Puppet Declarative config management

Facts facter on the node → sent to server → **Catalog** compiled from manifests + Hiera → agent fetches → **Agent** compares to real state → fixes drift → **Report** changes and failures

CONCEPTS

Puppet is declarative: you describe the end state and it only changes what differs, so runs are safe to repeat.

Always --noop first: it prints what would change without touching the system.

APPLY AND AGENT

puppet apply *<file.pp>*
Apply a manifest locally

puppet apply --noop *<file.pp>*
Dry run, show changes only

puppet apply -e '*<code>*'
Run a one-line snippet

puppet apply --test *<file.pp>*
Verbose, diffs, detailed exit codes

puppet apply --detailed-exitcodes *<file.pp>*
Exit code reports changes

puppet apply --modulepath *<dir>* *<file.pp>*
Use modules from a directory

APPLY AND AGENT (cont.)

puppet apply --debug *<file.pp>*
Full debug output

puppet agent -t
One foreground run, verbose

puppet agent -t --noop
Preview what the server would change

puppet agent --onetime --no-daemonize
Single run, stay in terminal

puppet agent -t --environment *<env>*
Run against another environment

puppet agent --server *<host>* **-t**
Point at a specific server

puppet agent --disable "*<reason>*"
Pause runs on this node

puppet agent --enable
Resume runs

INSPECT AND CHANGE

puppet resource *<type>*
List all resources of a type

INSPECT AND CHANGE (cont.)

puppet resource *<type>* *<name>*
Show one resource as code

puppet resource *<type>* *<name>* *<attr>*=*<value>*
Change a resource now

puppet describe *<type>*
Docs for a resource type

puppet help *<subcommand>*
Help for a subcommand

puppet --version
Print Puppet version

FACTS AND CONFIG

facter
Print all facts

facter *<fact>*
One fact, e.g. os.family

facter --json
All facts as JSON

puppet config print
All settings and values

puppet config print *<setting>*
One setting, e.g. modulepath

HIERA AND LOOKUP

puppet lookup *<key>*
Look up a data value

puppet lookup *<key>* **--explain**
Show where the value came from

puppet lookup *<key>* **--node** *<node>*
Look up as another node

MODULES

puppet module install *<author-name>*
Install from the Forge

puppet module install *<author-name>* **--version** *<ver>*
Pin a version

puppet module list
Installed modules

CHECK AND LINT

puppet parser validate *<file.pp>*
Syntax check only

puppet-lint *<file.pp>*
Style problems

puppet-lint --fix *<file.pp>*
Auto-fix what it can

MANIFEST SYNTAX

Resource

```
package { 'nginx':
  ensure => installed,
}
```

Service and file

```
service { 'nginx':
  ensure => running,
  enable => true,
}
file { '/etc/motd':
  ensure  => file,
  content => "Managed by
Puppet\n",
}
```

Class

```
class web (
  String $port = '80',
) {
  package { 'nginx': ensure
=> installed }
}
include web
```

Ordering

```
package { 'nginx': ensure =>
installed }
-> service { 'nginx': ensure
=> running }
# or: require =>
Package['nginx']
```

Notify on change

```
file { '/etc/nginx/
nginx.conf':
  source => 'puppet:///
modules/web/nginx.conf',
  notify =>
Service['nginx'],
}
```