Systemkonfiguration mit Puppet

Benedikt Trefzer
benedikt.trefzer@cirrax.com

21.8.2022

$whoami

Benedikt Trefzer[1]

- work and live in Rubigen/Berne/Switzerland
- communicate in (swiss)german (pardon the helvetisms), english and french

Cirrax GmbH[2]

- Puppet trainings (from beginner to professional)
- Puppet consultancy and contract work
- Maintain some Opensource puppet modules[3]
- OpenStack cloud
- Linux and OpenSource consultancy and contract work

[1] mailto:benedikt.trefzer@cirrax.com

[2] https://cirrax.com

[3] https://forge.puppet.com/modules/cirrax

ⓒⓘⓢ

software for configuration management

Software for configuration management:

- puppet[4]
- ansible[5]
- CFEngine[6]
- chef[7]
- salt[8]

Also look at comparison of configuration management software on wikipedia[9]

[4] https://en.wikipedia.org/wiki/Puppet_(software)

[5] https://en.wikipedia.org/wiki/Ansible_(software)

[6] https://en.wikipedia.org/wiki/CFEngine

[7] https://en.wikipedia.org/wiki/Chef_(software)

[8] https://en.wikipedia.org/wiki/Salt_(software)

[9] https://en.wikipedia.org/wiki/Comparison_of_open-source_configuration_management_software

🅭🅯🄎

definition

- puppet manages the configuration of computers (called nodes)
- description of the desired state using Puppet's declarative language (and hiera data)
- this information is stored in files called "Puppet manifests".[10]

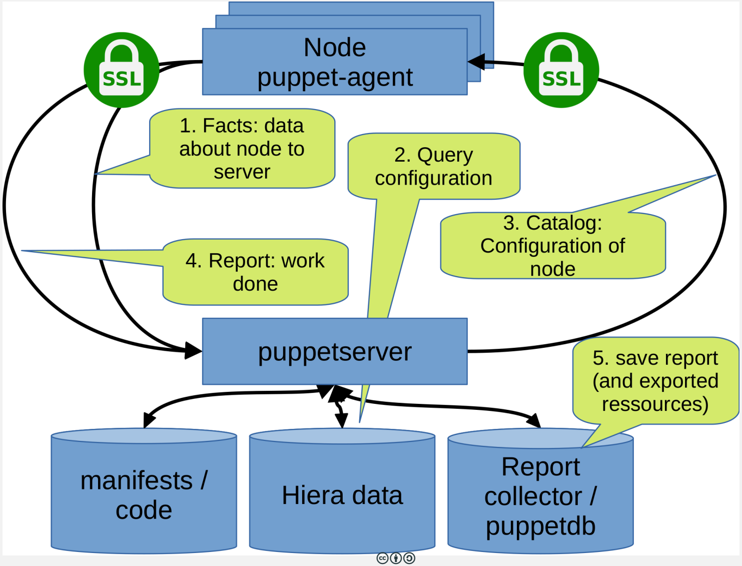
Steps during a puppet run (simplified):

1. discover the actual state of the target node (computer) (using facts)
2. compile the manifest into a system-specific catalog
3. transfer the catalog to the target system (node)
4. apply catalog on the node

[10] https://en.wikipedia.org/wiki/Puppet_(software)

cc by sa

Architecture


SSL
Node
puppet-agent
SSL
1. Facts: data about node to server
2. Query configuration
3. Catalog: Configuration of node
4. Report: work done
puppetserver
5. save report (and exported ressources)
manifests / code
Hiera data
Report collector / puppetdb

puppet declarative language

- the Puppet programming language is a declarative language that describes the state of a computer system in terms of "resources"
- the user assembles resources into manifests that describe the desired state of the system
- these manifests are stored on the puppetserver and compiled into configuration instructions for agents on request

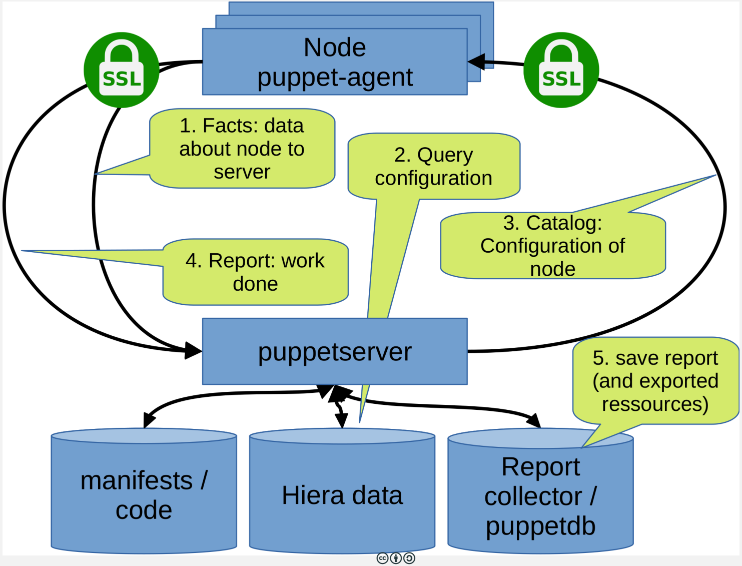
Example:

```
user { 'jbond':
  ensure  => present,
  comment => 'James bond',
  uid     => '1007',
  shell   => '/bin/bash',
  home    => '/home/jbond'
}
```

resource abstraction

- puppet allows to configure systems in a platform-agnostic way
- instead of specifying a system command to perform an action you:
    1. create a system-agnostic puppet resource
    2. puppet translates into system-specific instruction(s)
    3. puppet sends and executes them to the node to configure
- e.g. user creation can be declared with the same code for Windows and Unix systems
- the operation system specific implentation to use is called 'provider'

Architecture


Node
puppet-agent
SSL
SSL
1. Facts: data about node to server
2. Query configuration
3. Catalog: Configuration of node
4. Report: work done
puppetserver
5. save report (and exported ressources)
manifests / code
Hiera data
Report collector / puppetdb

🅭🅯🄎

Hiera[15]

hiera is key/value lookup tool. Data is organized in a hierarchy of several yaml (or json) files.

- separate code (structure) and data
- Hiera is now fully integrated into Puppet [11]
- eyaml[12] allows you to encrypt data you store in hiera
- several merge behaviours available[13]
- lookup_options configure how lookup is done and it's saved as a hiera data element per key
- command to manually query on puppetserver: `puppet lookup <KEY> --explain`[14]

[11] puppet >= 4.3 uses hiera 4, puppet >=4.9.3 uses hiera 5 with many new features

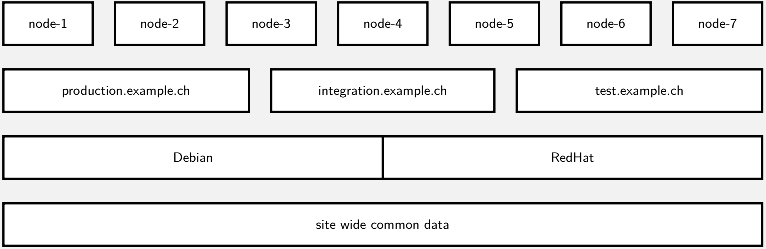
[12] puppet < 4: https://github.com/voxpupuli/hiera-eyaml,puppet >=4.9.3:https://puppet.com/docs/puppet/latest/hiera_config_yaml_5.html#configuring_a_hierarchy_level_hiera_eyaml

[13] https://puppet.com/docs/puppet/latest/hiera_merging.html#merge_behaviors

[14] https://puppet.com/docs/puppet/latest/hiera_automatic.html#using_puppet_lookup

[15] https://puppet.com/docs/puppet/latest/hiera_intro.html

hiera/lookup



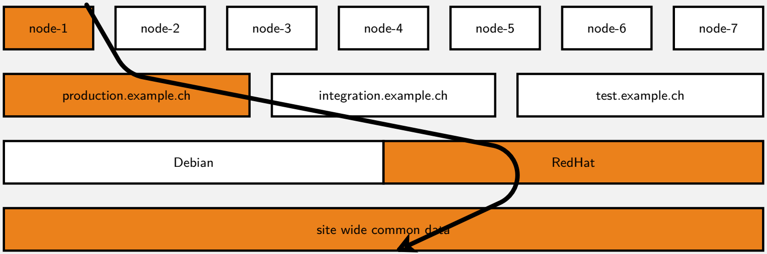
node-1 node-2 node-3 node-4 node-5 node-6 node-7

production.example.ch integration.example.ch test.example.ch

Debian RedHat

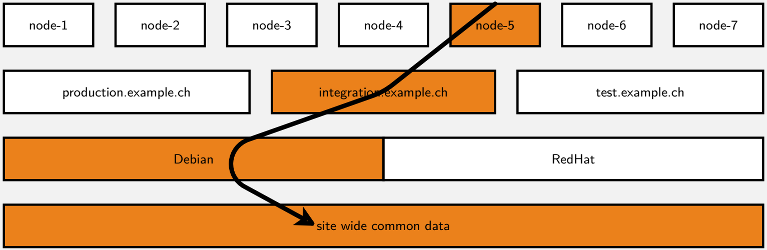
site wide common data

hiera/lookup



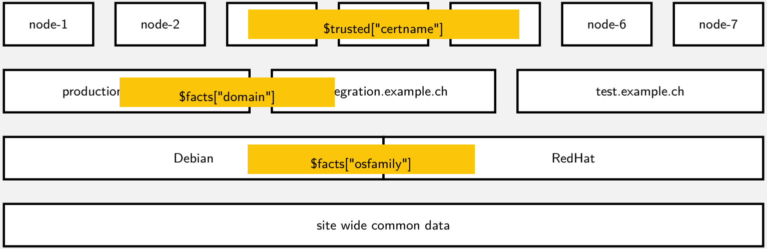
🅭🅯🄎

hiera/lookup



🅭🅯🄎

hiera/lookup



node-1
node-2
$trusted["certname"]
node-6
node-7

productio
$facts["domain"]
egration.example.ch
test.example.ch

Debian
$facts["osfamily"]
RedHat

site wide common data

```
hierarchy:
  - name: 'Per-node data'
    path: "nodes/%{trusted.certname}.yaml"
  - name: 'domain'
    path: "domain/%{::domain}.yaml"
  - name: 'OS'
    path: "osfamily/%{::osfamily}.yaml"
  - name: 'common'
    path: "common.yaml"
```

Hiera lookup examples

nodes

```
# node/node1.yaml
color: green
```

```
# node/node2.yaml
city: zurich
drinks:
  - coffee
  - tea
```

```
# node/node3.yaml
city: paris
country: france
```

```
# node/node4.yaml
city: hamburg
color: blue
```

osfamily

```
# osfamily/RedHat.yaml
city: bern
country: canada
```

```
# osfamily/Debian.yaml
country: switzerland
drinks:
  - beer
color: red
```

```
# osfamily/OpenBSD.yaml
song: Winter of 95
```

common

```
# common.yaml
city: berlin
country: switzerland
color: blue
drinks:
  - water
```

eyaml: overview

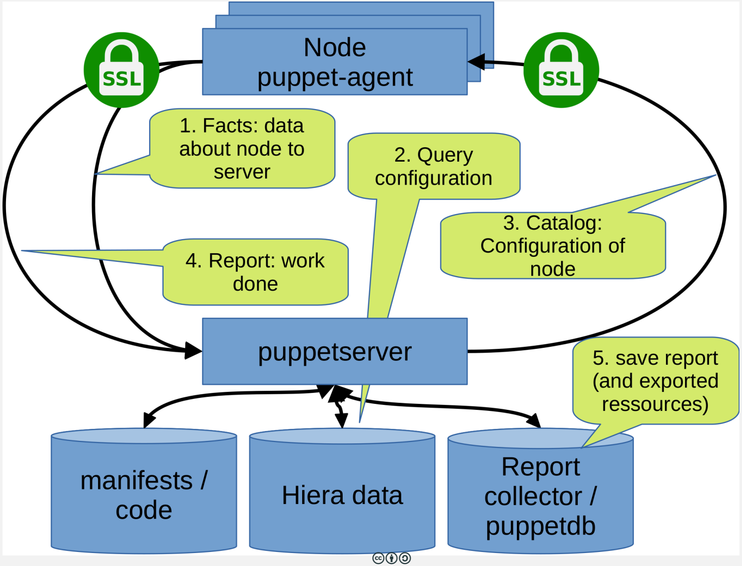
eyaml:[16] encrypt values in hiera YAML files.

- several encryption plugins available:
  - asymmetric encryption (PKCS#7) (default, same key for all developpers and server)
  - PGP available through plugin[17]
  - etc.
- Setup needs several steps:
  1. client setup to create an encrypted eyaml file
  2. puppetserver setup for decryption of eyaml files (libraries, keys)
  3. adapt hiera.yaml hierarchy for eyaml backend

[16] https://puppet.com/docs/puppet/latest/hiera_config_yaml_5.html#configuring_a_hierarchy_level_hiera_eyaml

[17] https://github.com/voxpupuli/hiera-eyaml-gpg

Architecture


Node
puppet-agent
SSL
SSL
1. Facts: data about node to server
2. Query configuration
3. Catalog: Configuration of node
4. Report: work done
puppetserver
5. save report (and exported ressources)
manifests / code
Hiera data
Report collector / puppetdb

🅭🅯🄎

PuppetDB

PuppetDB collects data generated by Puppet. It enables advanced Puppet features like exported resources.

- PuppetDB stores:
  - The most recent facts from every node
  - The most recent catalog for every node
  - Optionally, 14 days (configurable) of event reports for every node
- queried by the puppetserver (using puppetdb-termini)
- some performance patterns are available on http://localhost:8080[18]
- several dashboards[19] are available that also query puppetdb
- to install use the puppetdb[20] module

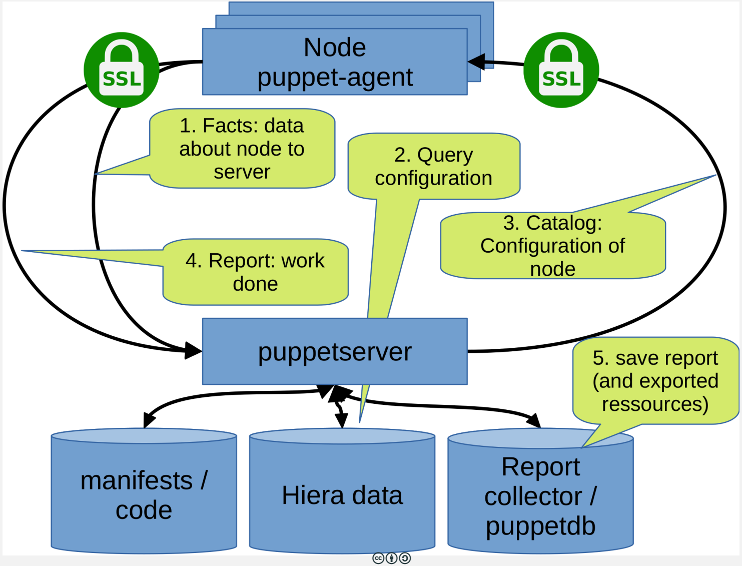
[18] hint: use ssh -L 8080:localhost:8080 root@YOUR_VM_IP to access with client

[19] e.g. https://github.com/dalen/puppetexplorer or https://github.com/voxpupuli/puppetboard or https://github.com/gillarkod/panopuppet(unmaintained)

[20] https://forge.puppet.com/puppetlabs/puppetdb

🅭🅯🄎

Architecture


Node
puppet-agent
SSL
SSL
1. Facts: data about node to server
2. Query configuration
3. Catalog: Configuration of node
4. Report: work done
puppetserver
5. save report (and exported ressources)
manifests / code
Hiera data
Report collector / puppetdb

modules

Modules are self-contained bundles of code and data.

- nearly all Puppet manifests belong in modules.
- a module consists mainly of[21]:
  - manifests (classes, defines etc)
  - hiera layer for data
  - templates
  - static files for download by a node
  - tests
- naming of directories is well defined (e.g. `templates` directory for templates, `manifests` for puppet code !)
- allowed module names must match [a-z][a-z0-9_]*
  (and not a reserved word[22])
- modules can be downloaded or written by you

[21] for the full module structure, see: `https://puppet.com/docs/puppet/latest/modules_fundamentals.html#module_structure`

[22] for reserved words see: `https://docs.puppet.com/puppet/latest/lang_reserved.html`

Howto install modules

- just copy into the file structure
- install from puppetforge (includes all dependencies): example:
  `puppet module install puppetlabs-stdlib`
- use git (e.g. with submodules)
- use special software (e.g. r10k[23])

Where to find modules:

- puppetforge[24] from puppetlabs
- github

[23] https://github.com/puppetlabs/r10k

[24] https://forge.puppet.com/

🅭🅯🄎

r10k: About

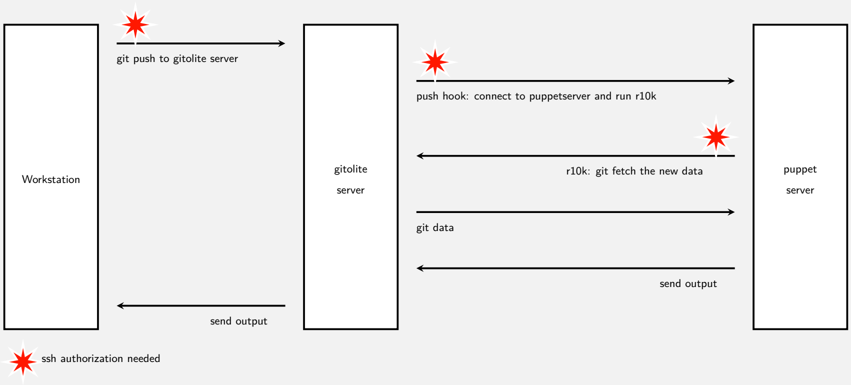
R10k[25] provides a general purpose toolset for deploying Puppet environments and modules. It implements the Puppetfile[26] format and provides a native implementation of Puppet environments.

- checkout each git branch into one puppet environment
- Puppetfile configures module versions to use per environment
- r10k ensures correct module and version per environment
- Modules can be defined from Puppet Forge, git repo, svn, tarball

[25] https://github.com/puppetlabs/r10k

[26] https://github.com/puppetlabs/r10k/blob/main/doc/puppetfile.mkd

r10k: example Workflow with gitolite and r10k


Workstation
git push to gitolite server
send output
gitolite server
push hook: connect to puppetserver and run r10k
r10k: git fetch the new data
git data
send output
puppet server
ssh authorization needed

use cirrax-r10k[27] and cirrax-gitolite[28] modules to implement

[27] https://forge.puppet.com/cirrax/r10k

[28] https://forge.puppet.com/cirrax/gitolite

🅭🅯🄎

Puppet Development Kit[30] (PDK)

PDK provides integrated testing tools and a command line interface to help you develop, validate, and test modules.

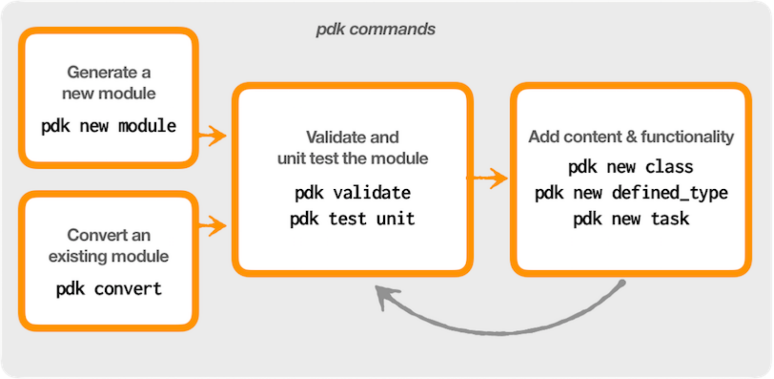
- sort of puppetlabs best practice
- in puppetforge compliant modules are marked with `PDK`
- simplify creation of new modules/classes/defines by adding basic tests etc.
- existing modules can be converted to make them compatible with PDK.
- add puppetlabs apt repository and use `apt install pdk` to install[29]
- includes it's own ruby environment which contains all libraries needed to run spec tests.

[29] https://puppet.com/docs/pdk/latest/pdk_install.html

[30] https://puppet.com/docs/pdk/latest/pdk.html

🅭🅯🄎

PDK: commands



pdk commands

Generate a new module

`pdk new module`

Convert an existing module

`pdk convert`

Validate and unit test the module

```
pdk validate
pdk test unit
```

Add content & functionality

```
pdk new class
pdk new defined_type
pdk new task
```

(from: https://puppet.com/docs/pdk/1.x/pdk_overview.html)

use `pdk bundle exec rake ...`

to run other commands (e.g. blacksmith[31], generation of module documentation etc)

[31] https://github.com/voxpupuli/puppet-blacksmith

Why should you use puppet ?

- **Consistency**: equal configuration on each node per profile/software
- **Automation**: eg. new dns resolver, time server etc
- **Documentation**: manifests/hiera in git and you know what you have changed at a certain time
- **Continuous Integration**: disallows manual configuration (will be overwritten)
- **On place for config**: new webhost also configures DB, DNS, backup, monitoring ...

but...

- **Initial work**: needs to be done, can be done step by step
- **Orchestration**: puppet is weak for node dependencies

🅭🅯🄎

Bolt[32]: Puppet open source orchestration tool

Bolt automates the manual work it takes to maintain your infrastructure. Use Bolt to automate tasks that you perform on an as-needed basis or as part of a greater orchestration workflow.

- connect to remote target via SSH (no agent needed)
- initiate commands and tasks to run on x nodes
- tasks are commands/scripts with metadata added (parameters etc.)
- create plans to orchestrate tasks on multiple nodes
- add plans/tasks to any puppet module

Use cases:

- Query remote node(s) for information/status
- do migrations
- ensure node dependency on installations

[32] https://puppet.com/docs/bolt/latest/bolt.html

🅭🅯🄎

$whoami

Benedikt Trefzer[33]

- work and live in Rubigen/Berne/Switzerland
- communicate in (swiss)german (pardon the helvetisms), english and french

Cirrax GmbH[34]

- Puppet trainings (from beginner to professional)
- Puppet consultancy and contract work
- Maintain some Opensource puppet modules[35]
- OpenStack cloud
- Linux and OpenSource consultancy and contract work

[33] mailto:benedikt.trefzer@cirrax.com

[34] https://cirrax.com

[35] https://forge.puppet.com/modules/cirrax