Software Configuration Management

Johannes Jasper, Thomas Werkmeister

Software Configuration Management

- abstract description of your infrastructure
  - packages dependencies
  - services
  - files
  - settings...
- declarative

  $ apt-get install nginx

  nginx:

      pkg.installed

Vagrant

- Managing and sharing VMs for development teams
- SCM to provision the VM

Benefits of Software Configuration Management

Reproducible Environments

- Working recipes can be rerun else where
  - identical VMs
  - automated
- Less error prone than manual processes

OS agnostic

- Resource Abstraction Layer (RAL)
- No need to know yum, apt, pacman…
  - pkg.installed

Infrastructure as code

- ‘executable documentation’
- Status of infrastructure is declared
  - no need to retrieve it by issuing commands
- Manageable with version control systems

Idempotence

- Recipes check if described state is already achieved
  - only run if not
- Can be safely run multiple times

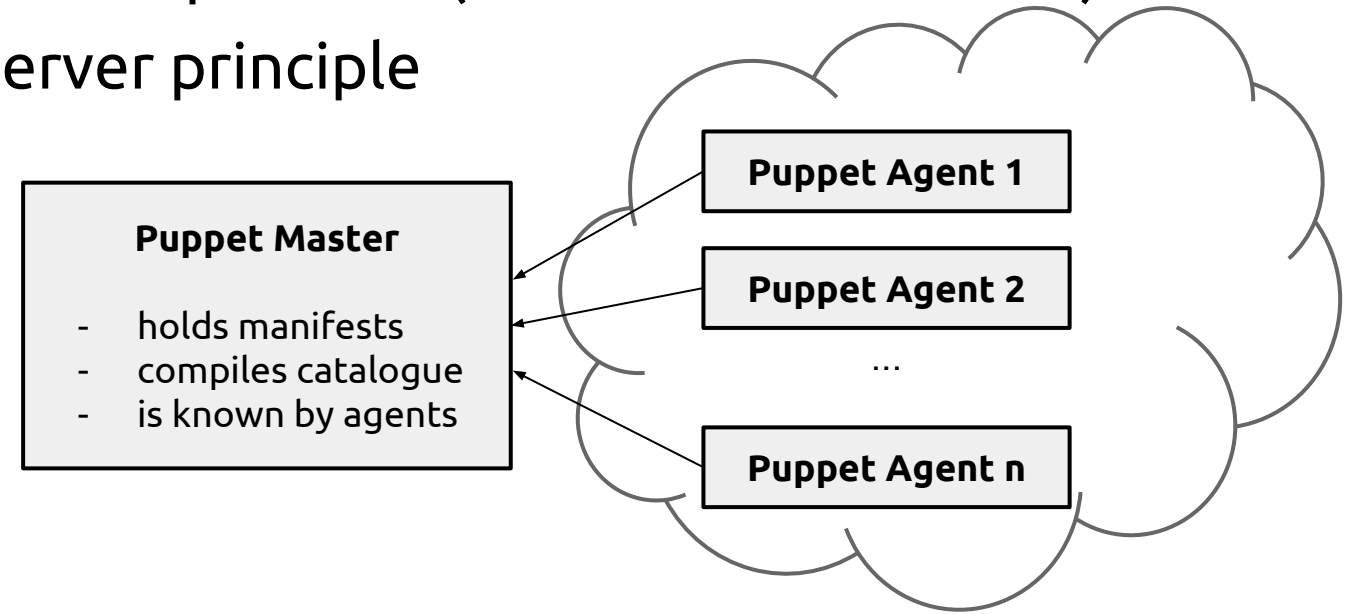
Verified State

- verify that the planned state is achieved after execution

Implementations

Puppet

- open source (Puppet Enterprise)
- platform independent (Master has to be *nix)
- client/server principle



Puppet Script

- Ruby based DSL to define resources

```
resource_type { 'resource_name'
  attribute => value
  ...
}
```

```
package { 'apache2':
  ensure => present,
}
```

```
user { "dave":
  ensure => present,
  uid => '507',
  gid => 'admin',
  shell => '/bin/zsh',
  home => '/home/dave',
  managehome => true,
}
```

Puppet Facter

```
root@puppetagent:~# facter
architecture => amd64
augeasversion => 1.2.0
bios_release_date => 12/03/2014
bios_vendor => Xen
bios_version => 4.2.amazon
blockdevices => xvda
domain => puppetexample.com
[...]
```

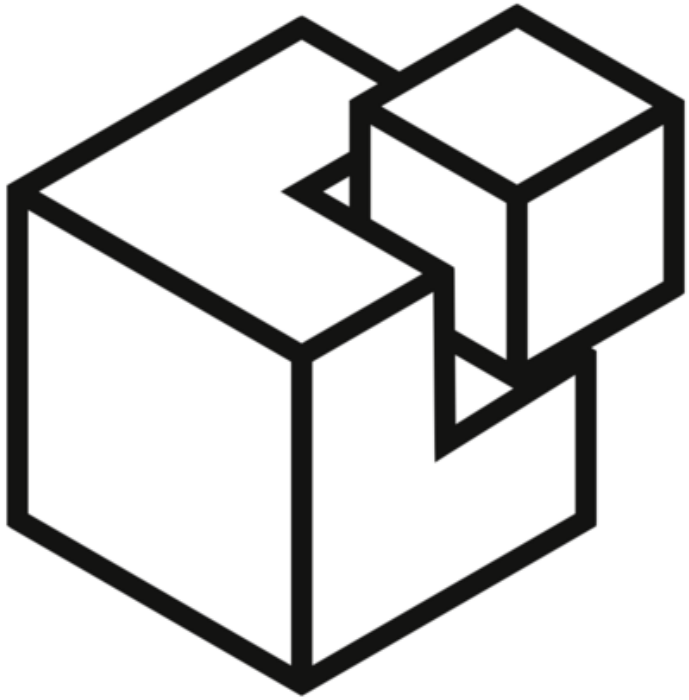
```
file {'/tmp/example-ip':
  ensure  => present,
  mode    => 0644,
  content => "private address: ${ipaddress_eth0}\n",
}
```

Live Demo

```
node 'puppetagent.puppetexample.com' {
  package { 'git':
     ensure => present,
  }

  file {'/tmp/example-ip':
     ensure  => present,
     mode => 0644,
     content => "private address: ${ipaddress_eth0}.\n",
  }

  exec { 'Run a command':
     command => "echo ${hostname} > /tmp/example-output",
     path => [ "/usr/local/bin/", "/bin/" ],
  }
}
```

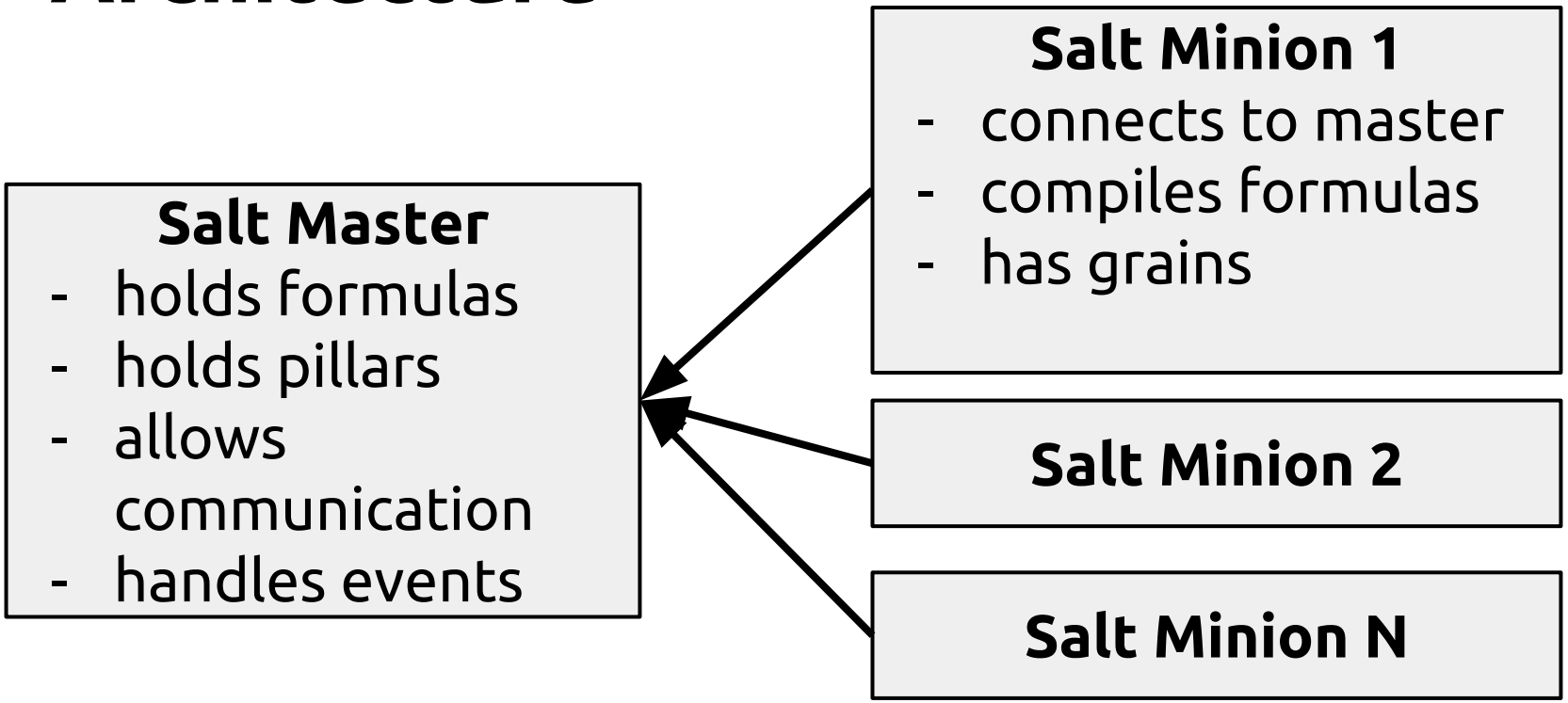
Docker Support

```
include 'docker'

docker::image { 'ubuntu': }
# sudo docker pull ubuntu

docker::run { 'helloworld':
  image   => 'ubuntu',
  command => '/bin/sh -c "while true; do echo hello world; sleep 1; done"',
}
# sudo docker run -d ubuntu /bin/sh -c "while true; do echo hello world; sleep 1;
done"
```



SALTSTACK

Architecture



Remote Execution vs SCM

```
salt ‘web*’ pkg.install nginx
```

```
nginx:
    pkg.installed
```

Formulas

- YAML + Jinja2 templating

OR

- pydsl
- pure python

```
screen:
  pkg:
    - latest

{% if grains["os"] == "Debian" %}
/etc/screenrc:
  file.managed:
    - source: salt://screen/screenrc
{% endif %}
```

Salt grains

nodename:
    ip-172-31-29-135
num_cpus:
    1
os:
    Ubuntu
role:
    memehost

```
user nginx;
worker_processes {{grains["num_cpus"]}};

error_log  /var/log/nginx/error.log warn;
pid    /var/run/nginx.pid;

....
```

Salt pillar

- information store
- can be used by multiple formulas
- can be made available to specific hosts only
- secrets
  - passwords
  - api keys

Docker support

- existing, functionality not complete
- can pull/push/built images and containers
- manipulating running docker instances still experimental

Live Demo