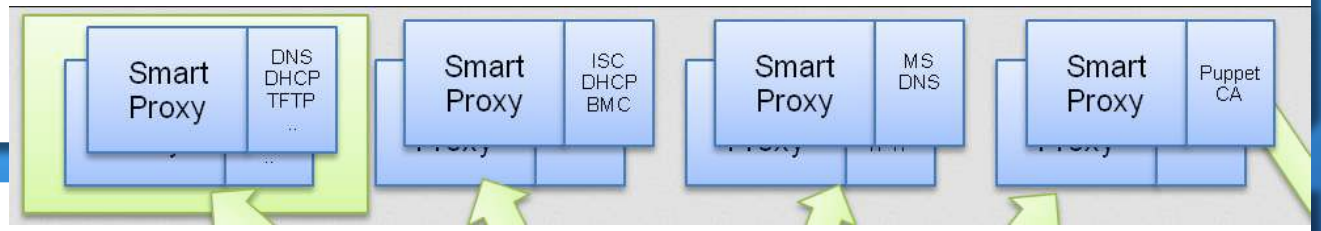
- Currently supported
  - **DHCP** - ISC DHCP and MS DHCP Servers
  - **DNS** - Bind and MS DNS Servers
  - **Puppet** - Any Puppet server from 0.24.x
  - **Puppet CA** - Manage certificate signing, cleaning and autosign on a Puppet CA server
  - **Realm** - Manage host registration to a realm (e.g. FreeIPA)
  - **Templates** - Proxy template requests from hosts in isolated networks
  - **TFTP** - Any UNIX based tftp server

Smart Proxies - DHCP



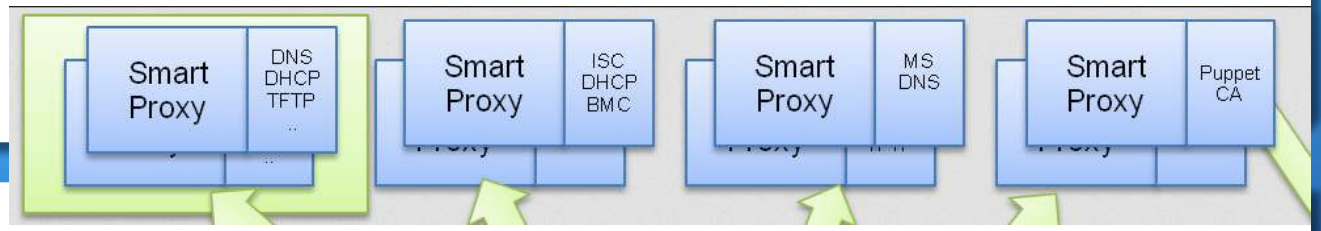
- Takes care of reserving the required IPs
- Provides IP auto-assignment
- Supports ISC DHCP (over OMAPI), MS DHCP & libvirt
- More providers can be installed or developed (e.g. InfoBlox)

Smart Proxies - DNS



- Update and remove DNS records automatically
- Takes care of A, AAAA & PTR records
- Supports Bind, PowerDNS, Route53, MS DNS Server, Libvirt
- More providers can be installed or developed (e.g. AWS53)

Smart Proxies - TFTP



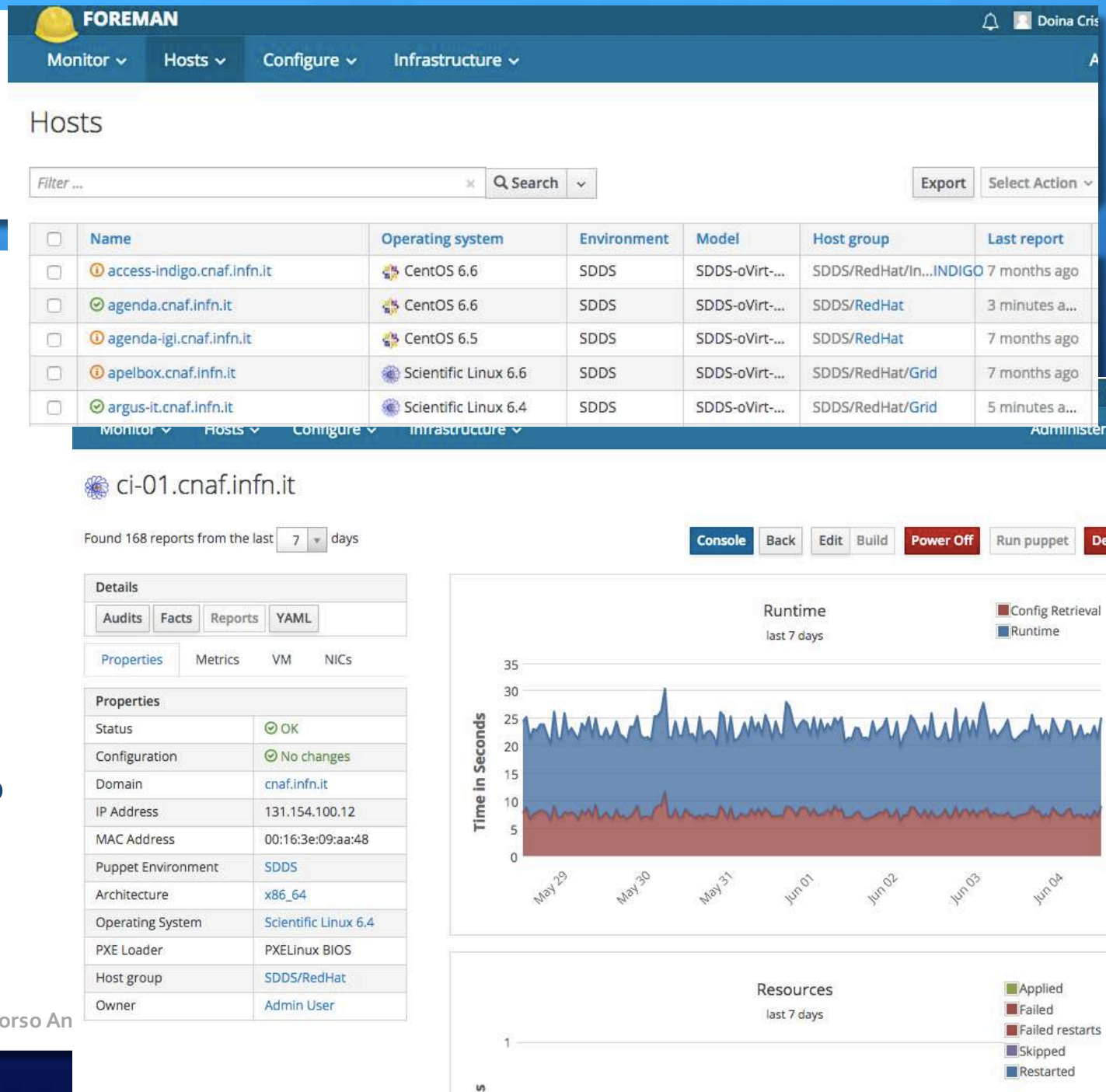
- Provide images during PXE boot
- Automagically downloads kernel + initrd (installer)
- Prepares MAC specific config depending on the build state
- Fallback to default

Terminology

- Host
- Installation media
- Provisioning templates
- Partition tables

Host

- Foreman concept that represents a ***server/host/system/computer***
- In addition to holding facts about the system, it:
  - Stores which ***operating system*** the system should be running
  - Stores which ***puppet classe***s should be assigned
  - Stores which ***parameters*** apply to which puppet classes
  - Allows you to ***re-provision*** the machine



Host

Host | Interfaces | Puppet Classes | Parameters | Additional Information

Included Config Groups
- SDDS-redhat
- SDDS-base

Available Config Groups
- SDDS-debian

Add

Included Classes

Inherited Classes from RedHat
- cnprov_puppetrun
- ntp
- puppet
- resolv_conf
- sdds_backuppc
- sdds_fail2ban
- sdds_motd
- sdds_packages
- sdds_rundeck
- sdds_security_checks
- sdds_security_updates
- sdds_sensu
- sdds_ssh
- sdds_yum
- stdlib

Available Classes

Filter classes
- apache
- apt
- archive
- bind
- cnaf_crond
- cnaf_puppet
- cnlog_beaver
- cnmon_sensu
- cnmon_sensucert
- cnprov_puppetrun
- cnprov_security
- concat
- dhcp
- dirtree
- dns
- docker

FOREMAN

Monitor | Hosts | Configure | Infrastructure | Administer

Edit ci-01.cnaf.infn.it

Manage host | Disassociate ho

Host | Interfaces | Puppet Classes | Parameters | Additional Information

	Identifier	Type	MAC Address	IPv4 Address	IPv6 Address	FQDN	Actions
	eth0	Interface physical	00:16:3e:09:aa:48	131.154.100.12		ci-01.cnaf.infn.it	Edit Delete
	virbr0	Interface virtual	52:54:00:2a:d5:77	192.168.122.1			Edit Delete
	virbr0_nic	Interface virtual attached to virbr0	52:54:00:2a:d5:77				Edit Delete

+ Add Interface

FOREMAN

Doina Cristina Duma

Monitor | Hosts | Configure | Infrastructure | Administer

dit ci-01.cnaf.infn.it

Manage host | Disassociate host

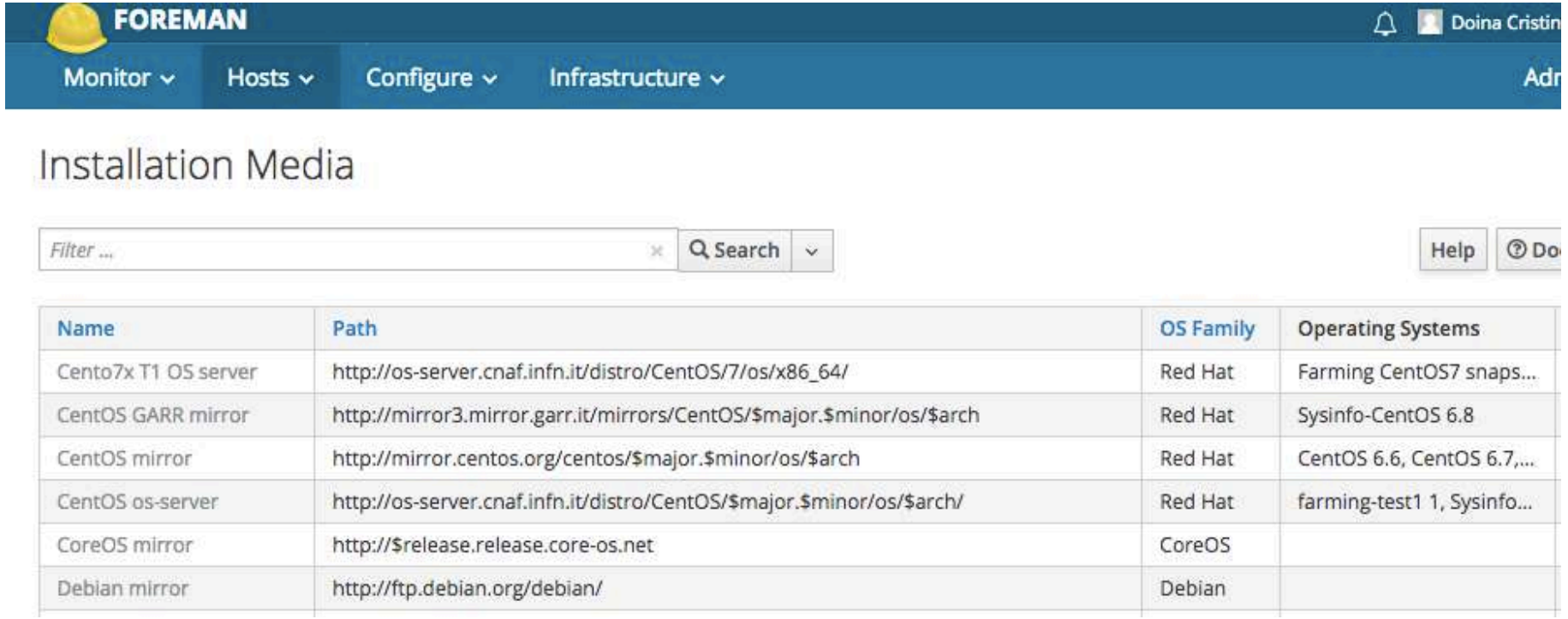
Host | Interfaces | Puppet Classes | Parameters | Additional Information

Puppet Class Parameters

Puppet Class	Name	Value	Omit
cnprov_puppetrun	manage_sudo	true	
	autoupgrade	true	
ntp	server_list	["ripe.cnaf.infn.it","ntp-3.infn.it","ntp-1.infn.it"]	
	server_options	iburst	
	service_enable	true	
	service_ensure	running	
puppet	ca_server	cnprov-pupca01.cr.cnaf.infn.it	
	cron_cmd	${$puppet::params::cron_cmd}	
	runinterval	3600	
	runmode	cron	
	splaylimit	3600	
	version	present	
resolv_conf	nameservers	["131.154.3.1","131.154.1.3"]	
	options	["timeout:1","rotate attempts:1"]	

Installation media

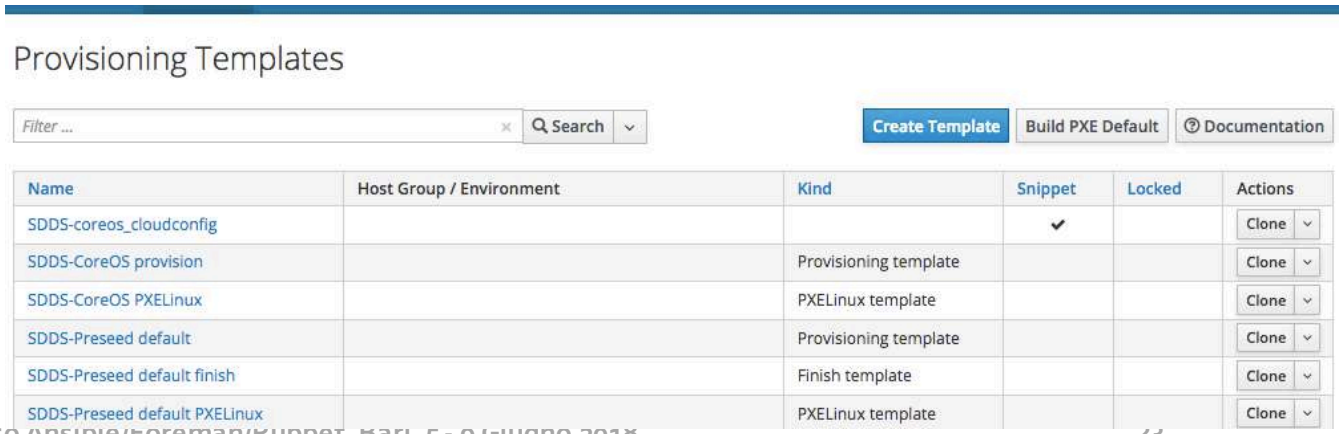
- the ***web URL*** from where the installation packages can be retrieved (i.e the OS mirror)
  - Some OS Media is pre-created for you when Foreman is first installed



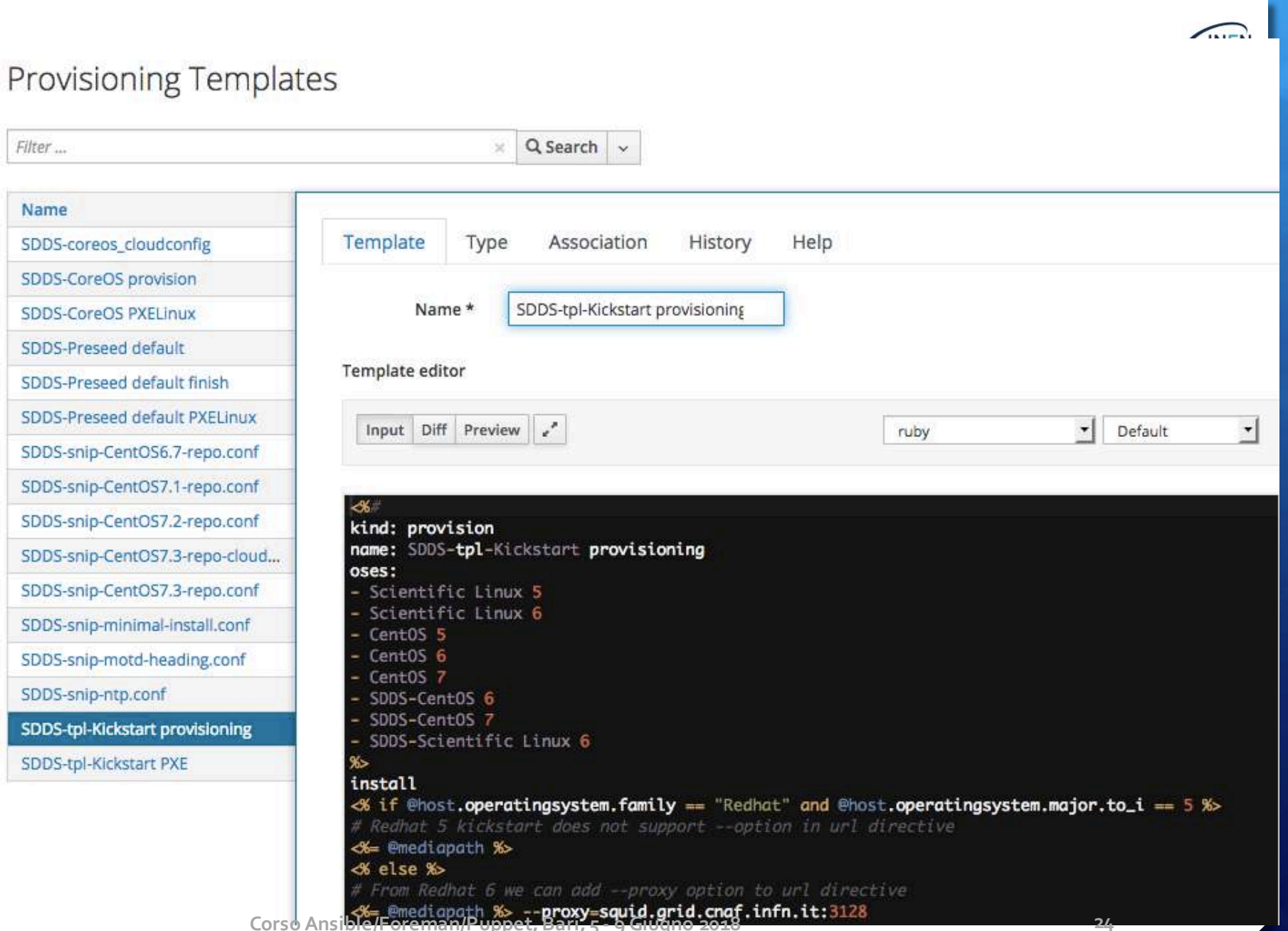
Name	Path	OS Family	Operating Systems
Cento7x T1 OS server	http://os-server.cnaf.infn.it/distro/CentOS/7/os/x86_64/	Red Hat	Farming CentOS7 snaps...
CentOS GARR mirror	http://mirror3.mirror.garr.it/mirrors/CentOS/$major.$minor/os/$arch	Red Hat	Sysinfo-CentOS 6.8
CentOS mirror	http://mirror.centos.org/centos/$major.$minor/os/$arch	Red Hat	CentOS 6.6, CentOS 6.7,...
CentOS os-server	http://os-server.cnaf.infn.it/distro/CentOS/$major.$minor/os/$arch/	Red Hat	farming-test1 1, Sysinfo...
CoreOS mirror	http://$release.release.core-os.net	CoreOS	
Debian mirror	http://ftp.debian.org/debian/	Debian	

Provisioning templates

- the core of Foreman's flexibility to deploy the right OS with the right options.
- ***several types*** of template + a ***flexible matching system*** to deliver different templates to different Hosts
  - ***PXELinux***, PXEGrub, PXEGrub2 - Deployed to the TFTP server to ensure the Host boots the correct installer with the correct kernel options (also referred to as PXE templates)
  - ***Provision*** - The main unattended installation file; e.g. Kickstart or Preseed
  - ***Finish*** - A post-install script used to take custom actions after the main provisioning is complete
  - user_data - Similar to a Finish script, this can be assigned to hosts built on user_data-capable images (e.g. Openstack, EC2, etc)
  - Script - An arbitrary script, not used by default, useful for certain custom tasks
  - iPXE - Used in {g,i}PXE environments in place of PXELinux (do not confuse with PXE templates above)
- pre-created templates for the mors common Os
  - community-templates repository



Provisioning templates



Partition tables

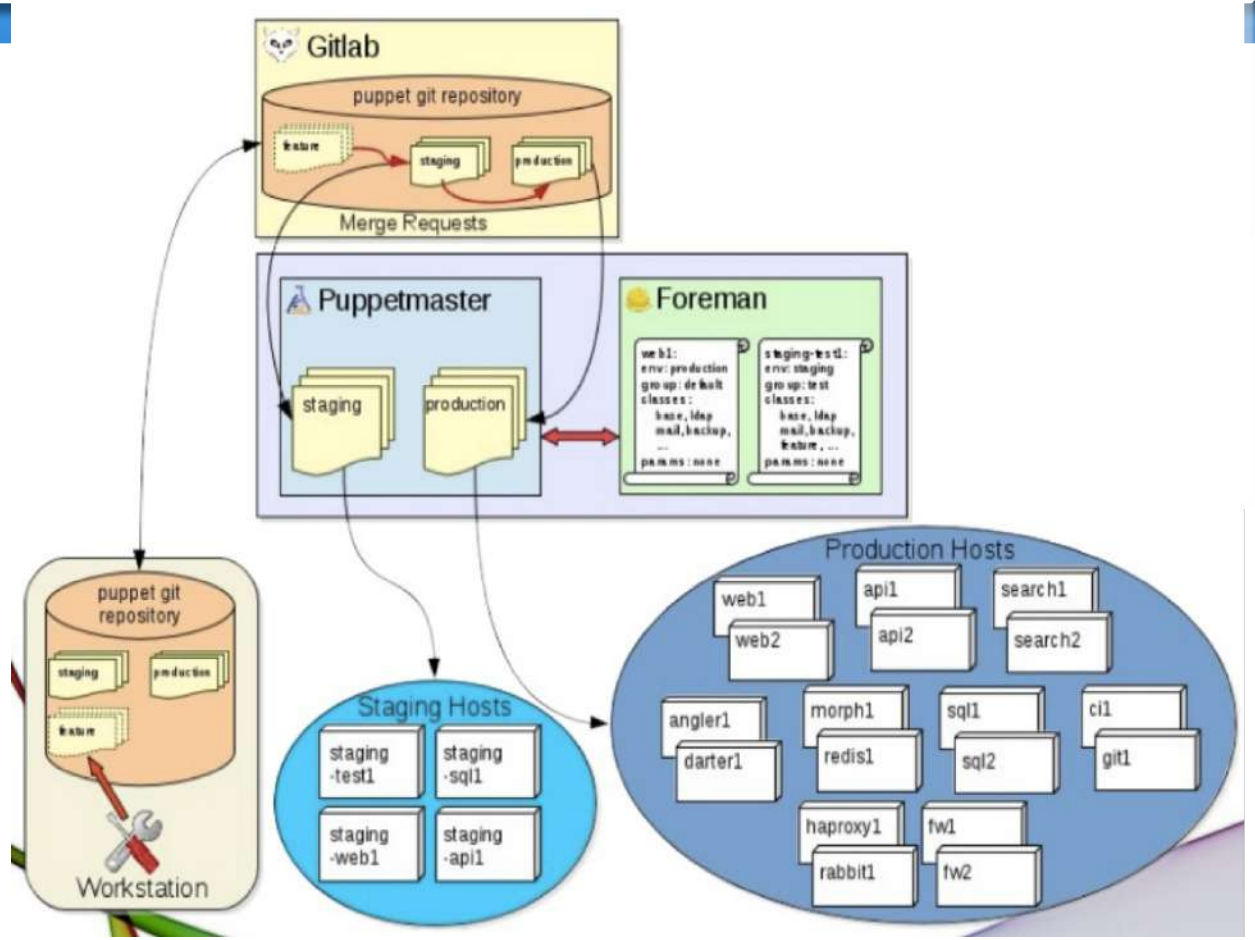
- subset of normal Provisioning Templates
  - ***handled separately*** - an admin wants to deploy the same host template (packages, services, etc) with just a different harddisk layout to account for different servers' capabilities
  - pre-created templates for common Operating Systems – editable
  - **Per-Host Partition tables**
  - **Dynamic Partition tables**
    - Some operating systems (*Kickstart* and *AutoYaST*) allow the creation of partition tables via scripts

Terminology

- Environment
- Compute resources
- Compute profiles

Environment

- Puppet environments ("*isolated groups of Puppet agent nodes*") are mapped directly into Foreman
- Generally used to ***separate classes*** from different types of Host
  - allowing changes to a module to tested in one environment (e.g. development) before being pushed to another (e.g production)



Environments

FOREMAN

Doina Cristina Duma

Monitor | Hosts | Configure | Infrastructure | Administer

Puppet Classes

environment = SDDS × Search

Import | Documentation

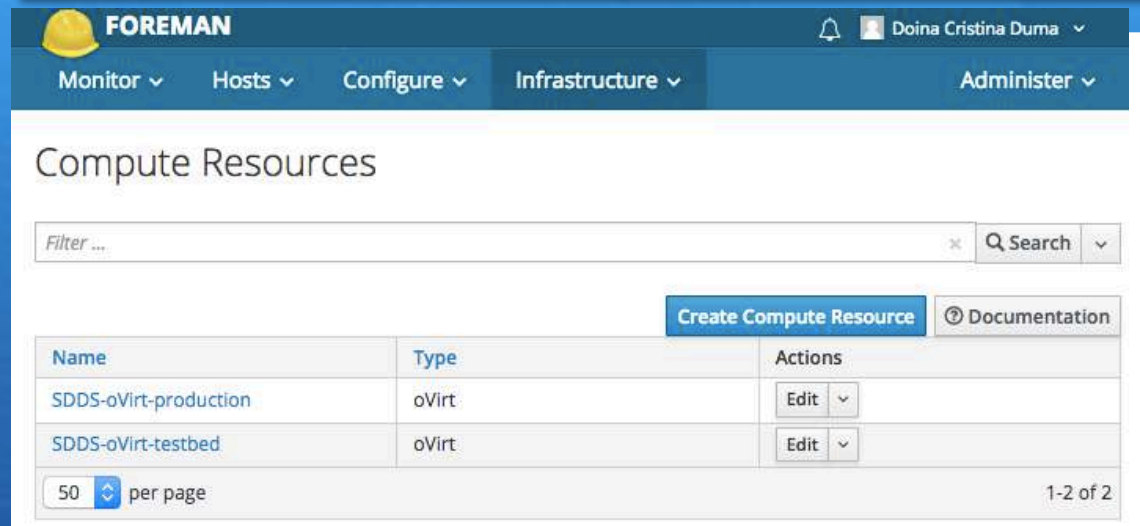
Class name	Environments	Host Groups	Hosts	Parameters	Variables	Actions
apache	SDDS		0	73	0	
apache::confd::no_accf	SDDS		0	0	0	
apache::default_confd_files	SDDS		0	1	0	
apache::default_mods	SDDS		0	4	0	
apache::dev	SDDS		0	0	0	
apache::mod::actions	SDDS		0	0	0	
apache::mod::alias	SDDS		0	3	0	
apache::mod::auth_basic	SDDS		0	0	0	
apache::mod::auth_cas	SDDS		0	24	0	
apache::mod::auth_kerb	SDDS		0	0	0	
apache::mod::auth_mellon	SDDS		0	7	0	
apache::mod::authn_core	SDDS		0	1	0	
apache::mod::authn_dbd	SDDS		0	8	0	
apache::mod::authn_file	SDDS		0	0	0	

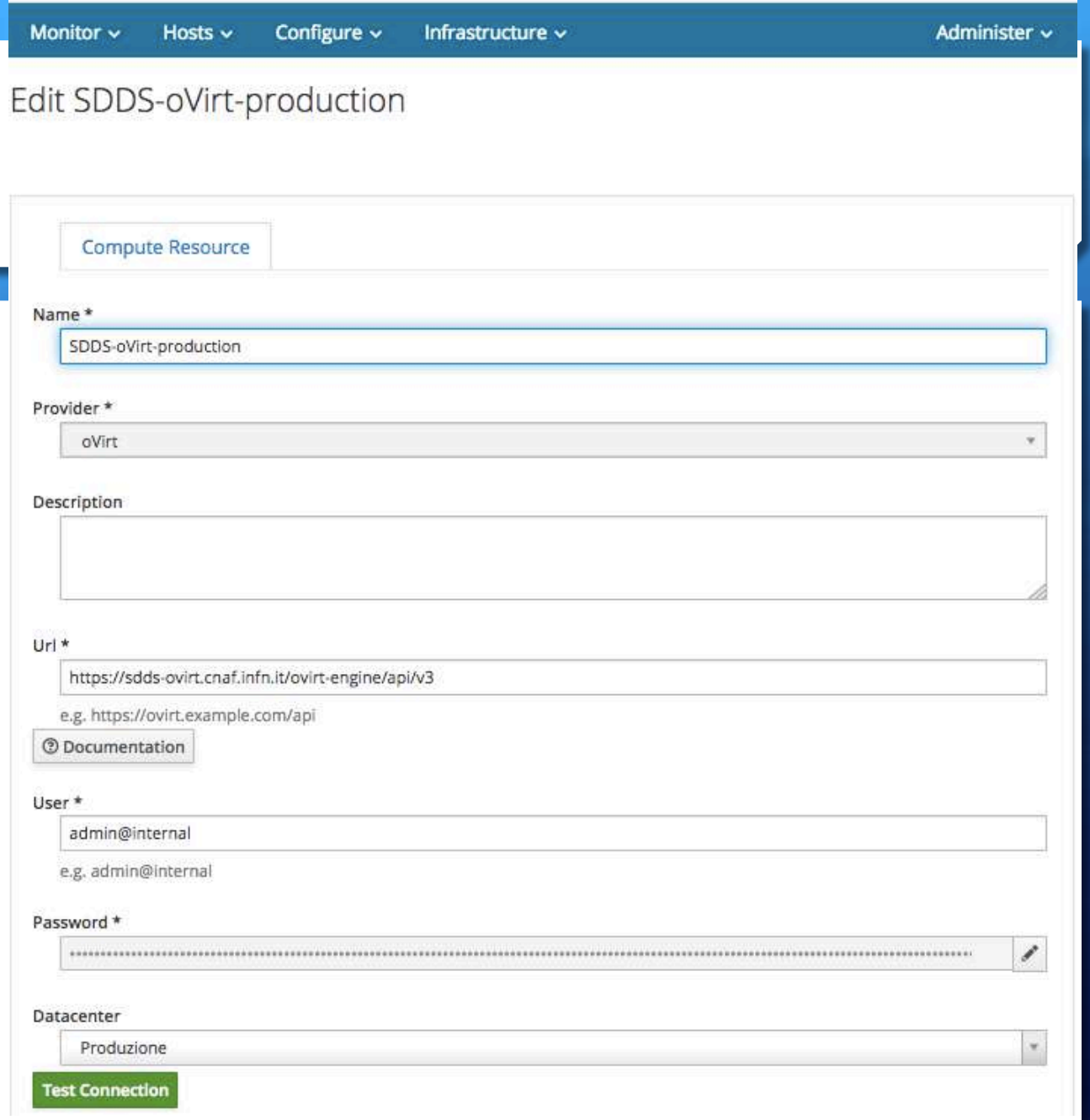
Compute Resources

- Foreman supports creating and managing hosts on a number of ***virtualization and cloud services*** - referred to as ***"compute resources"*** - as well as bare metal hosts

Provider	Package	Unattended installation	Image-based	Console	Power management	Networking
EC2	foreman-ec2	no	yes	read-only	yes	IPv4
Google Compute Engine	foreman-gce	no	yes	no	yes	IPv4
Libvirt	foreman-libvirt	yes	yes	VNC or SPICE	yes	MAC
OpenStack Nova	foreman-openstack	no	yes	no	yes	IPv4
oVirt / RHEV	foreman-ovirt	yes	yes	VNC or SPICE	yes	MAC
Rackspace	foreman-rackspace	no	yes	no	yes	IPv4 + IPv6
VMware	foreman-vmware	yes	yes	VNC	yes	MAC

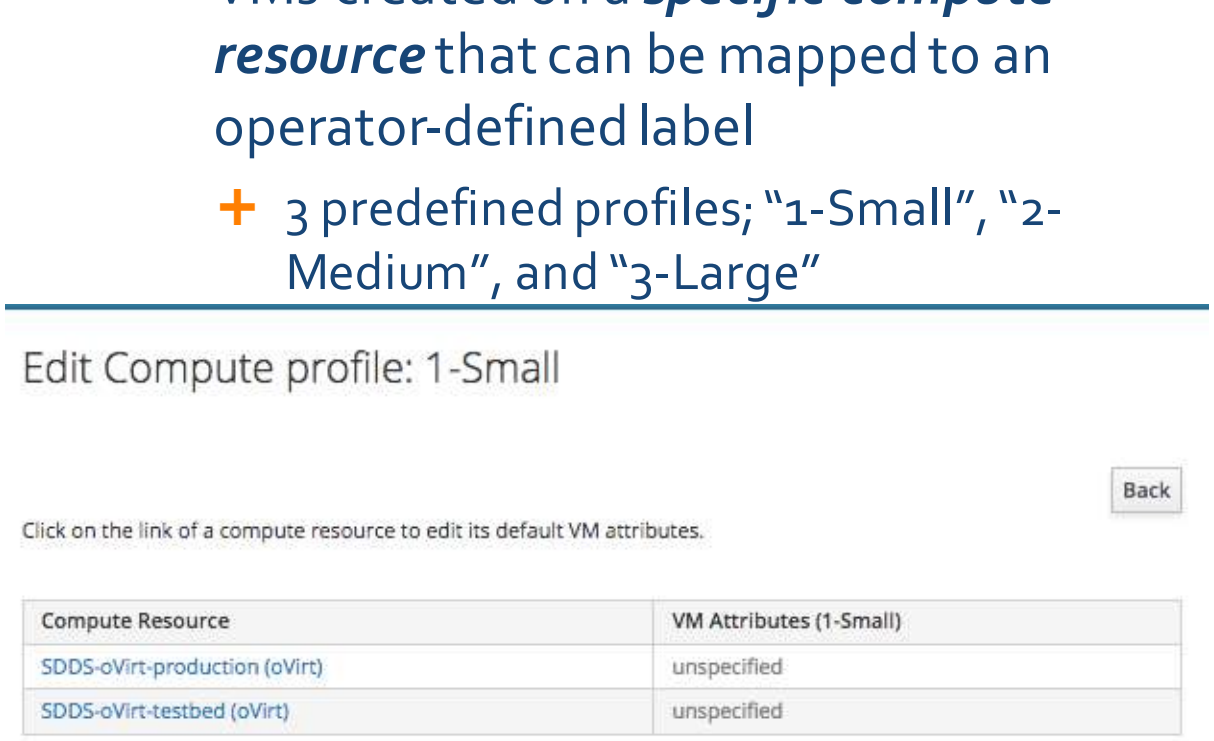
Compute Resources

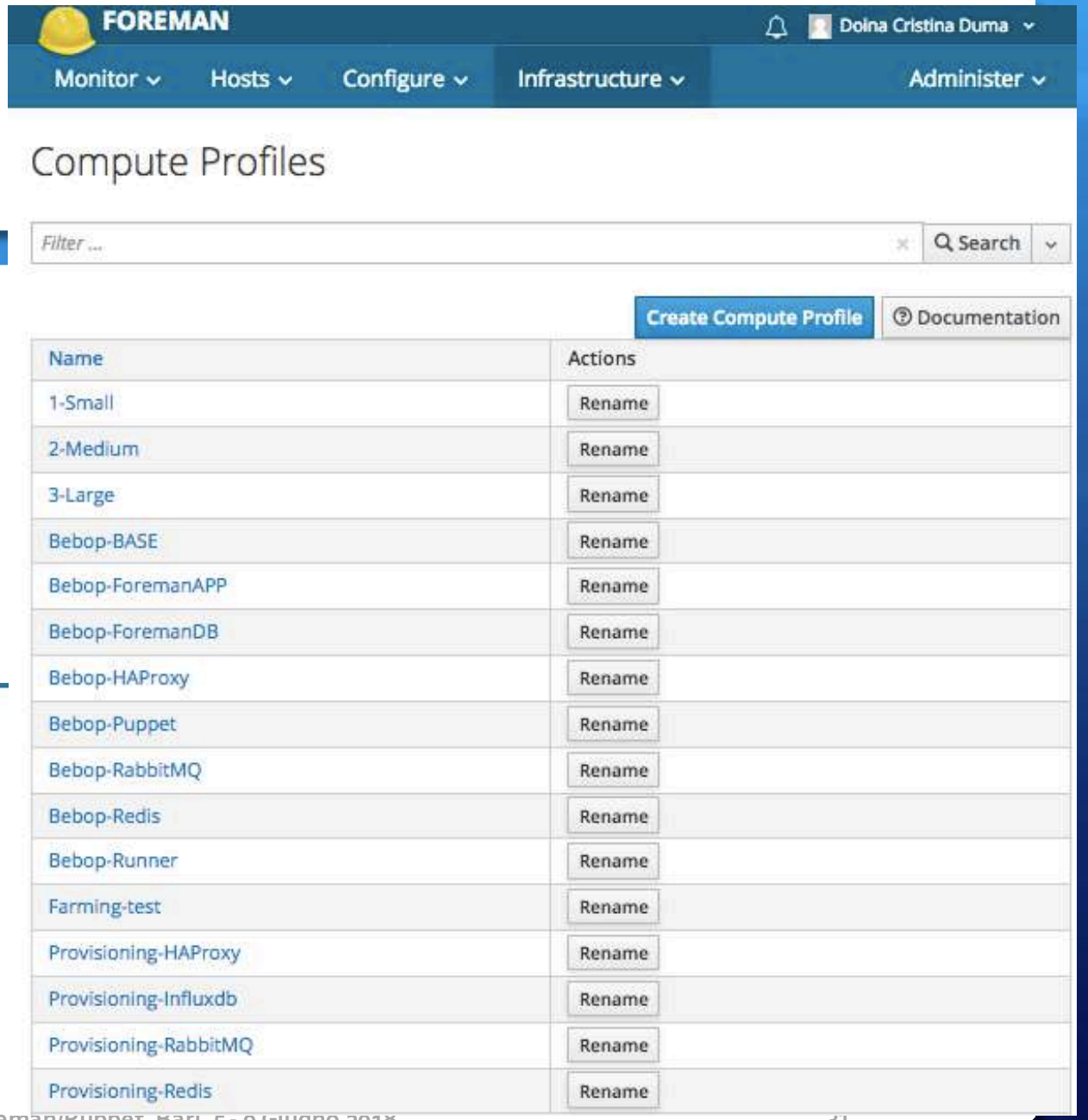




Compute profiles

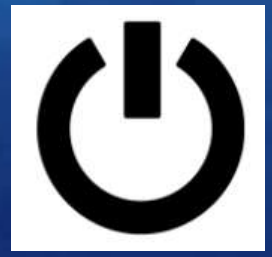
- A way of expressing a ***set of defaults*** for VMs created on a ***specific compute resource*** that can be mapped to an operator-defined label
  - 3 predefined profiles; "1-Small", "2-Medium", and "3-Large"





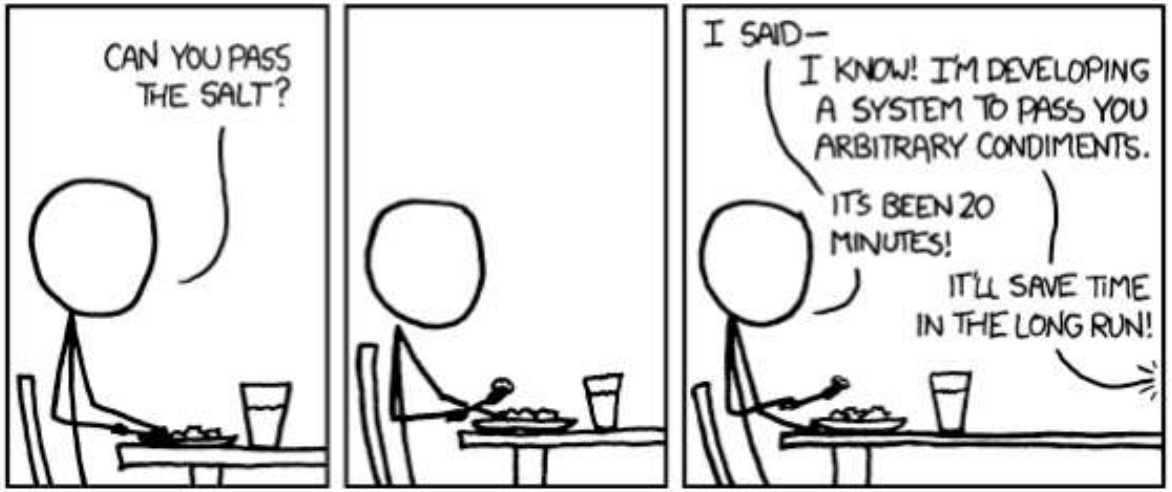
Provisioning

- Making deployments as easy as pie -



Provisioning

- Provisioning includes all the tasks required to setup a new machine
- Saving time isn't the main goal
- Enforce consistency across all deployments is key



Workflow

1. Boot the installer
   - PXE Boot (TFTP provided by Foreman)
   - ISO image
   - iPXE image
2. Start the installation
   - Tell the installer where further instructions are located
   - Red Hat Kickstart

     ```
     ks=http://foreman.example.com/unattended/provision
     ```

   - Debian Preseed

     ```
     url=http://foreman.example.com/unattended/provision
     ```

   - Defined as kernel parameters when loading the installer
3. Get further instructions from Foreman

Installer instructions

- Foreman provides ***templating functionality***
- ERB templates are rendered per host
- Contain variables, loops, snippets, etc.
- See provisioning templates & partition tables
- Templates:
  - Foreman provides community templates
  - *Vanilla* templates are locked by default
  - Can be deleted but some are mandatory (e.g. PXELinux global default)
  - Partition tables are used to define the filesystem layout – Provisioning, Finish, …

Requirements

- For a complete provisioning workflow we need some resources:
  - Architecture
    - ***x86_64***
  - Installation media (mirror)
    - ***http://mirror.centos.org/centos/$version/os/$arch***
  - OS
    - ***CentOS 7***
  - Templates
    - ***Default FS Layout, Kickstart & Finish script***

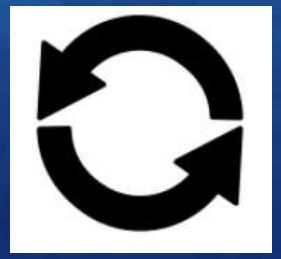
=> *Follow the Hands-on!*

The Foreman - Provisioning

- Provision new machines or containers to (almost) anything
  - Bare metal, oVirt, libvirt, VMware, Docker, EC2, Rackspace, Digital Ocean, OpenStack, etc.
- Provisioning types:
  - PXE: via PXELinux and kickstart, preseed, AutoYAST, etc
  - Image-based: cloning, configured over SSH or user data
- For virtualization provider, Foreman create VM
- For everything Foreman orchestrates related services through Smart Proxies:
  - DNS
  - DHCP / TFTP
  - Configuration Management

Configuration

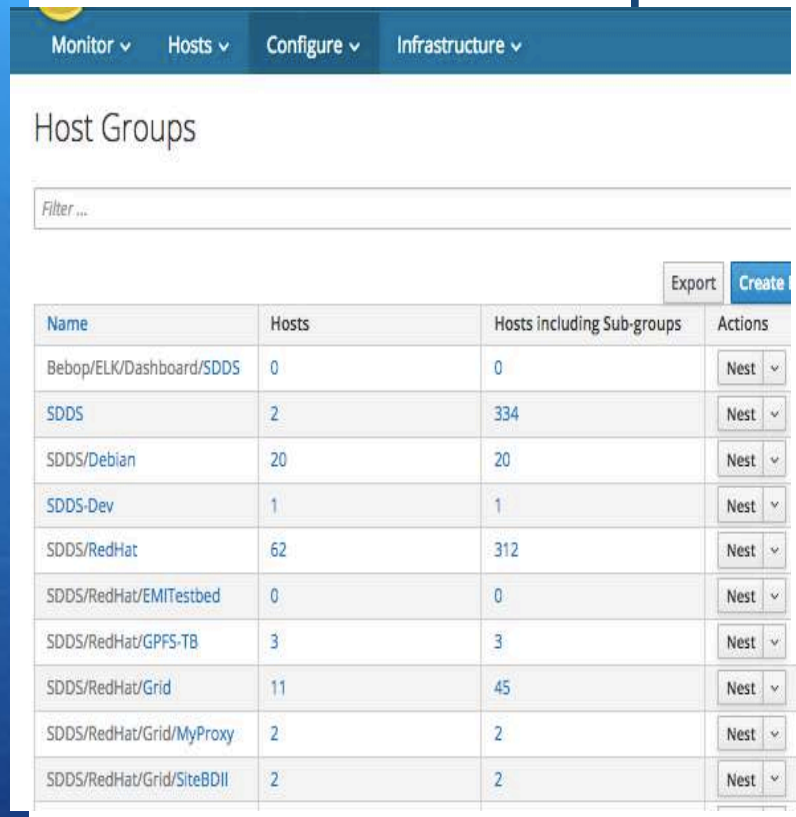
- Bring order into your organization -



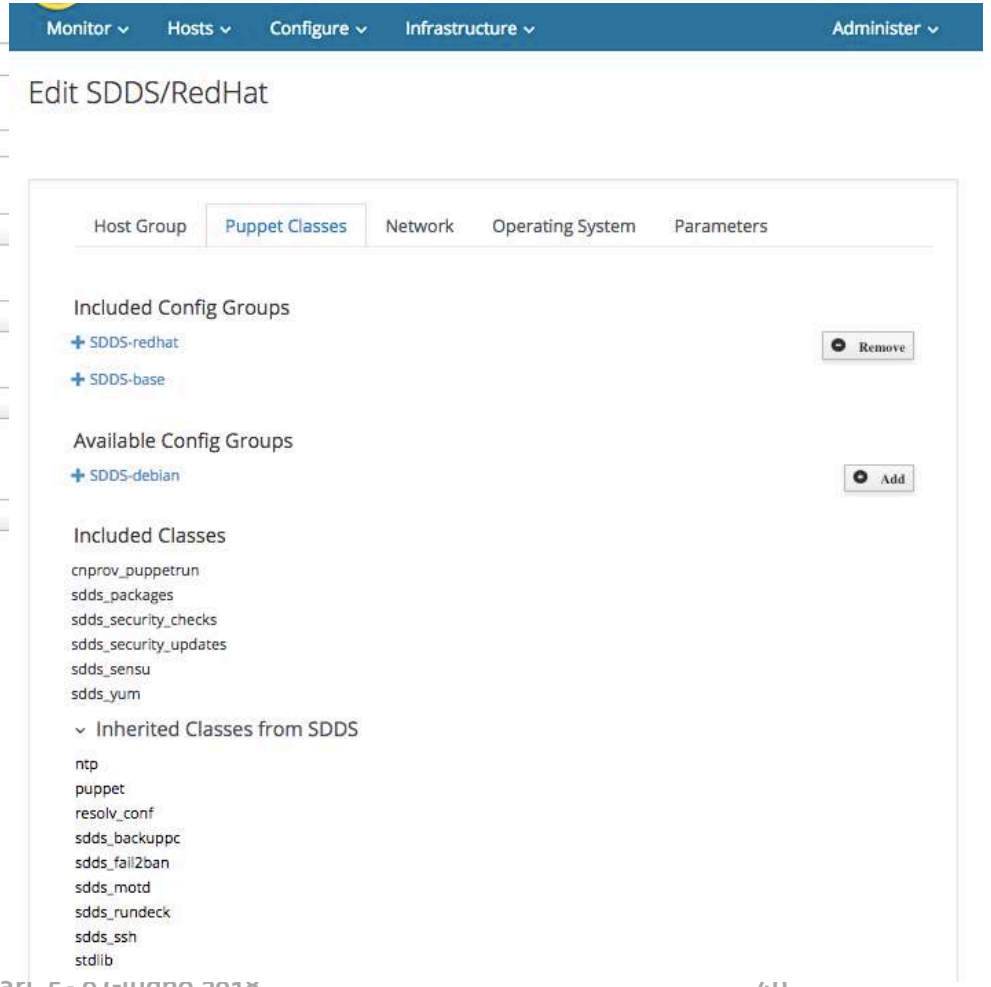
Structure

- Foreman provides different resources to organize hosts:
  - Hostgroup
    - inherited node declaration = a high level grouping of classes that can be named and treated as a unit.
    - treated as a template and is selectable during the creation of a new host
  - Domains
  - Environments
  - Organizations & Locations

Host Groups

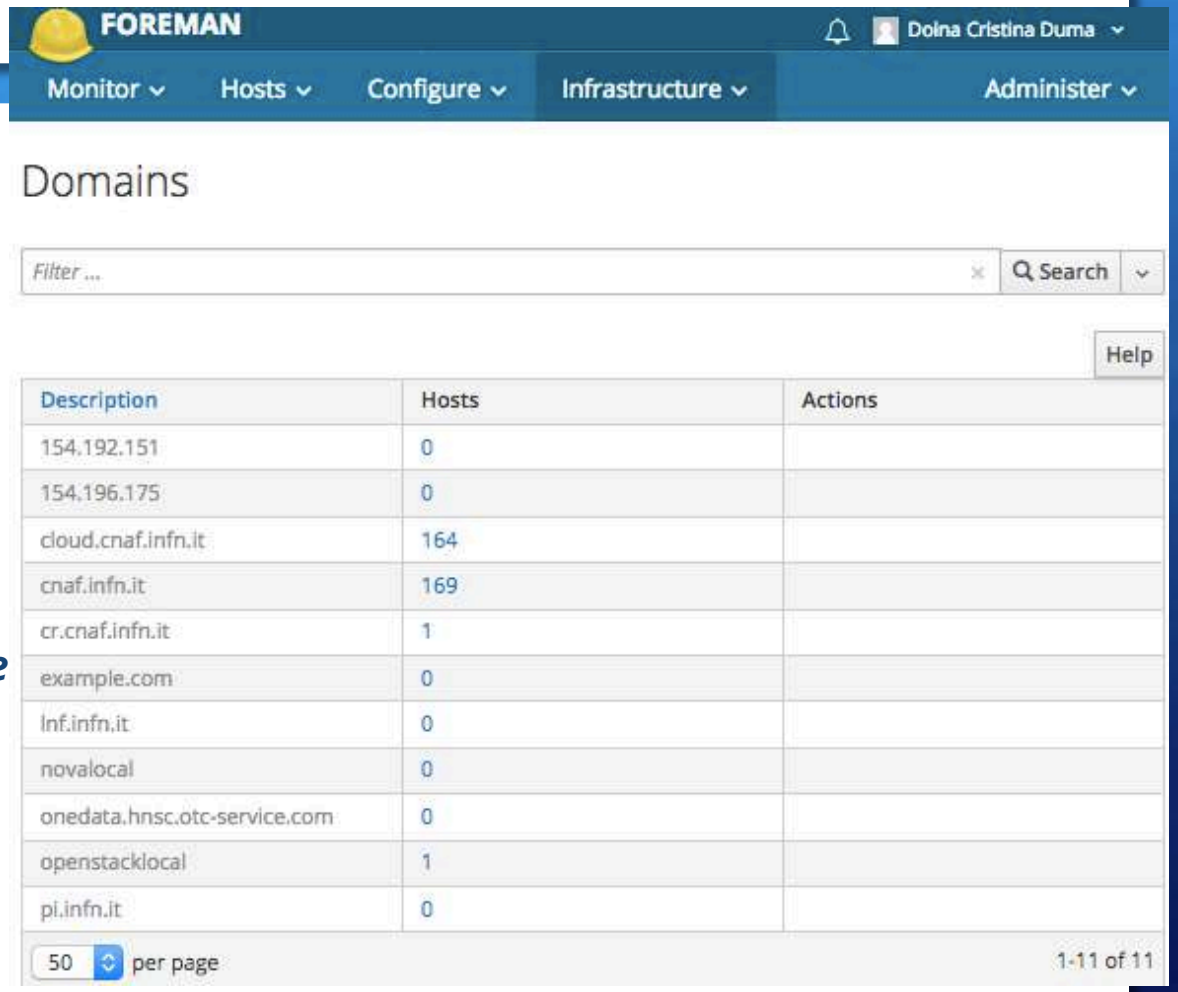

Monitor
Hosts
Configure
Infrastructure
Host Groups
Filter ...
Export
Create
Name
Hosts
Hosts including Sub-groups
Actions
Bebop/ELK/Dashboard/SDDS 0 0 Nest
SDDS 2 334 Nest
SDDS/Debian 20 20 Nest
SDDS-Dev 1 1 Nest
SDDS/RedHat 62 312 Nest
SDDS/RedHat/EMITestbed 0 0 Nest
SDDS/RedHat/GPFS-TB 3 3 Nest
SDDS/RedHat/Grid 11 45 Nest
SDDS/RedHat/Grid/MyProxy 2 2 Nest
SDDS/RedHat/Grid/SiteBDII 2 2 Nest


Monitor
Hosts
Configure
Infrastructure
Administer
Edit SDDS/RedHat
Host Group
Puppet Classes
Network
Operating System
Parameters
Parent
SDDS
Name *
RedHat
Description
Environment
Inherit parent (SDDS)
Compute profile
Inherit parent (no value)
Puppet Master
Inherit parent (Puppet)
Use this puppet server as an initial Puppet Server or to execute puppet runs
Puppet CA
Inherit parent (Puppet CA)
Use this puppet server as a CA server


Monitor
Hosts
Configure
Infrastructure
Administer
Edit SDDS/RedHat
Host Group
Puppet Classes
Network
Operating System
Parameters
Included Config Groups
SDDS-redhat
Remove
SDDS-base
Available Config Groups
SDDS-debian
Add
Included Classes
cnprov_puppetrun
sdds_packages
sdds_security_checks
sdds_security_updates
sdds_sensu
sdds_yum
Inherited Classes from SDDS
ntp
puppet
resolv_conf
sdds_backuppc
sdds_fail2ban
sdds_motd
sdds_rundeck
sdds_ssh
stdlib

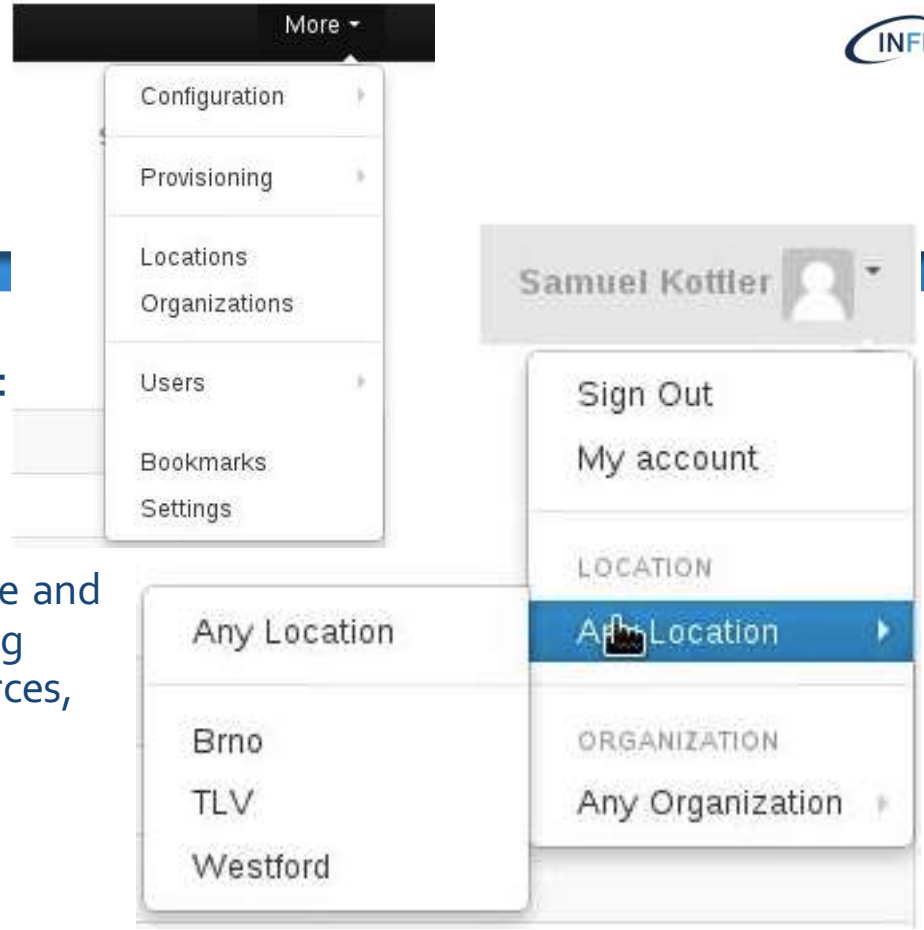
Structure

- Foreman provides different resources to organize hosts:
  - Hostgroup
  - Domains
    - "a domain and a DNS zone as the same thing"
    - *hostname*.**somewhere.com** o> domain is **somewhere.com**
    - allows Foreman to associate ***a puppet variable*** with a domain/site and ***automatically append*** this variable to ***all external node requests*** made by machines at that site
  - Environments
  - Organizations & Locations



Structure

- Foreman provides different resources to organize hosts:
  - ...
  - ***Organizations & Locations***
    - for managing different resources inside of a single instance and ***granting users specific access*** to those resources, including hosts, hostgroups, domains, subnets, and compute resources, environments
    - disabled by default
    - switch context



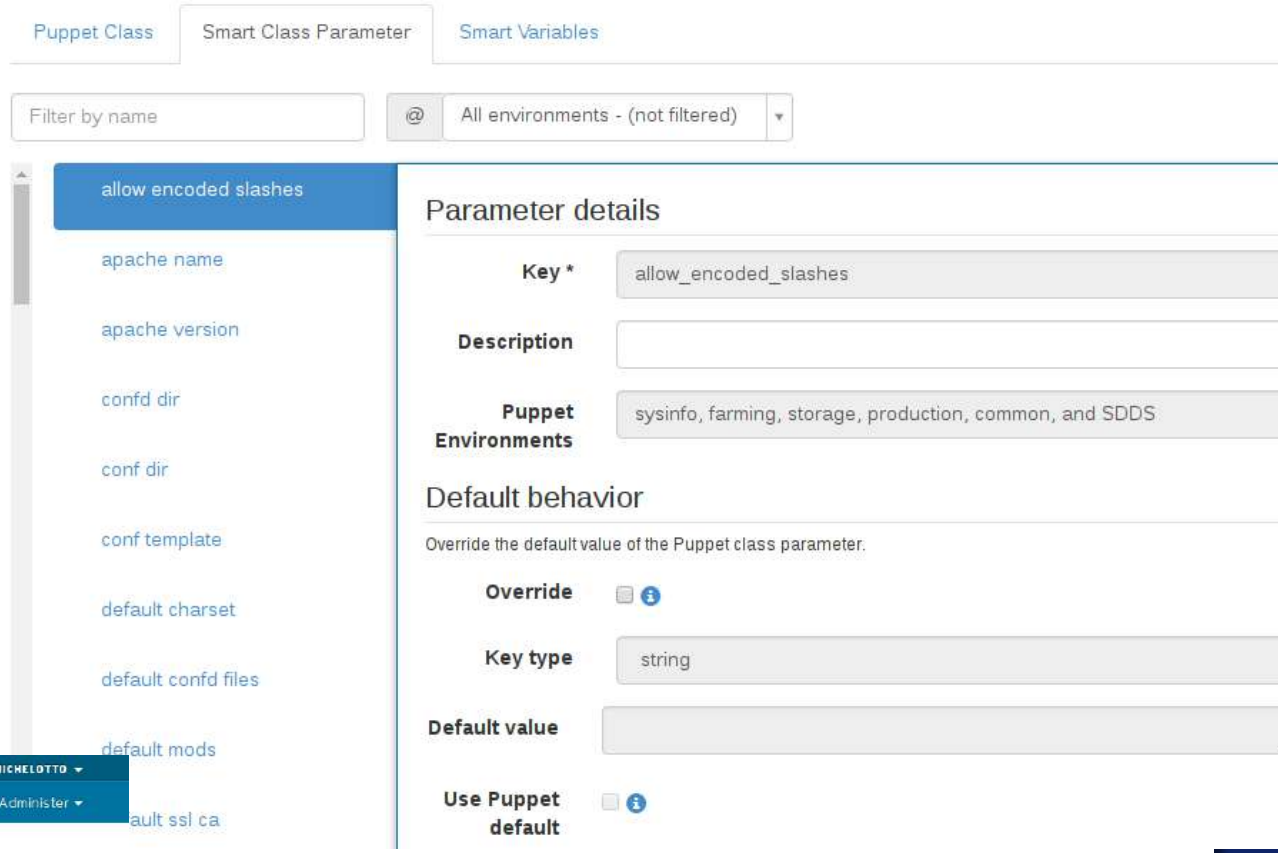
- **Parameter inheritance looks like this:**

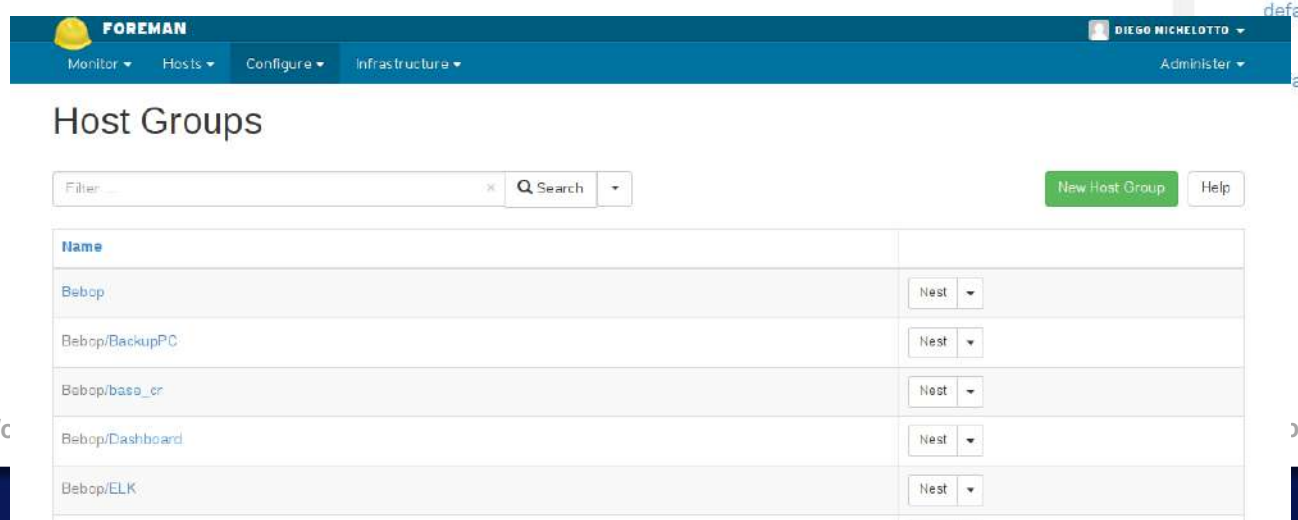
```
Environment
  -> Domains
       -> Hostgroup
            -> Host
```

Config. Management

„Define how a system should look like in an abstract way."

- Foreman provides ENC functionality
  - Simple UI to associate hosts with recipes.
  - Multiple Puppet environment support.
  - Allows grouping hosts.
  - Parameterized classes with built-in hierarchical data store.
  - Update multiple hosts at once.
- Supports mainly Puppet but extendable with plugins





Ansible

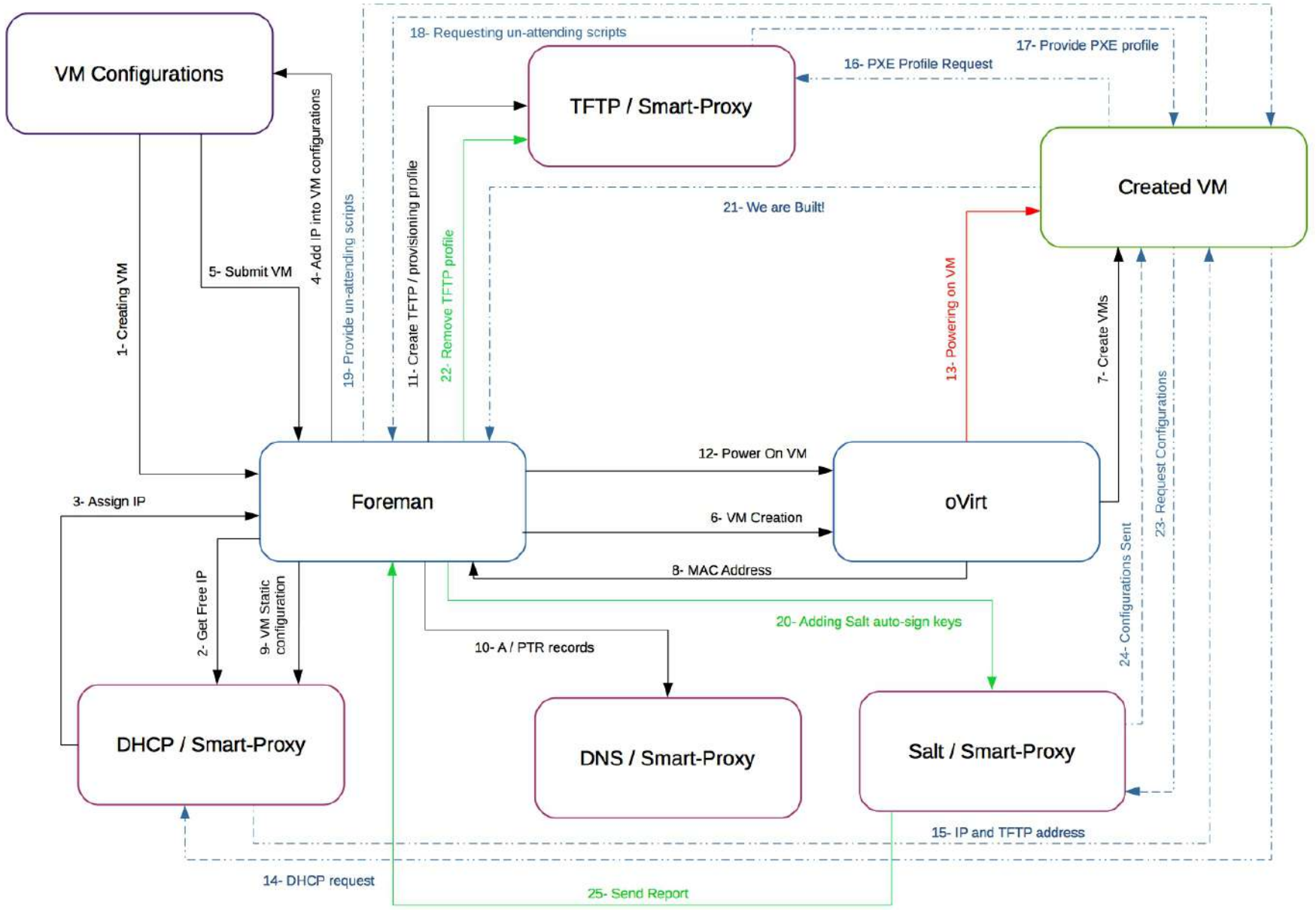
- Ansible plugin is still the new face in town
- Ansible provides dynamic Foreman inventory script
- Roles can be assigned to hosts and hostgroups
- Play roles through the GUI
- Import and delete roles through the GUI
- Doc: foreman_ansible plugin documentation.

The Foreman - Configuration

- Natively integrated with **Puppet**.
  - **Chef**, **Salt** and **Ansible** also available via plugin.
- Automatic registration & setup of clients, including auto-signing certs/keys.
- Defining:
  - **Classes / states**
  - **Parameters / pillars**
- Inventory data:
  - **Facts / Grains**
  - **Results of configuration runs**

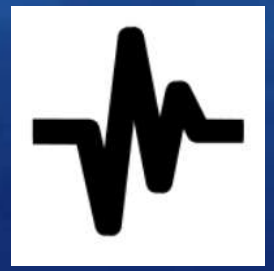
Workflow

- Compute resource (oVirt), also applicable to Libvirt and VMware
- Network (PXE) provisioning with DHCP and TFTP orchestration
- DNS orchestration
- Salt configuration management, also applicable to **Puppet**



Monitoring

- Collect and aggregate everything -

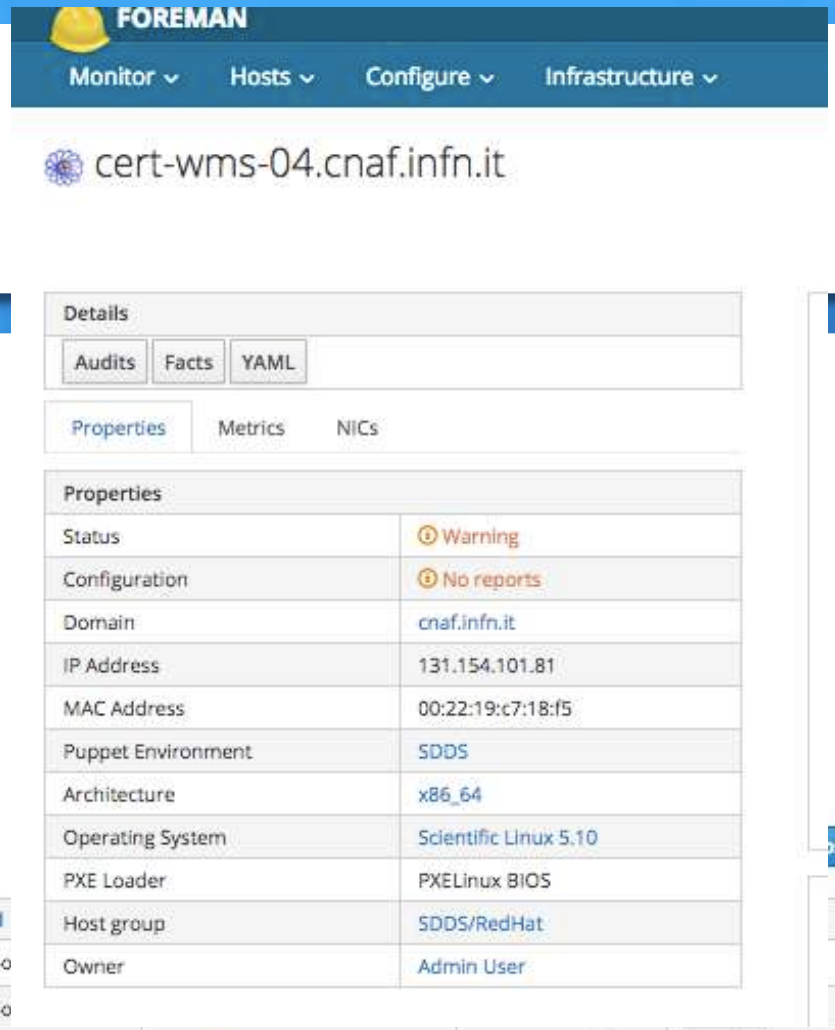


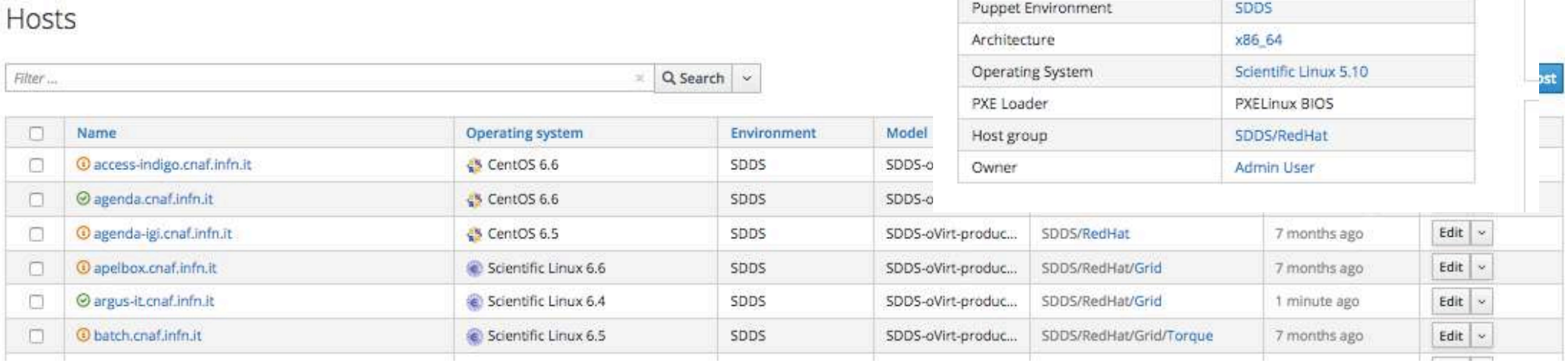
The Foreman - Monitoring

- Generic Report API with graphs/trends
  - System Inventories
  - Reports from runs
  - Generic reports: ABRT, OpenSCAP
- Context sensitive search:
  - Not full-text
  - Keyword completion
  - Works across whole application

Monitoring

- host statuses are useful
  - Global - **OK**, **Warning** or **Error**.
  - Sub-statuses





Reports

- Dashboard for all your Puppet hosts.
- Detailed log of puppet actions.
- Simple search through the log.
- Summary emails.
- Alerts through email and API.

FOREMAN PUPPET SUMMARY

Summary from 1 day ago to now

Summary report from Foreman server at https://prodest.cr.cnaf.infn.it

528	53	76
Changed	Out of sync	Disabled

Hosts with interesting values (changed, failures etc)

Hostname	Host group	Environment	applied	restarted	failed	failed_restarts	skipped	pending
access-indigo.cnaf.infn.it	SDDS/RedHat/Infra/INDIGO	SDDS	2	2	0	0	0	0
agenda-igi.cnaf.infn.it	SDDS/RedHat	SDDS	1	1	0	0	0	0
agenda.cnaf.infn.it	SDDS/RedHat	SDDS	1	1	0	0	0	0
				2	0	0	0	0
				1	0	0	0	0
				1	0	0	0	0
				0	0	0	0	0

FOREMAN PUPPET ERROR REPORT

Level	Resource	message
err	//farm-mon-ha01.cr.cnaf.infn.it/Puppet	Could not retrieve catalog from remote server: Error 400 on SERVER: Duplicate declaration: Class[Epel] is already declared; cannot redeclare at /etc/puppet/environments/farming/modules/farm_sensu/manifests/init.pp:90 on node farm-mon-ha01.cr.cnaf.infn.it
notice	//farm-mon-ha01.cr.cnaf.infn.it/Puppet	Using cached catalog

Reports

FOREMAN

DIEGO MICHELOTTO

Monitor | Hosts | Configure | Infrastructure | Administer

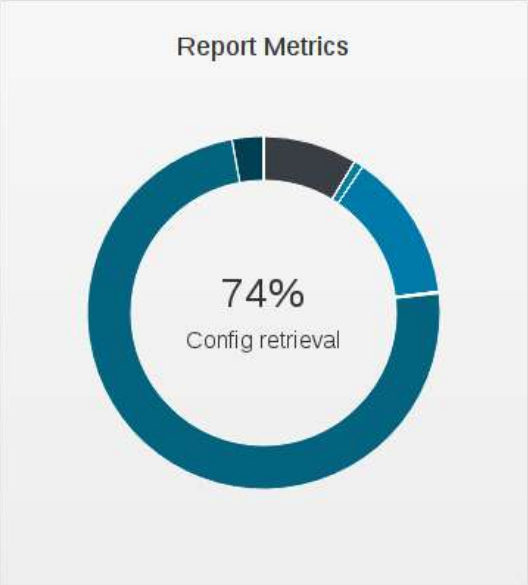
cnbebop-mon.cr.cnaf.infn.it

Show log messages: All messages

Back | Delete | Host details | Other reports for this host

Reported at 2016-11-21 16:04:02 UTC

Level	Resource	message
notice	//cnbebop-mon.cr.cnaf.infn.it//Stage[main]/Logrotate::Defaults::Redhat/Logrotate::Rule[wtmp]/File[/etc/logrotate.d/wtmp]/content	content changed '{md5}937918efc3352285c85d934b82916d9f' to '{md5}5027b375d1d10494905b4101cd322386'
notice	//cnbebop-mon.cr.cnaf.infn.it//Stage[main]/Logrotate::Defaults::Redhat/Logrotate::Rule[btmp]/File[/etc/logrotate.d/btmp]/content	content changed '{md5}ffe2335f6a971bc1f20b0a973fefbd43' to '{md5}b2d17cf3addcbba86a4075f792170eea'

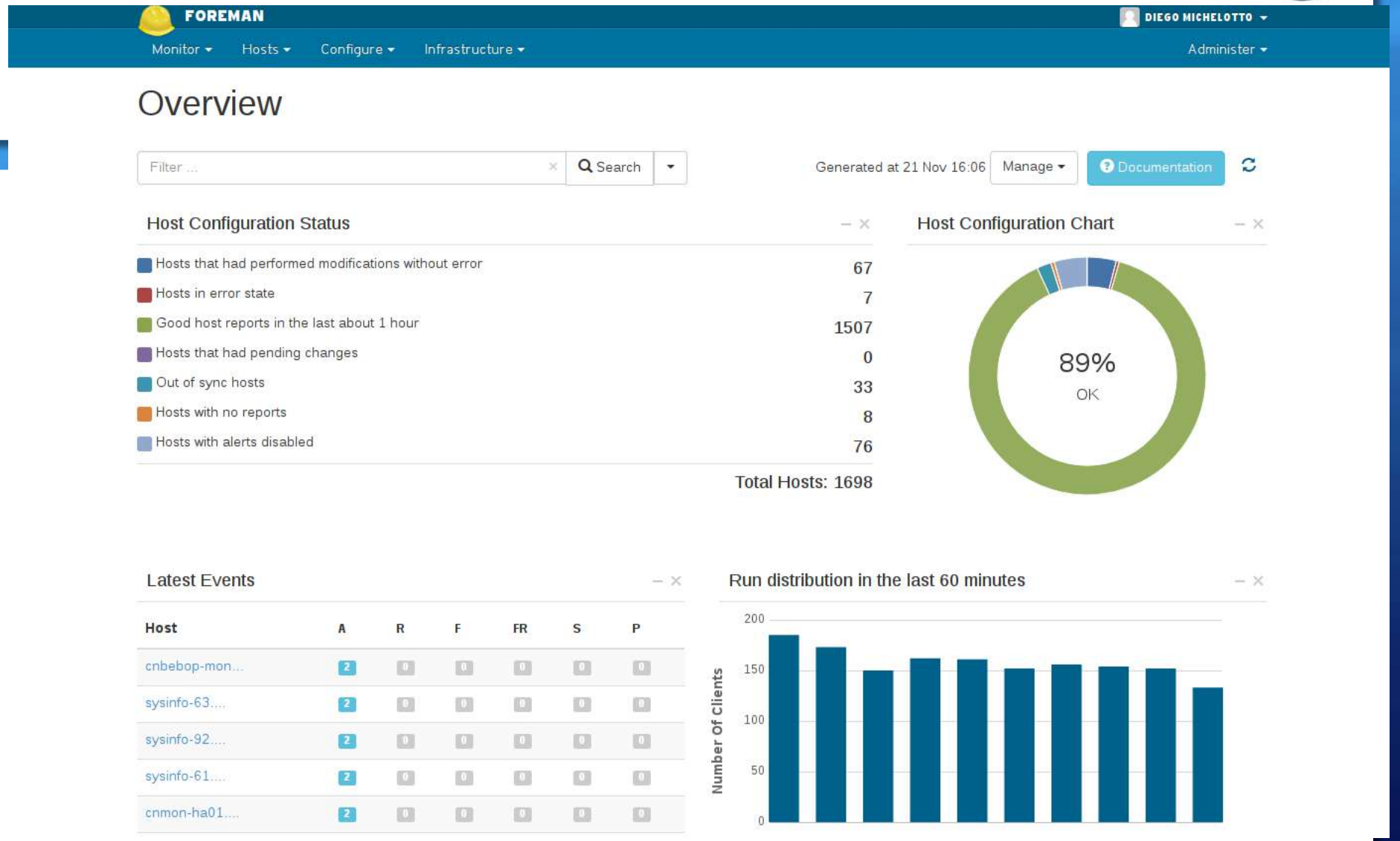


anchor	0.0013
concat_build	0.0024
config_retrieval	22.344
cron	0.0013
exec	0.2537
file	4.1074
package	2.5994
sensu_check	0.0433

Inventory

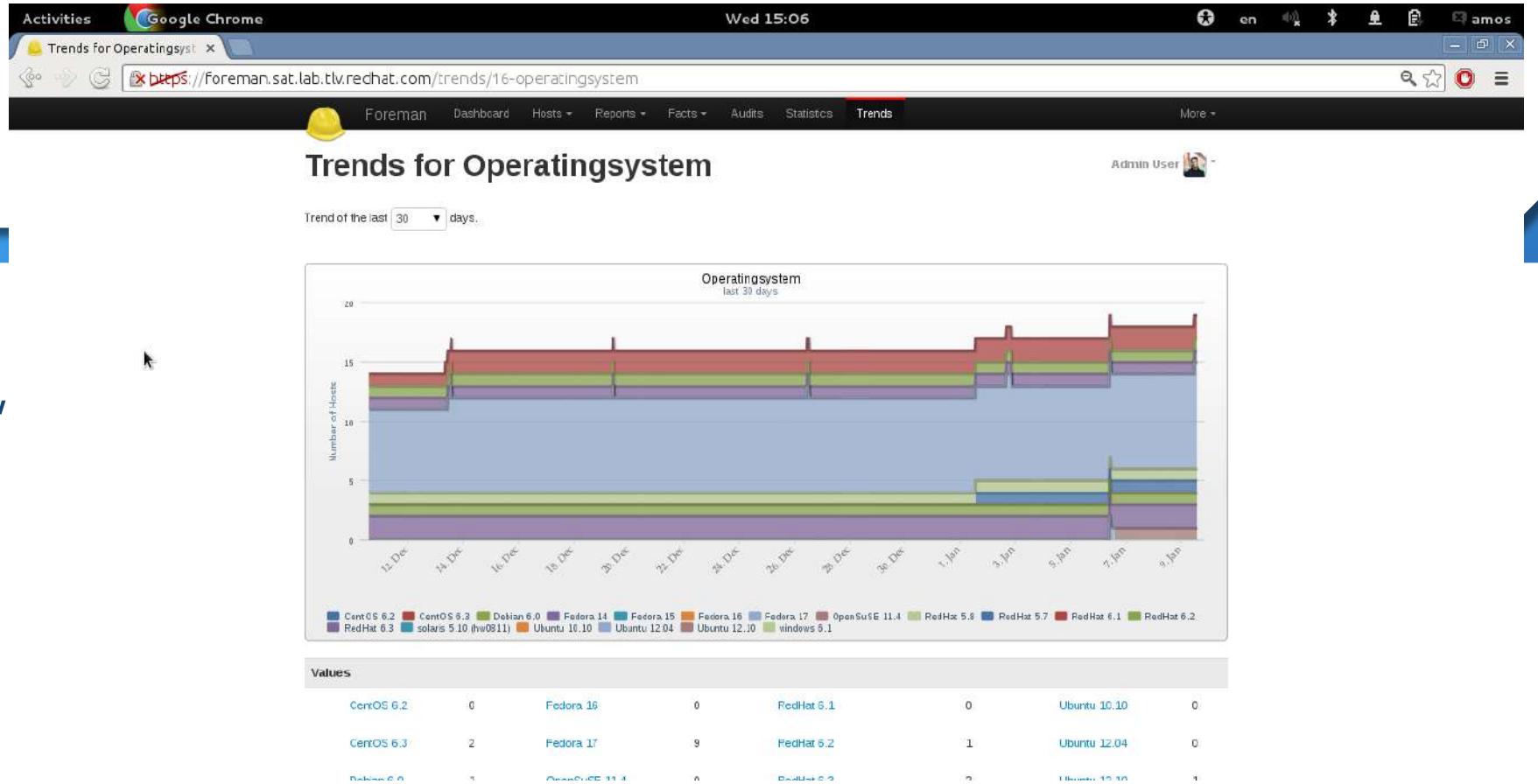
- Automatically collects your system inventory.
- Easy to browse and search through your inventory.
- Fact base permissions.
- Manage sets of host by fact values.
- Graphs!

Inventory


FOREMAN
DIEGO MICHELOTTO
Monitor
Hosts
Configure
Infrastructure
Administer
Overview
Filter ...
Search
Generated at 21 Nov 16:06
Manage
Documentation
Host Configuration Status
Hosts that had performed modifications without error 67
Hosts in error state 7
Good host reports in the last about 1 hour 1507
Hosts that had pending changes 0
Out of sync hosts 33
Hosts with no reports 8
Hosts with alerts disabled 76
Total Hosts: 1698
Host Configuration Chart
89%
OK
Latest Events
Host A R F FR S P
cnbebop-mon... 2 0 0 0 0 0
sysinfo-63.... 2 0 0 0 0 0
sysinfo-92.... 2 0 0 0 0 0
sysinfo-61.... 2 0 0 0 0 0
cnmon-ha01.... 2 0 0 0 0 0
Run distribution in the last 60 minutes
Number Of Clients
200
150
100
50
0

More data

- Audit log keeps track of all changes, very handy
- Trends give an overview of the infrastructure



Advanced Features

Plugins

- ***Katello*** -- Sync RPM (with snapshot), Docker, and Puppet content
- ***Discovery*** -- Metal-as-a-Service functionality for Foreman - Auto discovery and installation of bare metal
- ***Hooks*** -- Behavior extension of provisioning workflows
  - Triggered on actions: on action, do X
    - host create/update/delete, build complete, etc.
    - X could be anything
      - send an email
    - Can be shell, python, ruby, etc.
- ***Column_view*** -- GUI plugins
- ***Salt/Chef*** -- Integration with Salt/Chef
- Resources
- Many other plugins here.

Plugins

http://projects.theforeman.org/projects/foreman/wiki/List_of_Plugins

RESTful API

- Most of the UI actions are available in the API.
- Automatic Documentation.
- Current stable version is API V2.
- Powerful search API that can be used as an alternative to store-config.
- Used by foreman CLI and Remote-Admin.

Hammer CLI – Next Gen. CLI for Foreman

- A new pluggable CLI tool for Foreman and Katello - https://github.com/theforeman/hammer-cli
- Full CRUD, Uses APIV2
- Modeled on git with sub-commands
- In development, available since Foreman 1.3

Installation

- Manual installation
  - Install manually, repository, frontend, backend, database, smart-proxy, puppet and eventually dhcp, tftp and dns.
  - Configure each component manually
- Puppet installation
  - Use masterless puppet installation method in order to apply an manifest that use theforeman/foreman and theforeman/puppet modules.
  - Configure each component through an puppet manifest.
- ***Foreman-installer***
  - Automate installation of Foreman, Smart Proxy, Puppet master, Apache and Passanger.
  - Use Kafo - A new Ruby-gem that provides Puppet based installer.
    - It is in use since Foreman 1.3,
    - It has a CLI interface with progress bar, arguments for class-parameter, answer file and logging

The Foreman

- Foreman is a ***complete lifecycle management tool*** for physical and virtual servers.
- Give system administrators the power:
  - Easily ***automate*** repetitive tasks
  - Quickly ***deploy*** applications
  - Proactively ***manage*** servers
- Supported by ***huge community***
- Used by ***big institutes or companies***
  - CERN, Mozilla, Ericson, DHL, BBC, Citrix, Symantec
- Sponsored by RedHat, Rackspace

Hands-on !

Definitions

- **Configuration management** (CM)

  Wikipedia - *"field of management that focuses **on establishing and maintaining consistency** of a system's or product's performance and its functional and physical attributes with its requirements, design, and operational information throughout its life. For information assurance, CM can be defined as the **management of security features** and assurances through **control of changes made to hardware, software, firmware, documentation, test, test fixtures, and test documentation** throughout the life cycle of an information system"*

  PuppetLabs – "the process of **standardizing resource configurations** and **enforcing their state** across IT infrastructure in an automated yet agile manner. Configuration management is critical to the success of other IT processes, including provisioning, change management, release management, patch management, compliance and security."

- "Configuration management tools should probably be considered as an essential tool when moving into the cloud"

- From OpenStack:

  *"Maintaining an OpenStack cloud requires that you manage multiple physical servers, and this number might grow over time. Because **managing nodes manually is error-prone**, we strongly recommend that you **use a configuration management tool**. These tools automate the process of ensuring that all of your **nodes are configured properly** and encourage you to maintain your configuration information (such as packages and configuration options) in a version controlled repository"*