Managing CoreOS with Puppet

What? Why? How?

Gareth Rushgrove
Puppet

@garethr

puppet

The shortest path to better software.

This talk

What we’ll cover

- What is configuration management?
- CoreOS and Config management?
- Running Puppet on CoreOS
- Useful super powers

I’m assuming some knowledge of CoreOS and of Puppet (or similar tools)

LIVE DEMOS

puppet

What is Configuration Management?

Useful background

- 1950s research
- 1960s 480 series
- 1991 MIL-HDBK-61
- 1998 ANSI-EIA-649

- Identification
- Control
- Status accounting
- Verification and audit

Military Handbook Configuration Management Guidance MIL-HDBK-61B

Configuration management verifies that a system is identified and documented in sufficient detail

National Consensus Standard for Configuration Management EIA-649

Configuration management verifies that a system performs as intended

National Consensus Standard for Configuration Management EIA-649

But CoreOS and Config Management?

The why

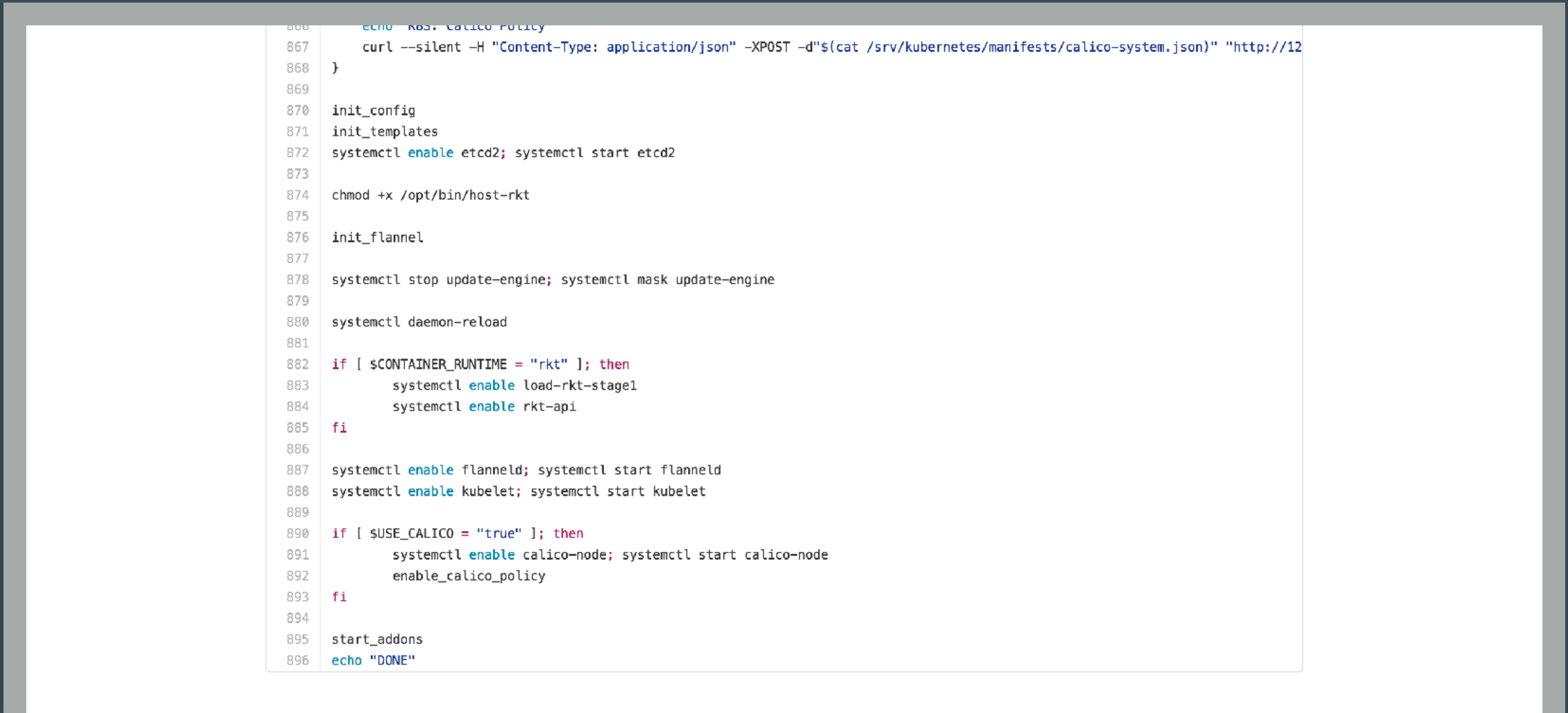
> “Fleet unit files tend toward chaos”

Gabriel Monroy, CTO, Dies and CoreOS contributor

> “Don't use cloud init for configuration management”

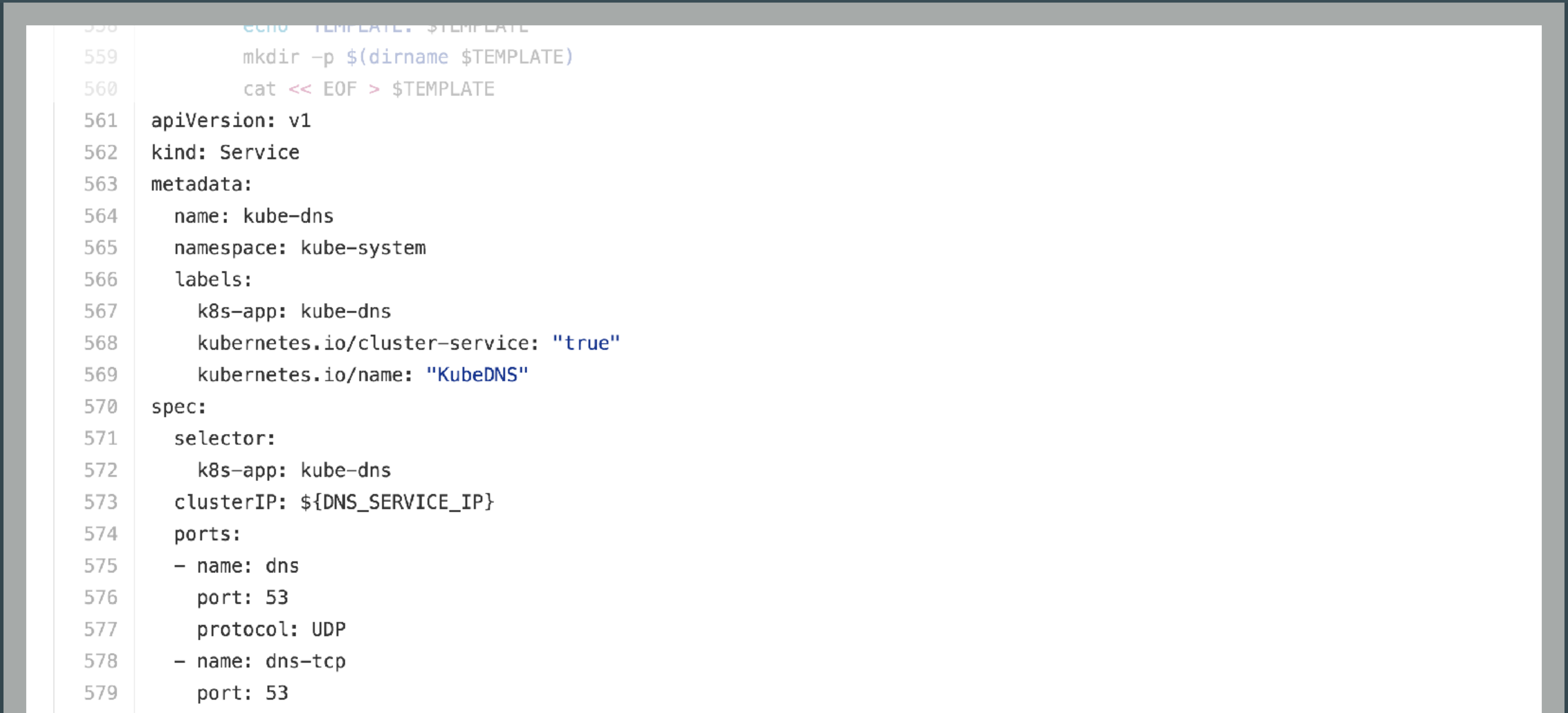
Gabriel Monroy, CTO, Dies and CoreOS contributor

900 line user data script!



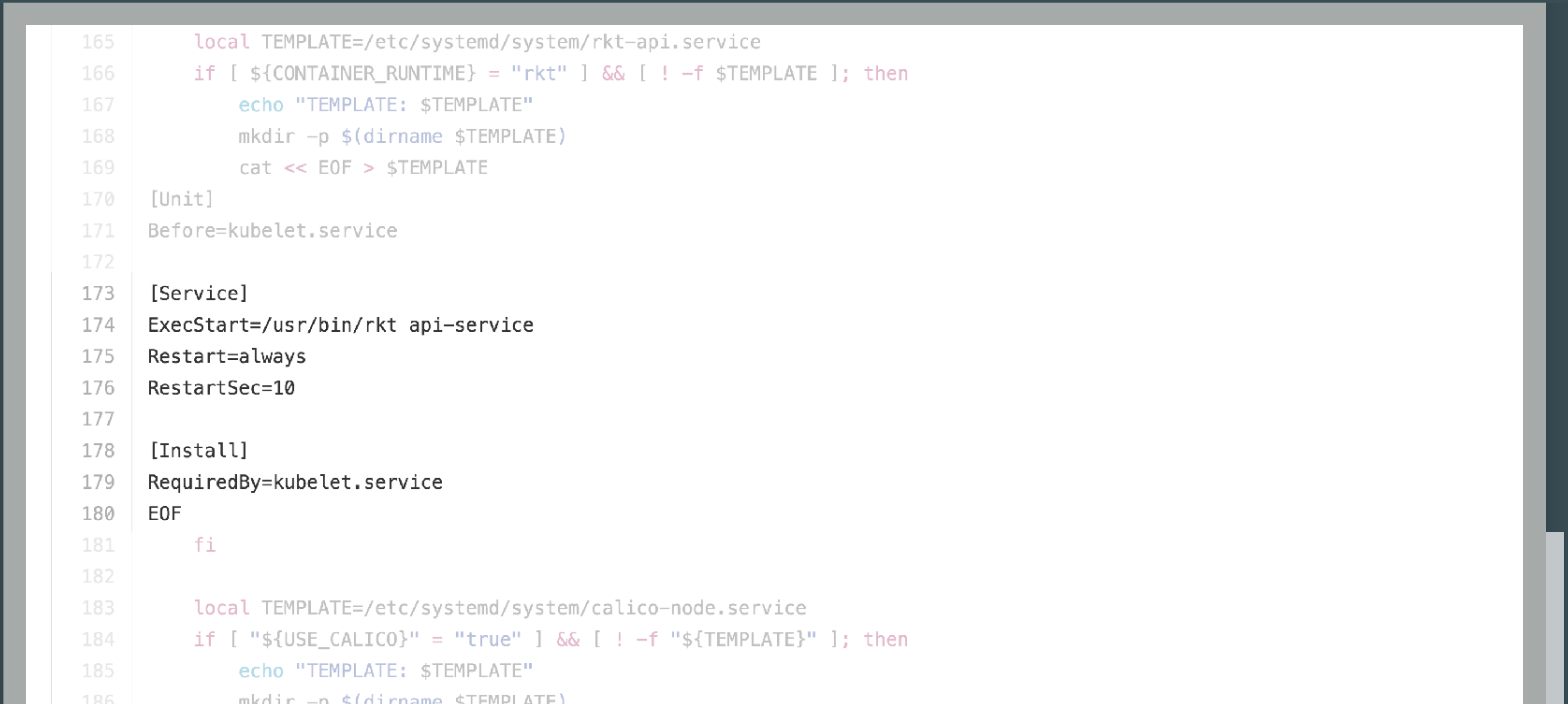
```
    echo "K8S: Calico Policy"
    curl --silent -H "Content-Type: application/json" -XPOST -d"$(cat /srv/kubernetes/manifests/calico-system.json)" "http://12
}

init_config
init_templates
systemctl enable etcd2; systemctl start etcd2

chmod +x /opt/bin/host-rkt

init_flannel

systemctl stop update-engine; systemctl mask update-engine

systemctl daemon-reload

if [ $CONTAINER_RUNTIME = "rkt" ]; then
        systemctl enable load-rkt-stage1
        systemctl enable rkt-api
fi

systemctl enable flanneld; systemctl start flanneld
systemctl enable kubelet; systemctl start kubelet

if [ $USE_CALICO = "true" ]; then
        systemctl enable calico-node; systemctl start calico-node
        enable_calico_policy
fi

start_addons
echo "DONE"
```

With embedded YAML



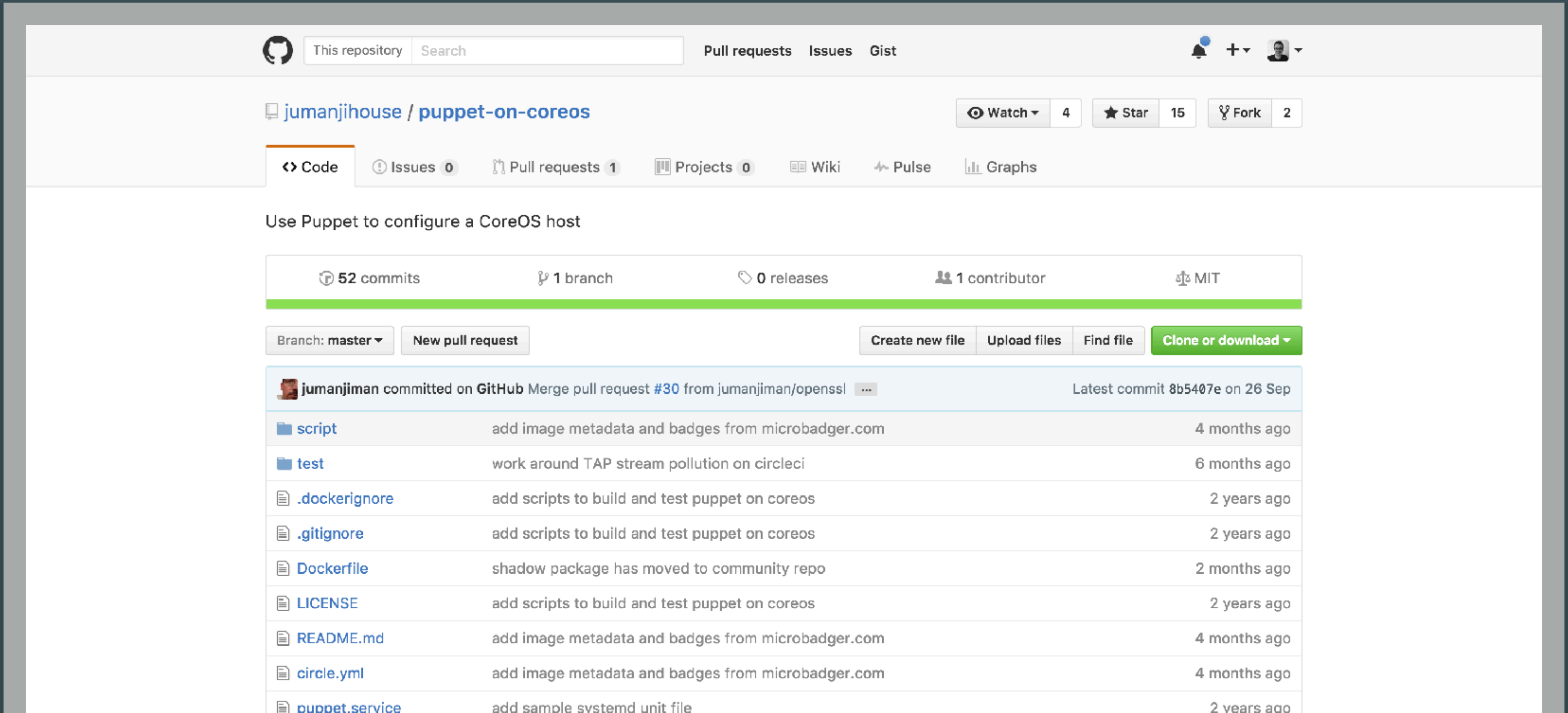
```
        echo "TEMPLATE: $TEMPLATE"
        mkdir -p $(dirname $TEMPLATE)
        cat << EOF > $TEMPLATE
apiVersion: v1
kind: Service
metadata:
  name: kube-dns
  namespace: kube-system
  labels:
    k8s-app: kube-dns
    kubernetes.io/cluster-service: "true"
    kubernetes.io/name: "KubeDNS"
spec:
  selector:
    k8s-app: kube-dns
  clusterIP: ${DNS_SERVICE_IP}
  ports:
  - name: dns
    port: 53
    protocol: UDP
  - name: dns-tcp
    port: 53
```

and systemd unit files



```
    local TEMPLATE=/etc/systemd/system/rkt-api.service
    if [ ${CONTAINER_RUNTIME} = "rkt" ] && [ ! -f $TEMPLATE ]; then
        echo "TEMPLATE: $TEMPLATE"
        mkdir -p $(dirname $TEMPLATE)
        cat << EOF > $TEMPLATE
[Unit]
Before=kubelet.service

[Service]
ExecStart=/usr/bin/rkt api-service
Restart=always
RestartSec=10

[Install]
RequiredBy=kubelet.service
EOF
    fi

    local TEMPLATE=/etc/systemd/system/calico-node.service
    if [ "${USE_CALICO}" = "true" ] && [ ! -f "${TEMPLATE}" ]; then
        echo "TEMPLATE: $TEMPLATE"
        mkdir -p $(dirname $TEMPLATE)
```

jumanjihouse/puppet-on-coreos



jumanjihouse / puppet-on-coreos

Use Puppet to configure a CoreOS host

52 commits · 1 branch · 0 releases · 1 contributor · MIT

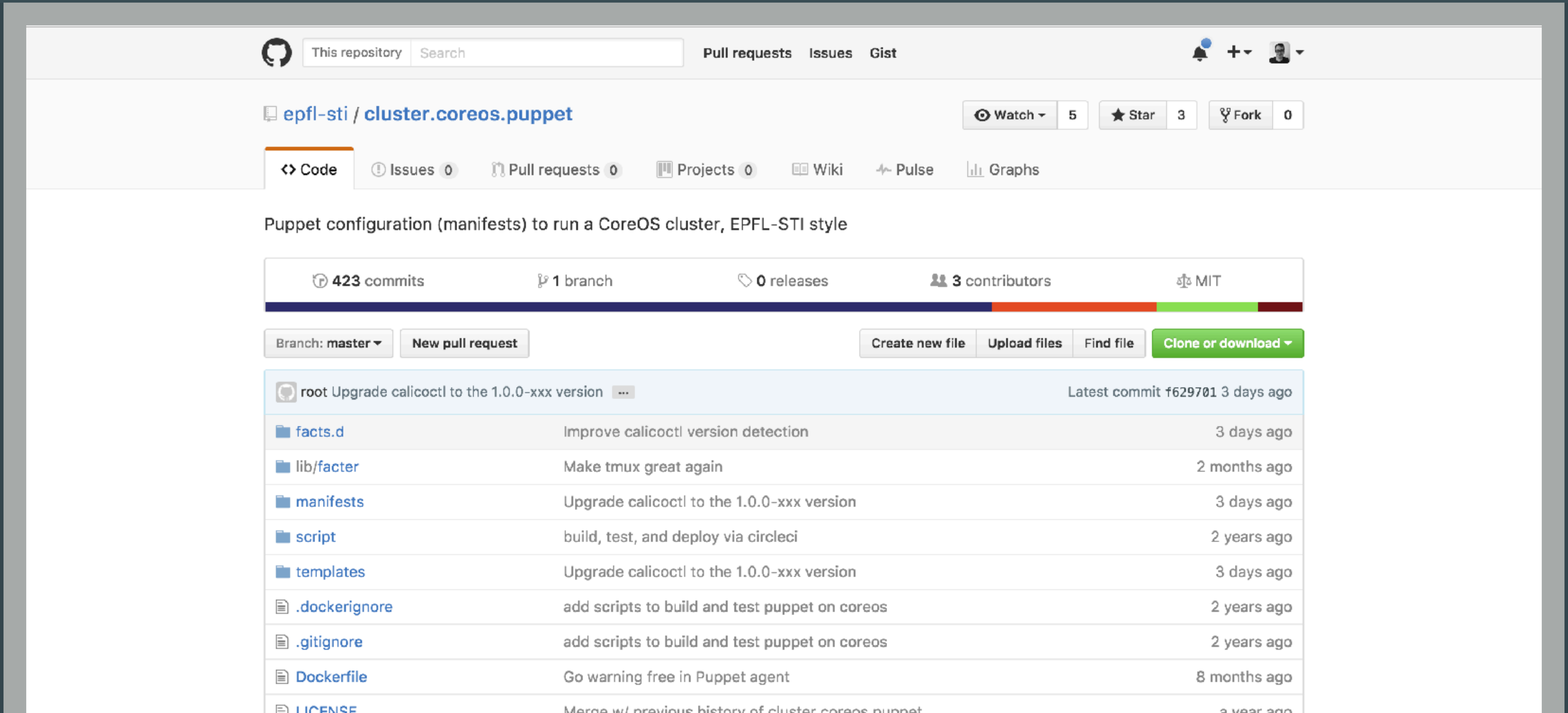
jumanjiman committed on GitHub Merge pull request #30 from jumanjiman/openssl … Latest commit 8b5407e on 26 Sep

script	add image metadata and badges from microbadger.com	4 months ago
test	work around TAP stream pollution on circleci	6 months ago
.dockerignore	add scripts to build and test puppet on coreos	2 years ago
.gitignore	add scripts to build and test puppet on coreos	2 years ago
Dockerfile	shadow package has moved to community repo	2 months ago
LICENSE	add scripts to build and test puppet on coreos	2 years ago
README.md	add image metadata and badges from microbadger.com	4 months ago
circle.yml	add image metadata and badges from microbadger.com	4 months ago
puppet.service	add sample systemd unit file	2 years ago

> “Cloud-init is fine for bootstrapping CoreOS, but sometimes you want to consolidate inventory data
> for all your hosts”

Paul Morgan, Architect, NYSE

École Polytechnique Fédérale de Lausanne



epfl-sti / cluster.coreos.puppet

Puppet configuration (manifests) to run a CoreOS cluster, EPFL-STI style

root Upgrade calicoctl to the 1.0.0-xxx version … Latest commit f629701 3 days ago

facts.d	Improve calicoctl version detection	3 days ago
lib/facter	Make tmux great again	2 months ago
manifests	Upgrade calicoctl to the 1.0.0-xxx version	3 days ago
script	build, test, and deploy via circleci	2 years ago
templates	Upgrade calicoctl to the 1.0.0-xxx version	3 days ago
.dockerignore	add scripts to build and test puppet on coreos	2 years ago
.gitignore	add scripts to build and test puppet on coreos	2 years ago
Dockerfile	Go warning free in Puppet agent	8 months ago
LICENSE	Merge w/ previous history of cluster.coreos.puppet	a year ago

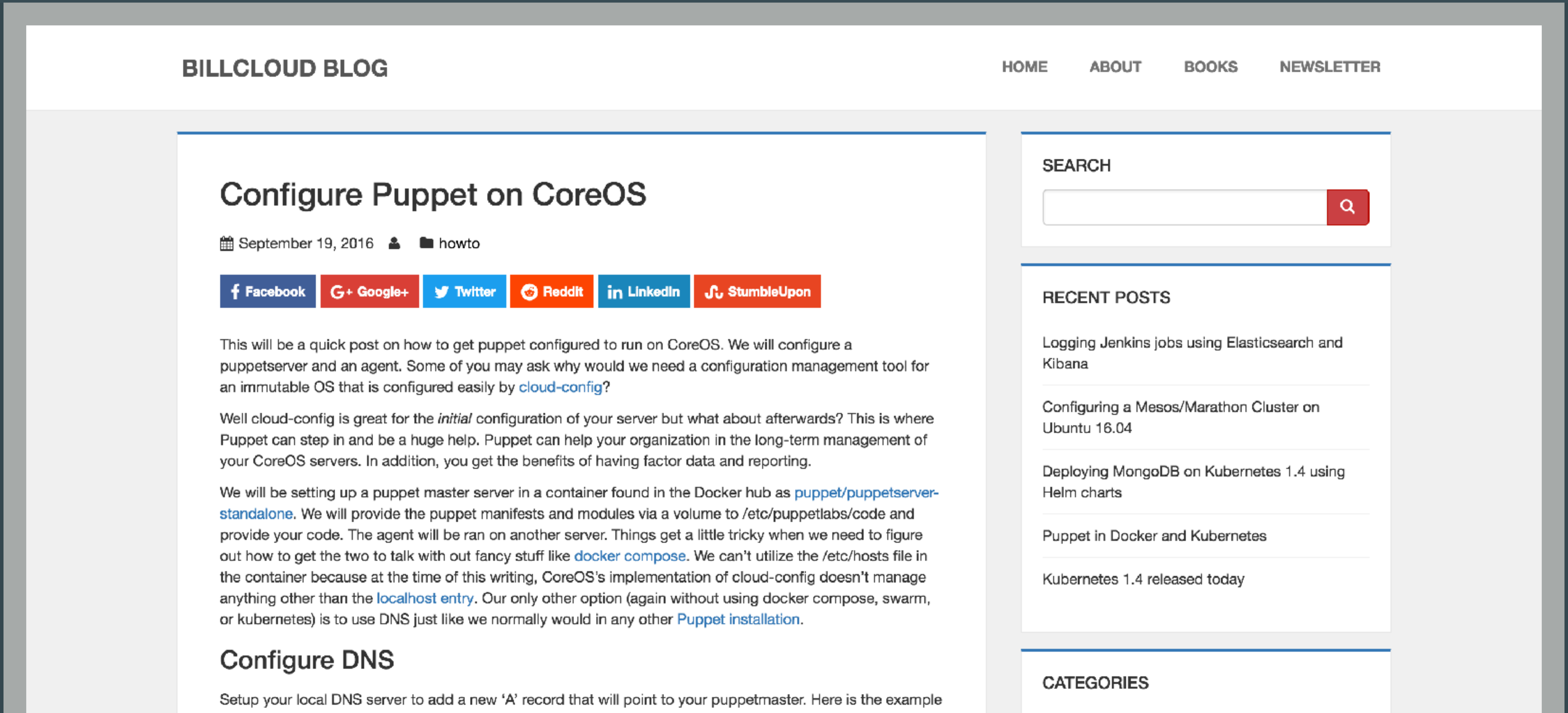
> “Continuous (re)configuration: add or modify services without reinstalling or rebooting”

École Polytechnique Fédérale de Lausanne

> “Specialized configuration of individual nodes when you really do need it. eg. gateway node with the physical Ethernet connection to the outside world ”

École Polytechnique Fédérale de Lausanne

@billcloud_me



BILLCLOUD BLOG

Configure Puppet on CoreOS

September 19, 2016 howto

This will be a quick post on how to get puppet configured to run on CoreOS. We will configure a puppetserver and an agent. Some of you may ask why would we need a configuration management tool for an immutable OS that is configured easily by cloud-config?

Well cloud-config is great for the *initial* configuration of your server but what about afterwards? This is where Puppet can step in and be a huge help. Puppet can help your organization in the long-term management of your CoreOS servers. In addition, you get the benefits of having factor data and reporting.

We will be setting up a puppet master server in a container found in the Docker hub as puppet/puppetserver-standalone. We will provide the puppet manifests and modules via a volume to /etc/puppetlabs/code and provide your code. The agent will be ran on another server. Things get a little tricky when we need to figure out how to get the two to talk with out fancy stuff like docker compose. We can't utilize the /etc/hosts file in the container because at the time of this writing, CoreOS's implementation of cloud-config doesn't manage anything other than the localhost entry. Our only other option (again without using docker compose, swarm, or kubernetes) is to use DNS just like we normally would in any other Puppet installation.

Configure DNS

Setup your local DNS server to add a new 'A' record that will point to your puppetmaster. Here is the example

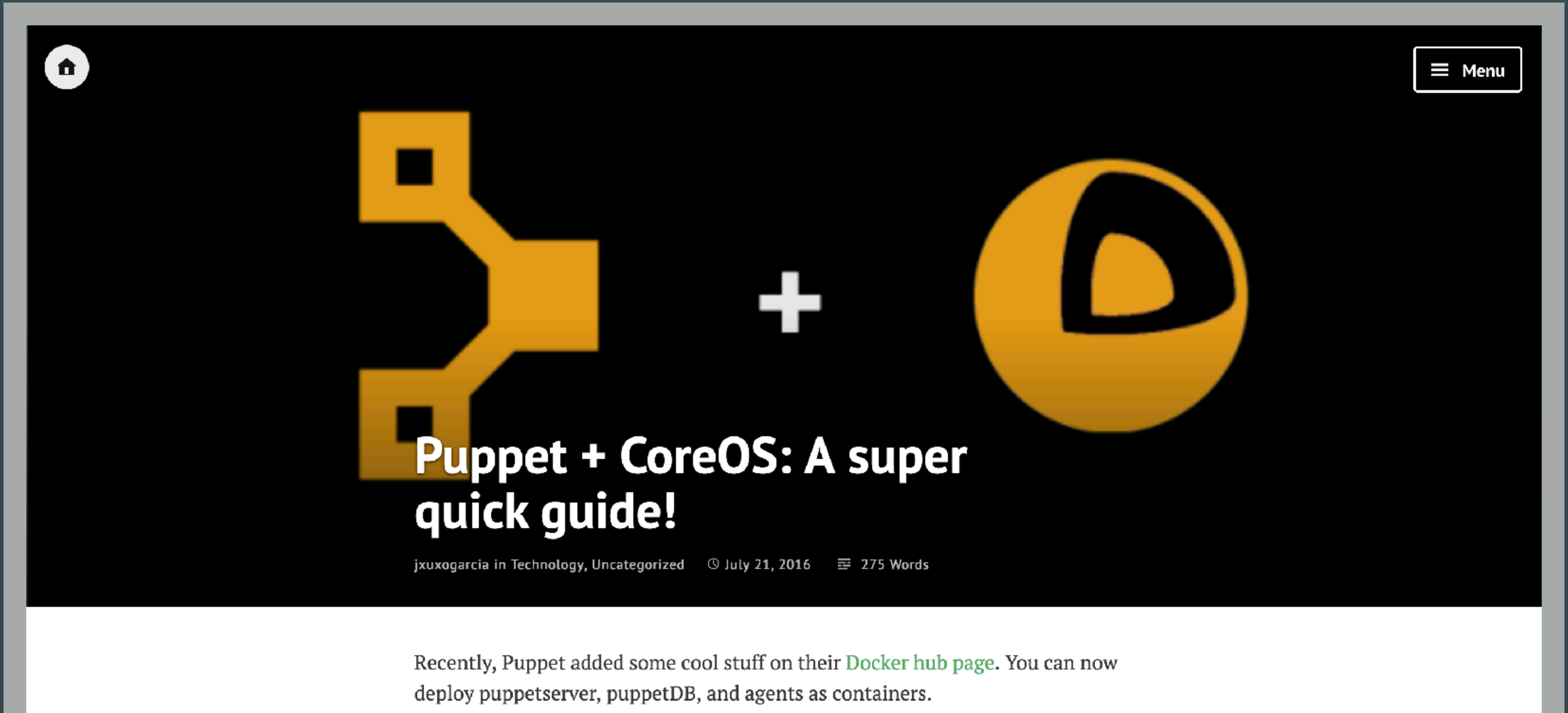
SEARCH

RECENT POSTS

- Logging Jenkins jobs using Elasticsearch and Kibana
- Configuring a Mesos/Marathon Cluster on Ubuntu 16.04
- Deploying MongoDB on Kubernetes 1.4 using Helm charts
- Puppet in Docker and Kubernetes
- Kubernetes 1.4 released today

CATEGORIES

@GarciaXuxo



Puppet + CoreOS: A super quick guide!

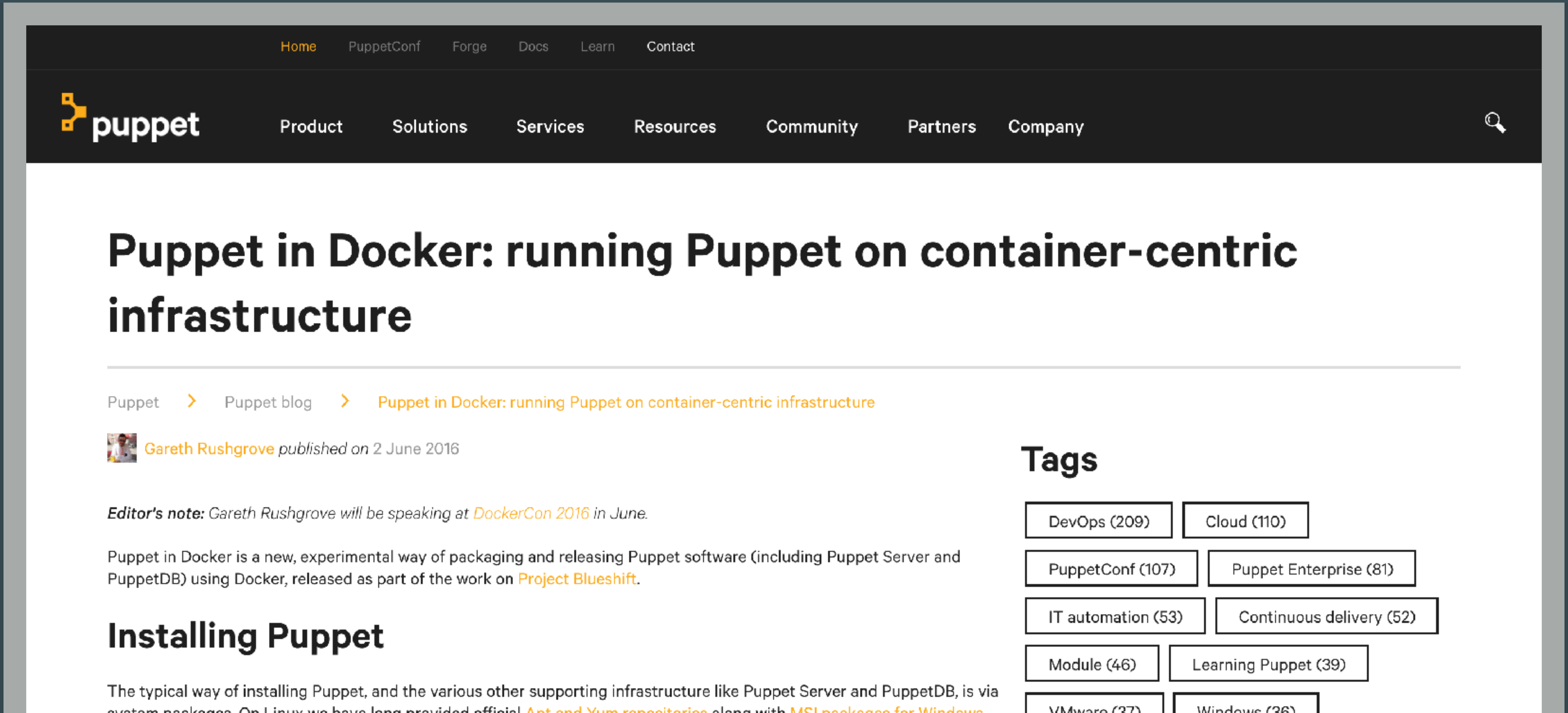
jxuxogarcia in Technology, Uncategorized   July 21, 2016   275 Words

Recently, Puppet added some cool stuff on their Docker hub page. You can now deploy puppetserver, puppetDB, and agents as containers.

How to run Puppet

When everything is a container

Container-centric infrastrucure



Puppet in Docker: running Puppet on container-centric infrastructure

Puppet > Puppet blog > Puppet in Docker: running Puppet on container-centric infrastructure

Gareth Rushgrove *published on* 2 June 2016

***Editor's note:** Gareth Rushgrove will be speaking at DockerCon 2016 in June.*

Puppet in Docker is a new, experimental way of packaging and releasing Puppet software (including Puppet Server and PuppetDB) using Docker, released as part of the work on Project Blueshift.

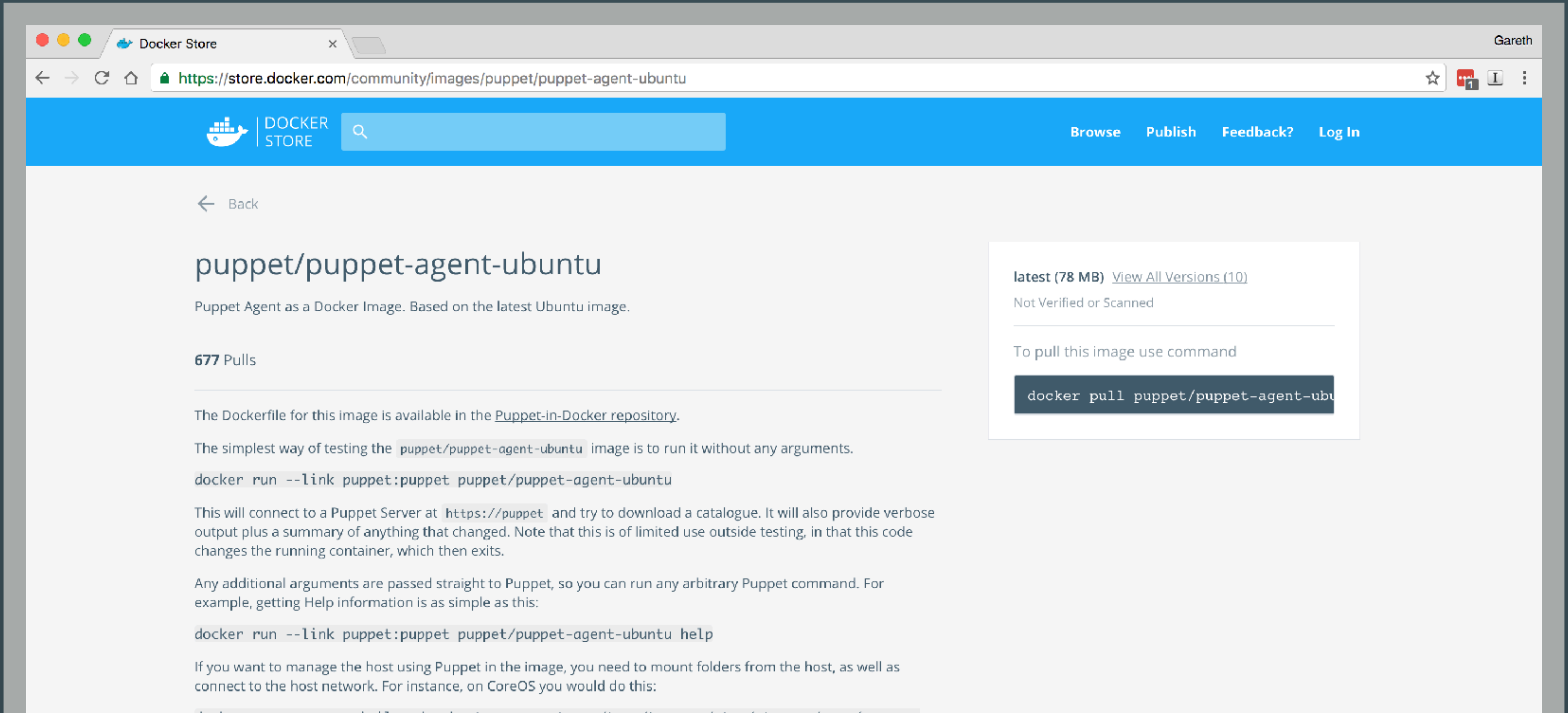
Installing Puppet

The typical way of installing Puppet, and the various other supporting infrastructure like Puppet Server and PuppetDB, is via system packages. On Linux we have long provided official Apt and Yum repositories along with MSI packages for Windows

Tags

DevOps (209)
Cloud (110)
PuppetConf (107)
Puppet Enterprise (81)
IT automation (53)
Continuous delivery (52)
Module (46)
Learning Puppet (39)
VMware (37)
Windows (36)

Available on Docker Store


Docker Store
Gareth
https://store.docker.com/community/images/puppet/puppet-agent-ubuntu
DOCKER STORE
Browse
Publish
Feedback?
Log In
Back

puppet/puppet-agent-ubuntu

Puppet Agent as a Docker Image. Based on the latest Ubuntu image.

677 Pulls

latest (78 MB) View All Versions (10)

Not Verified or Scanned

To pull this image use command

```
docker pull puppet/puppet-agent-ubu
```

The Dockerfile for this image is available in the Puppet-in-Docker repository.

The simplest way of testing the `puppet/puppet-agent-ubuntu` image is to run it without any arguments.

```
docker run --link puppet:puppet puppet/puppet-agent-ubuntu
```

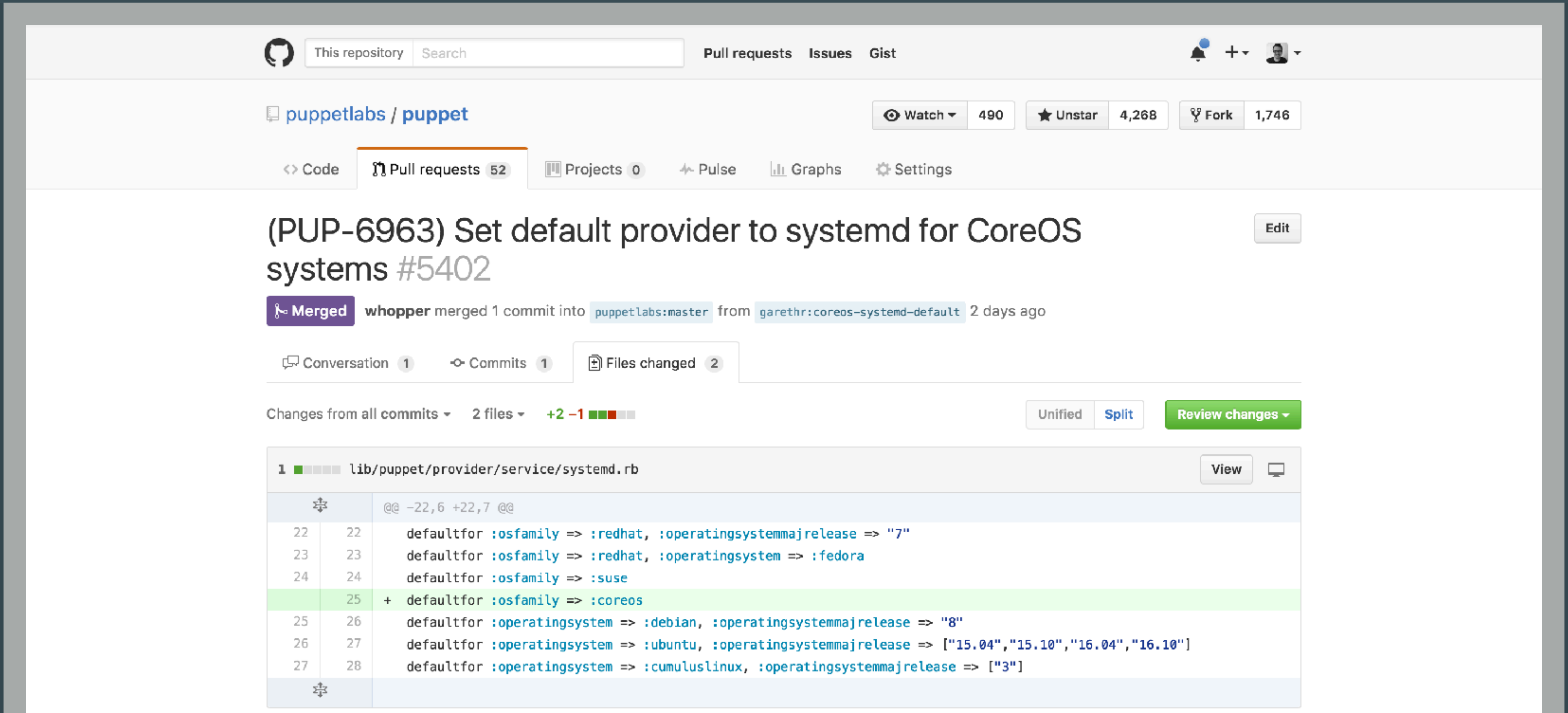
This will connect to a Puppet Server at `https://puppet` and try to download a catalogue. It will also provide verbose output plus a summary of anything that changed. Note that this is of limited use outside testing, in that this code changes the running container, which then exits.

Any additional arguments are passed straight to Puppet, so you can run any arbitrary Puppet command. For example, getting Help information is as simple as this:

```
docker run --link puppet:puppet puppet/puppet-agent-ubuntu help
```

If you want to manage the host using Puppet in the image, you need to mount folders from the host, as well as connect to the host network. For instance, on CoreOS you would do this:

Talk driven development



puppetlabs / puppet

(PUP-6963) Set default provider to systemd for CoreOS systems #5402

Merged **whopper** merged 1 commit into `puppetlabs:master` from `garethr:coreos-systemd-default` 2 days ago

1 lib/puppet/provider/service/systemd.rb

```
@@ -22,6 +22,7 @@
    defaultfor :osfamily => :redhat, :operatingsystemmajrelease => "7"
    defaultfor :osfamily => :redhat, :operatingsystem => :fedora
    defaultfor :osfamily => :suse
+   defaultfor :osfamily => :coreos
    defaultfor :operatingsystem => :debian, :operatingsystemmajrelease => "8"
    defaultfor :operatingsystem => :ubuntu, :operatingsystemmajrelease => ["15.04","15.10","16.04","16.10"]
    defaultfor :operatingsystem => :cumuluslinux, :operatingsystemmajrelease => ["3"]
```

Puppet in containers

```
$ docker pull garethr/puppet-agent-coreos
$ docker pull garethr/facter-coreos
$ docker pull puppet/r10k
```

Helpful aliases

```
alias puppet="docker run --rm --privileged \
              -v /tmp:/tmp -v /etc:/etc \
              -v /var:/var -v /usr:/usr \
              -v /var/run/dbus:/var/run/dbus \
              -v /run/systemd:/run/system \
              garethr/puppet-agent-coreos"
```

Facter

```
$ facter os
{
  architecture => "x86_64",
  family => "CoreOS",
  hardware => "x86_64",
  name => "CoreOS",
  release => {
    full => "1185.3.0",
    major => "1185",
    minor => "3"
  },
  selinux => {
```

Manage modules with r10k

```
$ docker run -v /etc:/etc \
  -v /home/core/Puppetfile:/Puppetfile:ro \
  puppet/r10k puppetfile install --verbose \
  --moduledir /etc/puppetlabs/code/modules
```

Puppet resource

```
$ puppet resource service etcd
service { 'etcd':
  ensure => 'stopped',
  enable => 'true',
}
$ puppet resource service etcd ensure=running
$ sudo systemctl status etcd
etcd.service - etcd
   Loaded: loaded (/usr/lib/systemd/system/etcd.service; static;
disabled)
   Active: active (running) since Fri 2016-12-02 16:36:13 UTC; 5
```

LIVE DEMOS

puppet

New things you can do

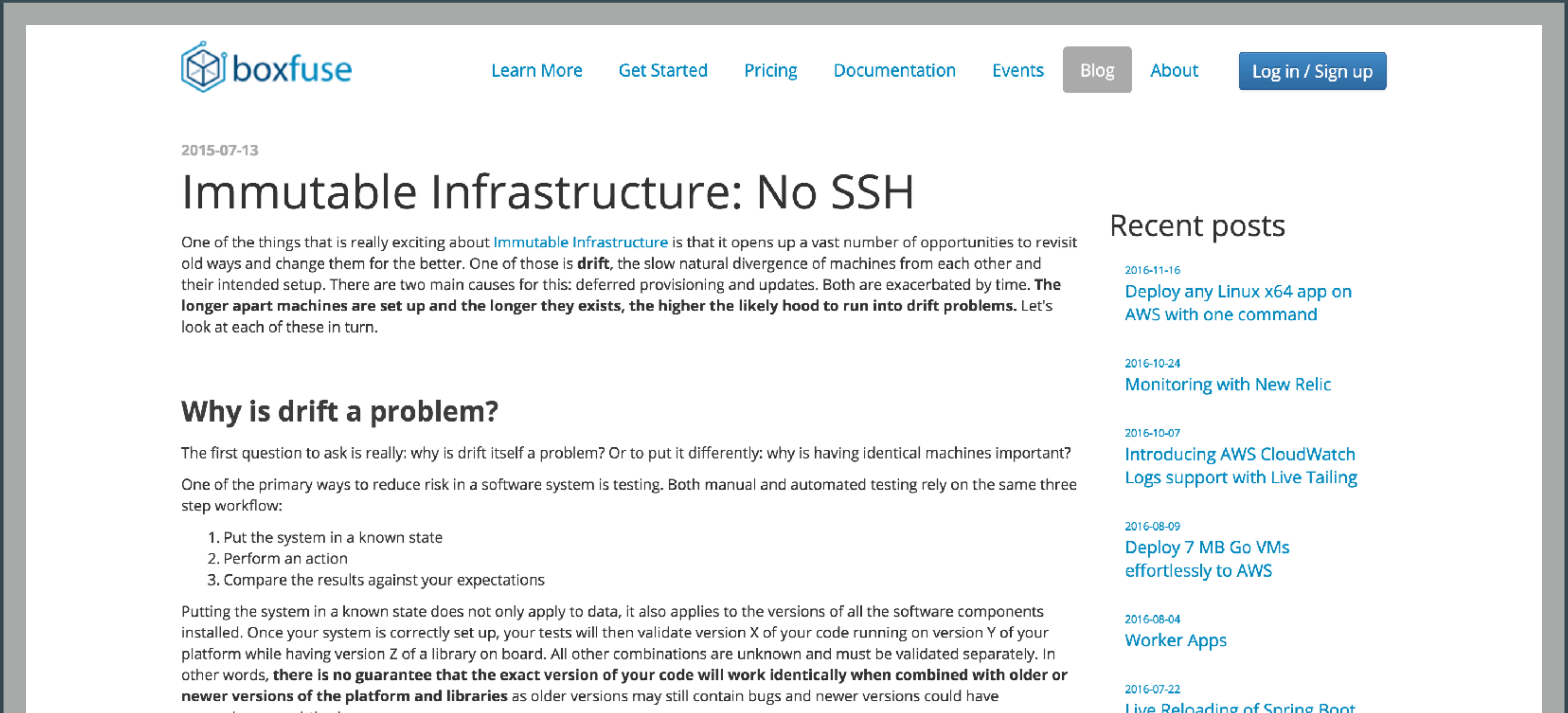
Nice hack, now what?

Obviously you can manage your users, groups, services, ssh-keys, DNS, etc. using Puppet

You can have a consistent user interface across your CoreOS and non-CoreOS hosts

(In larger organisations this can make it easier to introduce a new OS like CoreOS too)

No SSH



2015-07-13

Immutable Infrastructure: No SSH

One of the things that is really exciting about Immutable Infrastructure is that it opens up a vast number of opportunities to revisit old ways and change them for the better. One of those is **drift**, the slow natural divergence of machines from each other and their intended setup. There are two main causes for this: deferred provisioning and updates. Both are exacerbated by time. **The longer apart machines are set up and the longer they exists, the higher the likely hood to run into drift problems.** Let's look at each of these in turn.

Why is drift a problem?

The first question to ask is really: why is drift itself a problem? Or to put it differently: why is having identical machines important?

One of the primary ways to reduce risk in a software system is testing. Both manual and automated testing rely on the same three step workflow:

1. Put the system in a known state
2. Perform an action
3. Compare the results against your expectations

Putting the system in a known state does not only apply to data, it also applies to the versions of all the software components installed. Once your system is correctly set up, your tests will then validate version X of your code running on version Y of your platform while having version Z of a library on board. All other combinations are unknown and must be validated separately. In other words, **there is no guarantee that the exact version of your code will work identically when combined with older or newer versions of the platform and libraries** as older versions may still contain bugs and newer versions could have

Recent posts

2016-11-16
Deploy any Linux x64 app on AWS with one command

2016-10-24
Monitoring with New Relic

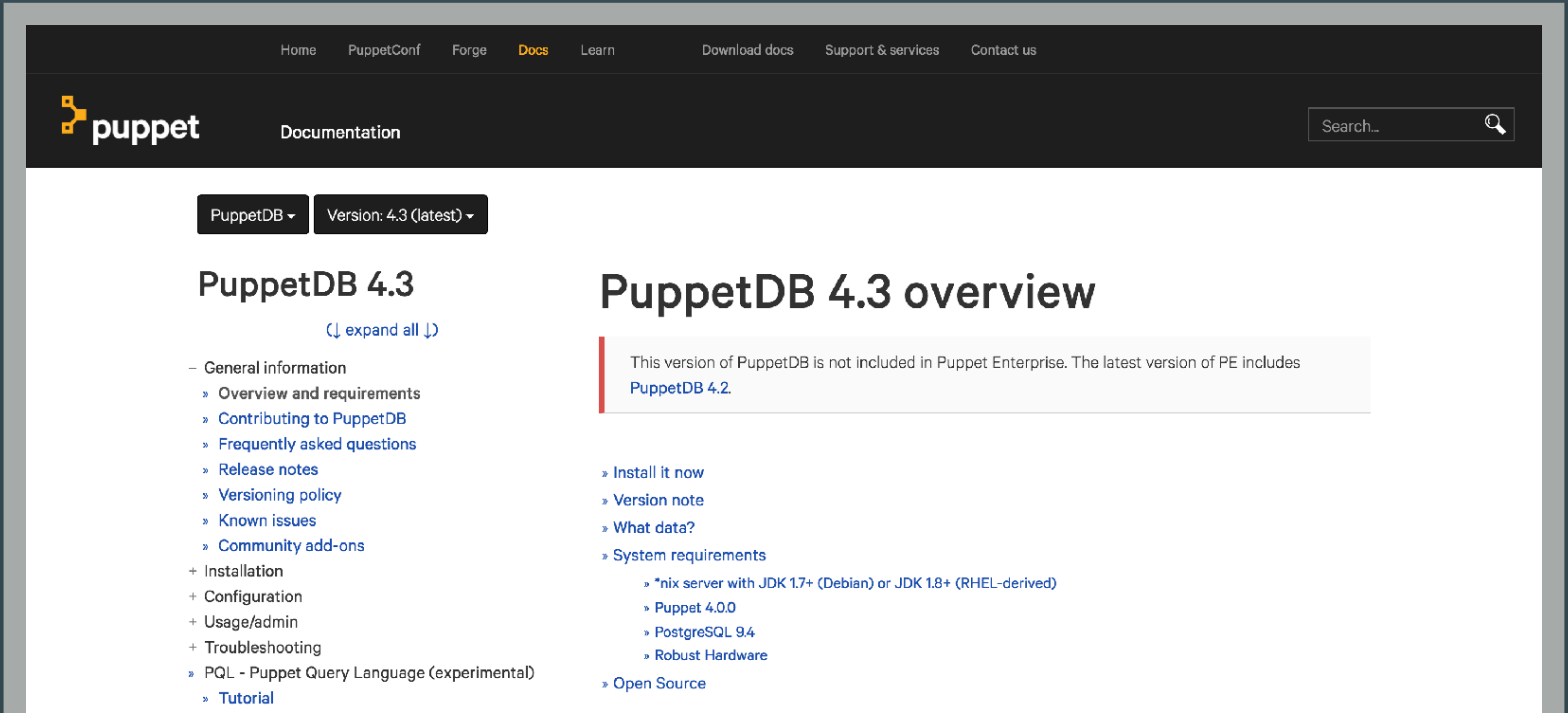
2016-10-07
Introducing AWS CloudWatch Logs support with Live Tailing

2016-08-09
Deploy 7 MB Go VMs effortlessly to AWS

2016-08-04
Worker Apps

2016-07-22
Live Reloading of Spring Boot

Inventory with PuppetDB



PuppetDB 4.3

(↓ expand all ↓)

– General information
  - » Overview and requirements
  - » Contributing to PuppetDB
  - » Frequently asked questions
  - » Release notes
  - » Versioning policy
  - » Known issues
  - » Community add-ons
+ Installation
+ Configuration
+ Usage/admin
+ Troubleshooting
» PQL - Puppet Query Language (experimental)
  - » Tutorial

PuppetDB 4.3 overview

> This version of PuppetDB is not included in Puppet Enterprise. The latest version of PE includes PuppetDB 4.2.

- » Install it now
- » Version note
- » What data?
- » System requirements
  - » *nix server with JDK 1.7+ (Debian) or JDK 1.8+ (RHEL-derived)
  - » Puppet 4.0.0
  - » PostgreSQL 9.4
  - » Robust Hardware
- » Open Source

Puppet Query Language

```
inventory { facts.os.name = "CoreOS" }
```

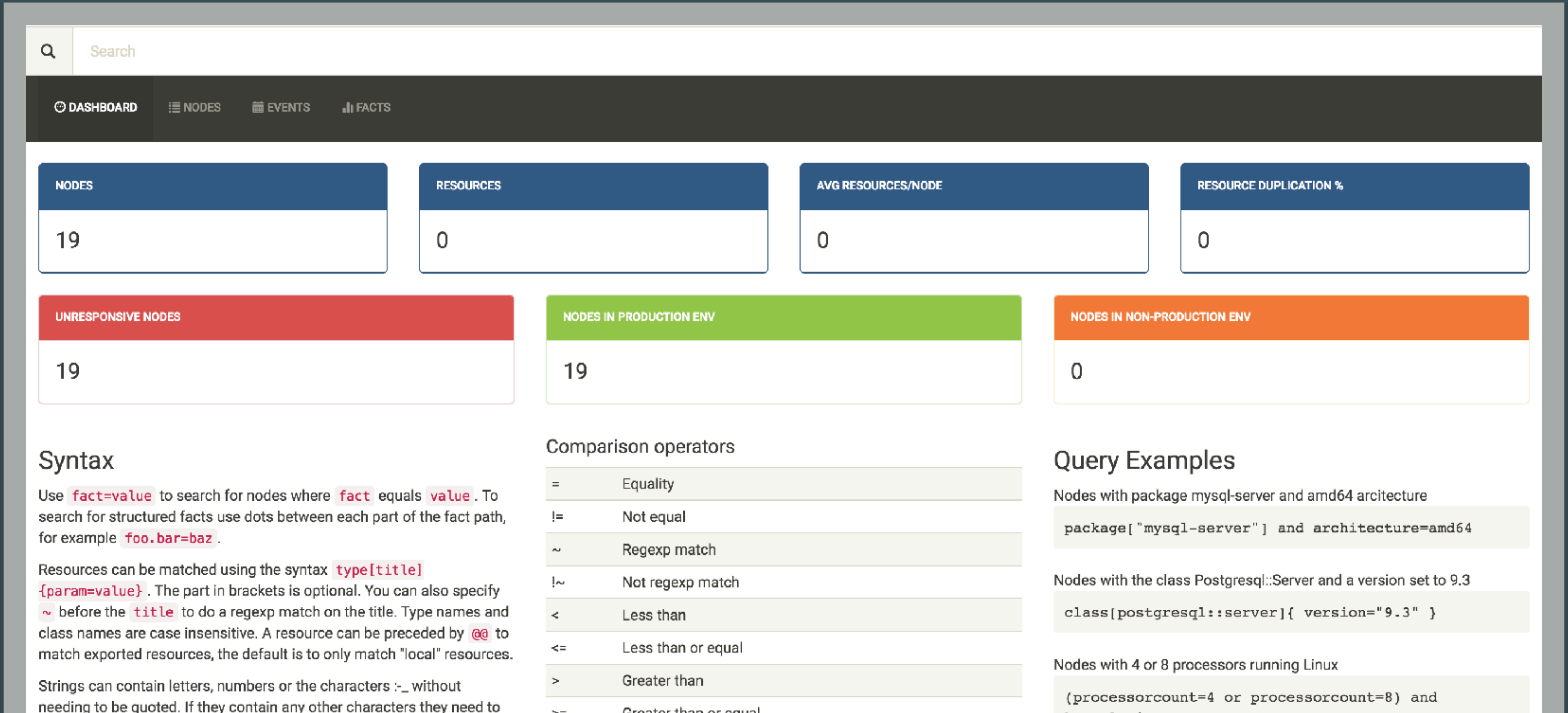
Nodes not running latest

```
nodes[certname] { facts.osfamily = "CoreOS" and
                !(facts.os.release = "1185.3.0") }
```

More complex queries

```
inventory { facts.osfamily = "CoreOS" and
            facts.datacentre = "Lon1" and
            resources { type = "Service" and
                        title = "etcd" and
                        parameters.ensure = "stopped" } }
```

Visibility and dashboards



NODES	RESOURCES	AVG RESOURCES/NODE	RESOURCE DUPLICATION %
19	0	0	0

UNRESPONSIVE NODES	NODES IN PRODUCTION ENV	NODES IN NON-PRODUCTION ENV
19	19	0

Syntax

Use `fact=value` to search for nodes where `fact` equals `value`. To search for structured facts use dots between each part of the fact path, for example `foo.bar=baz`.

Resources can be matched using the syntax `type[title]{param=value}`. The part in brackets is optional. You can also specify `~` before the `title` to do a regexp match on the title. Type names and class names are case insensitive. A resource can be preceded by `@@` to match exported resources, the default is to only match "local" resources.

Strings can contain letters, numbers or the characters :-_ without needing to be quoted. If they contain any other characters they need to

Comparison operators

=	Equality
!=	Not equal
~	Regexp match
!~	Not regexp match
<	Less than
<=	Less than or equal
>	Greater than
>=	Greater than or equal

Query Examples

Nodes with package mysql-server and amd64 arcitecture

```
package["mysql-server"] and architecture=amd64
```

Nodes with the class Postgresql::Server and a version set to 9.3

```
class[postgresql::server]{ version="9.3" }
```

Nodes with 4 or 8 processors running Linux

```
(processorcount=4 or processorcount=8) and
```

Questions?

And thanks for listening

puppet