Vagrant Puppetmaster demo

Marc Lambrichs

June 11, 2015

Abstract

First goal is to create a test environment for any application or set of applications using a central - but local! - vagrant box running puppetmaster, r10k and puppetdb.

In a different project we're creating a test environment with a combination of Elasticsearch, Logstash, Kibana (ELK-stack), combined with redis for caching. Having your own test environment allows you to develop and test your own puppetmodules, before pushing them on into a CI pipeline.

1 Prerequisites

Your environment should contain:

1. git
2. ruby
3. vagrant
4. virtualbox
5. and finally, some rubygems:
   (a) vagrant-cachier
   (b) vagrant-hosts
   (c) vagrant-vbguest

Example: installing vagrant-cachier:

```
marc:~ marc$ sudo gem install vagrant-cachier
```

2 Structure

The puppetmaster functionality is split up in 3 separate repositories.

1. puppetmaster
2. hiera
3. r10k

All three of them, are - separate - git repositories, with different purposes. The *puppetmaster* repo contains a Vagrantfile that serves as our starting point for building a puppetmaster. The other two repositories, *r10k* and *hiera* are being used during the puppetmaster install to create a complete production-like environment of puppetmaster on a vagrant box. Our first goal is to install puppetmaster, so we can leave the *hiera* and *r10k* repo's, for now.

So check out the puppetmaster repository in a directory of your liking.

```
marc:~ marc$ mkdir vagrant
marc:~ marc$ cd vagrant
marc:~ marc$ git clone https://github.com/mlambrichs/puppetmaster-vagrant.git puppetmaster-vagrant
marc:~ marc$ cd puppetmaster
marc:~ marc$ git submodule update --init --recursive
```

Now, you're ready to install your own puppetmaster.

3 Boxes

4 Step 1: Puppetmaster

4.1 Puppetmaster, r10k, puppetdb and foreman in one go

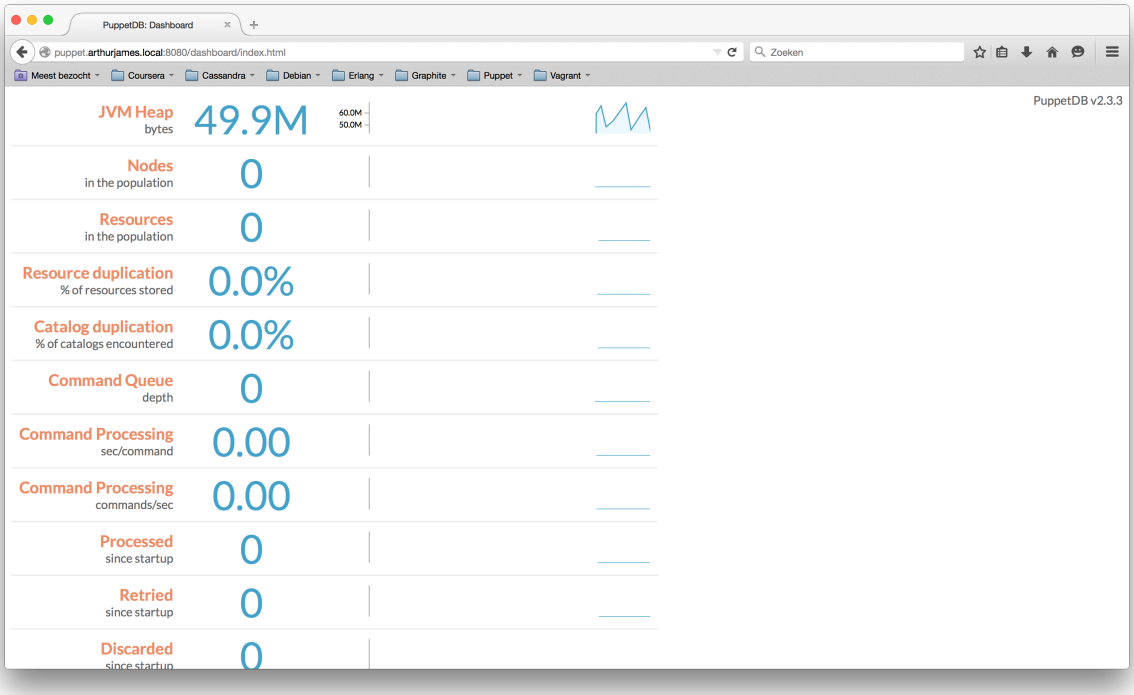
Cd into your puppetmaster directory and start up vagrant.

```
marc:~ marc$ cd ~/vagrant/puppetmaster
marc:puppetmaster marc$ vagrant up
```

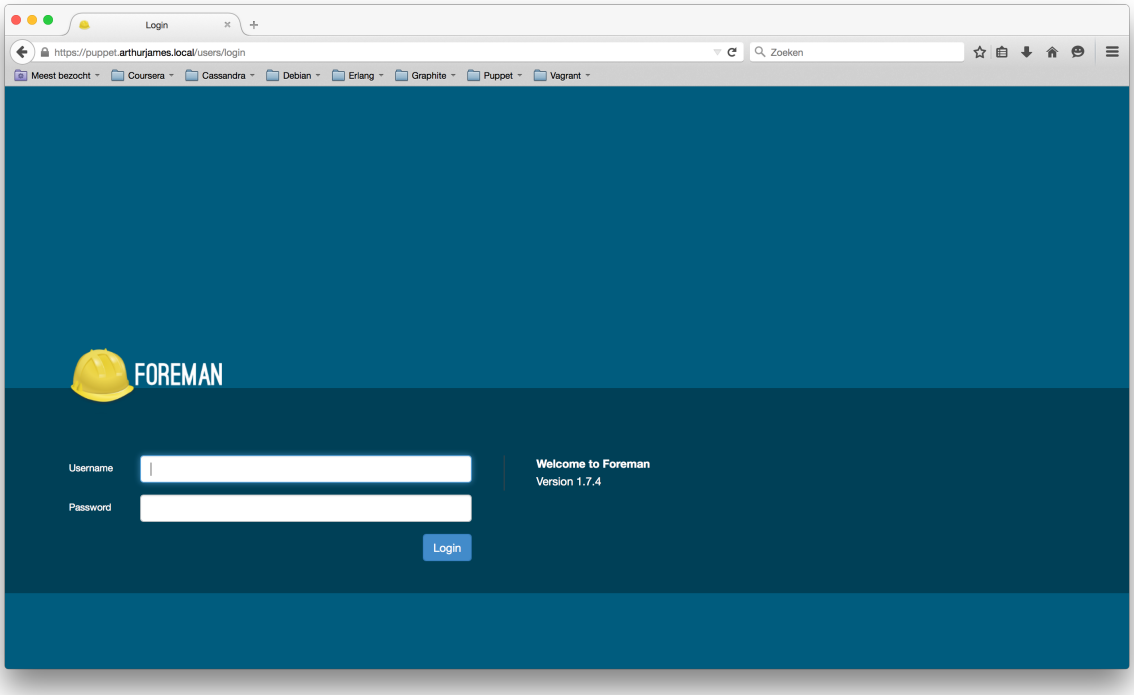
This will start off a lot of output, ending with a *puppet apply* on your vagrant box. Next thing to do, is to check if everything is working.

4.2 Check and doublecheck

First, let's check if your puppetdb is running. Open your browser with http://puppet.arthurjames.vagrant:8080 and you should see something like this:



Next, check if you can access foreman. On the same host on port 80 you should see:



If you can, check if you can log in to foreman with credentials *admin/changeme*.
Probably no nodes have been recognized, so there are two options here:

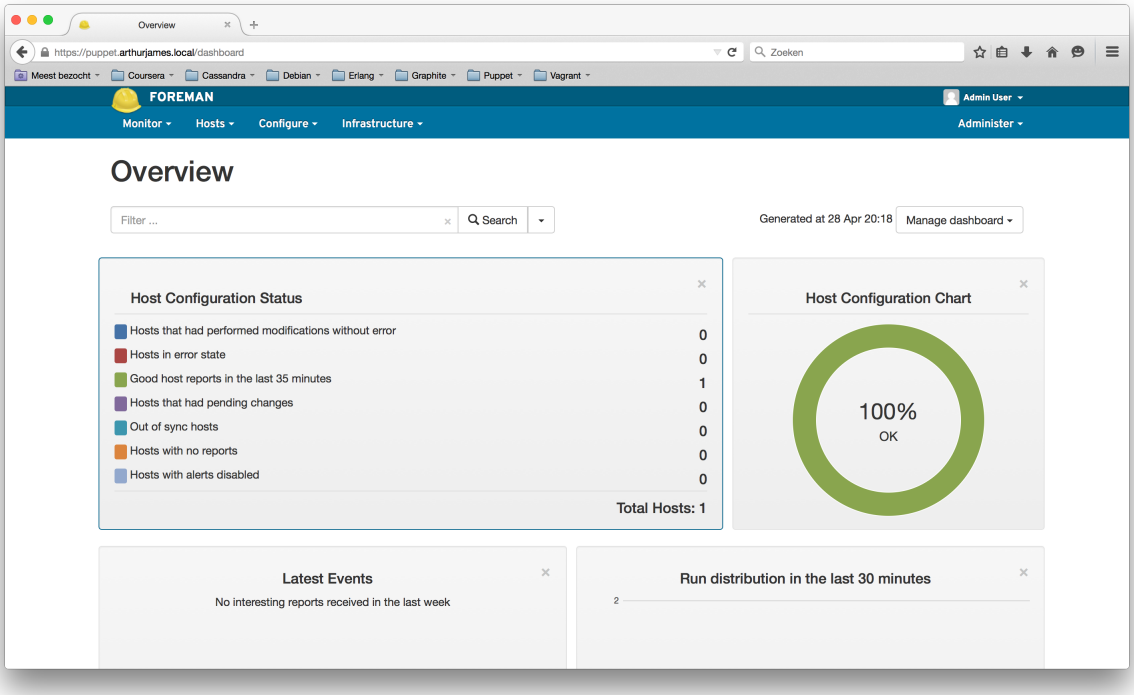
- wait for 30 min (the default setting inbetween puppet agent runs)
- ssh into your vagrant box to kick off a puppetrun.

```
marc:~ marc$ cd ~/vagrant/puppetmaster
marc:puppetmaster marc$ vagrant ssh
Linux debian 3.2.0-4-amd64 #1 SMP Debian 3.2.65-1+deb7u1 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr 28 12:47:02 2015 from 10.0.2.2
vagrant@puppet:~
```

Now you're logged in, do:

```
vagrant@puppet:~$ sudo puppet agent -v -t
```

After the puppetrun, you should be able to see 1 node defined.



5 Step 2: Elasticsearch box

5.1 native box

6 Clients

6.1 Elasticsearch

6.2 Kibana

6.2.1 Kibana 3

6.2.2 Kibana 4

6.3 Logstash

6.4 Redis