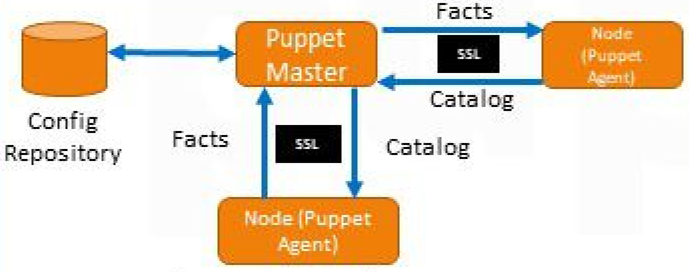
Puppet
CHEAT SHEET

Puppet

It is an open source configuration framework which is used when a lot of machines need to be configured in a similar form, or there is an infrastructure that requires dynamic scaling up or down with pre-determined config, or to have control over all the config machines so that a centralized change gets propagated.

Architecture

- Puppet Master: The node which controls the flow and has the authority
- Catalog: It is a document which describes the state of resources on a node which is managed by Puppet
- Report: The actions and infrastructure applied by a catalog during a Puppet run
- SSL Secure encryption on all data transport



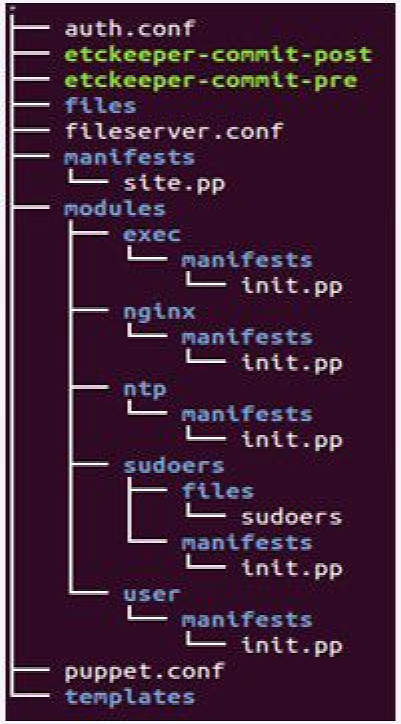
Client-Server Architecture

Files

- Attributes:
  - Ensure- if the file exists or not, what it should be.
  - Normal files- source of file, desired contents as a string
  - Recursively manage files and delete unmanaged files
  - Symlinks- symlink target
  - Others- backup, checksum, force, ignore, links, replace
- Package:
  - Manage software packages
    - Name-name of package
    - Ensure-if it should be installed or not
    - Present, Latest, Absent, Purged
    - Source- where to obtain the package
    - Provider- which packaging system to use
- Service:
  - Name- The name of the service to run
  - Ensure-status - running or stopped
  - Enable- if it should start on boot
  - Hasrestart- Use init script restart for stop+start
  - hasstatus- If to use the init script status command.

Classes and Modules

- Defining classes makes the class available by name but not by code
- Declaring a class evaluated the code inside too
  - class my_class{
    ... puppet code ...
    }
    include my_class
- Showing the implementation of a feature in a module, our main manifest can become smaller, more readable, and policy-focused.
- Module structure:



```
├── auth.conf
├── etckeeper-commit-post
├── etckeeper-commit-pre
├── files
├── fileserver.conf
├── manifests
│   └── site.pp
├── modules
│   ├── exec
│   │   └── manifests
│   │       └── init.pp
│   ├── nginx
│   │   └── manifests
│   │       └── init.pp
│   ├── ntp
│   │   └── manifests
│   │       └── init.pp
│   ├── sudoers
│   │   ├── files
│   │   │   └── sudoers
│   │   └── manifests
│   │       └── init.pp
│   └── user
│       └── manifests
│           └── init.pp
├── puppet.conf
└── templates
```

- A module is directory
- Module and directory name has to be same
- Contains manifest directory(contains .pp files)
- Should contain init.pp file

Advantages

- Enables to define the Infrastructure as Code (IAC) with easy coding. Consists version control, review, automated testing & delivery.
- Downtime is reduced
- Faster deployment times
- Automating repetitive tasks are easy.
- Supports a lot of platforms like windows, debian, BS

Puppet CLI

- Bootstrap client
  - puppet agent -t --server <puppet master> [<options>]
- Display facts
```
facter           # All system facts
facter -p        # All system and Puppet facts
facter -y        # YAML
facter -j        # JSON
facter [-p] <name>  # A specific fact
```
- Injecting facts
  - env FACTER_=puppet apply site.pp
- Find out effective classes on a node
  - cat /var/lib/puppet/classes.txt
- File modification date
  - cd /var/lib/puppet
```
for i in $(find clientbucket/ -name paths); do
  echo "$(stat -c %y $i | sed 's/\..*//')     $(cat $i)";
done | sort -n
```
- Disable to enable
```
puppet agent --disable
puppet agent --disable <info message>  # recent versions
puppet agent --enable\
```
- Managing certificates
```
puppet cert list
puppet cert list --all
puppet cert sign <name>
puppet cert clean <name>  # removes cert
```
- Managing nodes
```
puppet node clean <name>  # removes node + cert
```
- Managing modules
```
puppet module list
puppet module install <name>
puppet module uninstall <name>
puppet module upgrade <name>
puppet module search <name>
```
- Inspect resources and types
```
puppet describe -l
puppet resource <type name>
puppet kick <name>
puppet kick -p 5 <names>   #trigger puppet from master
```
- Debugging and deployment
```
puppetd --test # enable standard debugging options
puppetd --debug # enable full debugging
puppetd --one-time --detailed-exitcodes
```
- There are other commands as well

Other Commands

FUNCTION	COMMANDS
Check version	Puppet --version

File content manifest - site.pp	file { '/tmp/dafile':
	content => "datext\n",
	}

File content manifest- multiple nodes	node 'danodename' {
	file { '/tmp/dafile':
	content => "datext\n",
	}
	}

Remove packages	package { 'apache2.0-common':
	ensure => absent,
	}

Update packages	package { 'puppet':
	ensure => latest,
	}

Start service at boot time	service { 'nginx':
	ensure => running,
	enable => true, #false disables auto-startup
	}

Set to specific version	package { 'nginx':
	ensure => '1.1.18-1ubuntu0.1',
	}