

Build your own OpenShift

Jorge Morales
OpenShift Developer Advocate



I'm Jorge Morales



I work at



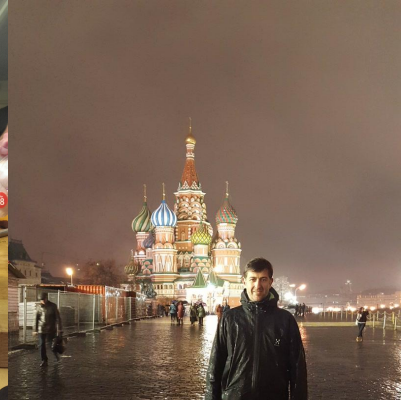
redhat.®

as a Developer Advocate for



I do:

- demos
- workshops
- talks
- conferences
- blogs
- travel
- drink beer
- ...



And I wanted OpenShift in a VM

What did I already have access to?

A whole lot of options

- OpenShift Origin Vagrant
- CDKv2 (Container Development Kit)
- adb (Atomic Development Bundle)
- OpenShift 3 demo Vagrant environment
- Multi-Host OpenShift Enterprise via Vagrant
- OpenShift Enterprise v3 Automated Installation
- OpenShift Enterprise 3 nodes setup with Vagrant and quick installer
- oinc - OpenShift in Container
- Fabric8
- And many more...

OpenShift Origin Vagrant

- Official Image by OpenShift community for developing in Origin
- <https://github.com/openshift/origin/blob/master/CONTRIBUTING.adoc#develop-on-virtual-machine-using-vagrant>

PRO:

- Community version
- Continuously maintained

CON:

- Only set up the VM and git cloning. A lot of manual steps (Registry, Router, Monitoring, Logging,...)
- No easy update

Container Development Kit v2 CDKv2

- Red Hat's official Image for OpenShift Enterprise developers
- https://access.redhat.com/downloads/content/293/ver=2/rhel---7/2.0.0/x86_64/product-software

PRO:

- Full image. Ready to work
- Continuously maintained

CON:

- Enterprise version
- Installation process still incomplete (Beta4 at the moment)
- Vagrant plugins required to set up
- Multipurpose project. Lot of manual steps required to set up an OpenShift VM.

Atomic Developer Bundle

- Upstream version of CDK.
- <https://github.com/projectatomic/adb-atomic-developer-bundle>

PRO:

- Community version
- Full image. Ready to work
- Continuously maintained

CON:

- Vagrant plugins required to set up.
- Multipurpose project. Lot of manual steps required to set up an OpenShift VM.

OpenShift 3 demo Vagrant environment

- Automated installation of an environment for using with openshift/training material
- <https://github.com/jorgemoralespou/jboss-virtual-environments/tree/master/vagrant-vm/openshift3>

PRO:

-

CON:

- Enterprise version
- Not maintained. Working with 3.0
- Not an all-in-one VM

Multihost OpenShift Enterprise via Vagrant

-
- <https://github.com/thoraxe/vagrant-openshift-multihost>

PRO:

- Active developed.

CON:

- Enterprise version
- Not an all-in-one VM

OpenShift Enterprise 3 automated installation

-
- <https://github.com/jcordes73/ose-installation-automation>

PRO:

-

CON:

- Enterprise version
- Not maintained. Working with 3.0
- Not an all-in-one VM

OpenShift Enterprise 3 nodes setup w/ Vagrant and quick installer

-
- <https://github.com/nekop/openshift-sandbox/tree/master/vagrant/openshift-enterprise>

PRO:

-

CON:

- Enterprise version
- Not maintained. Working with 3.0
-

OpenShift In Container (oinc)

-
- <https://github.com/mfojtik/oinc/>

PRO:

- Community version

CON:

-

Fabric8 Vagrant Image

-
- <http://fabric8.io/guide/getStartedVagrant.html>

PRO:

- Community version

CON:

- Vagrant plugins required to set up.
-

None of these worked for me!

When I say me, I mean my team.

To be more precise, I mean TheSteve0

What we wanted:

- Community based
- Open source
- Maintained up-to-date
- Easy to understand
- Flexible in options
 - VM provisioning configuration tunable
 - VM runtime configuration tunable
- Packageable. Can make a base-box for redistribution out of it.

So here, I'm presenting....

OpenShift Origin's Vagrant EVG Version

As easy as:
vagrant up

No additional plugins required!

No fancy port redirection!

Fully configured!

Fully maintained and supported!

By me :-D

Provisioning options:

```
ORIGIN_REPO = ENV['ORIGIN_REPO'] || "openshift"
ORIGIN_BRANCH = ENV['ORIGIN_BRANCH'] || "master"
PUBLIC_ADDRESS = ENV['ORIGIN_VM_IP'] || "10.2.2.2"
PUBLIC_DOMAIN = ENV['ORIGIN_VM_DOMAIN'] || "apps.#{PUBLIC_ADDRESS}.xip.io"
VM_MEM = ENV['ORIGIN_VM_MEM'] || "4096" # Memory used for the VM
ACTION = ENV['ACTION'] || "none" # (none, clean, build, config)
CONFIG = ENV['CONFIG'] || "osetemplates,metrics" # testusers,originimages,centosimages,rhelimages,
xpaasimages,otherimages,osetemplates,metrics
FORCE_PREREQS = ENV['FORCE_PREREQS']
FORCE_DOCKER = ENV['FORCE_DOCKER']
FORCE_ADDONS = ENV['FORCE_ADDONS']
BUILD_IMAGES = ENV['BUILD_IMAGES'] || "false" # (true|false)
JOURNAL_SIZE = ENV['JOURNAL_SIZE'] || "100M" # (Use a number suffixed by M,G)
DOCKER_STORAGE_SIZE = ENV['DOCKER_STORAGE_SIZE'] || "30G" # (Use a number suffixed by G)
```


Runtime options:

```
vb.memory = "4096"
```

```
vb.cpus = 2
```

It works with:
Libvirt and VirtualBox

Scripted in:
bash

4 simple scripts:

os

docker

origin

addons

OS:

- install docker git go
- limit journal size (configurable)

docker:

- configure and start docker
 - container filesystem size

origin:

- checkout or update origin source code
- build origin
- configure origin
- start origin
- add services (registry, router, templates)

addons:

- metrics
- logging (WIP)
- ose templates
- additional users (WIP)
- pull down images (origin, centos builders, rhel builders, xpaas)

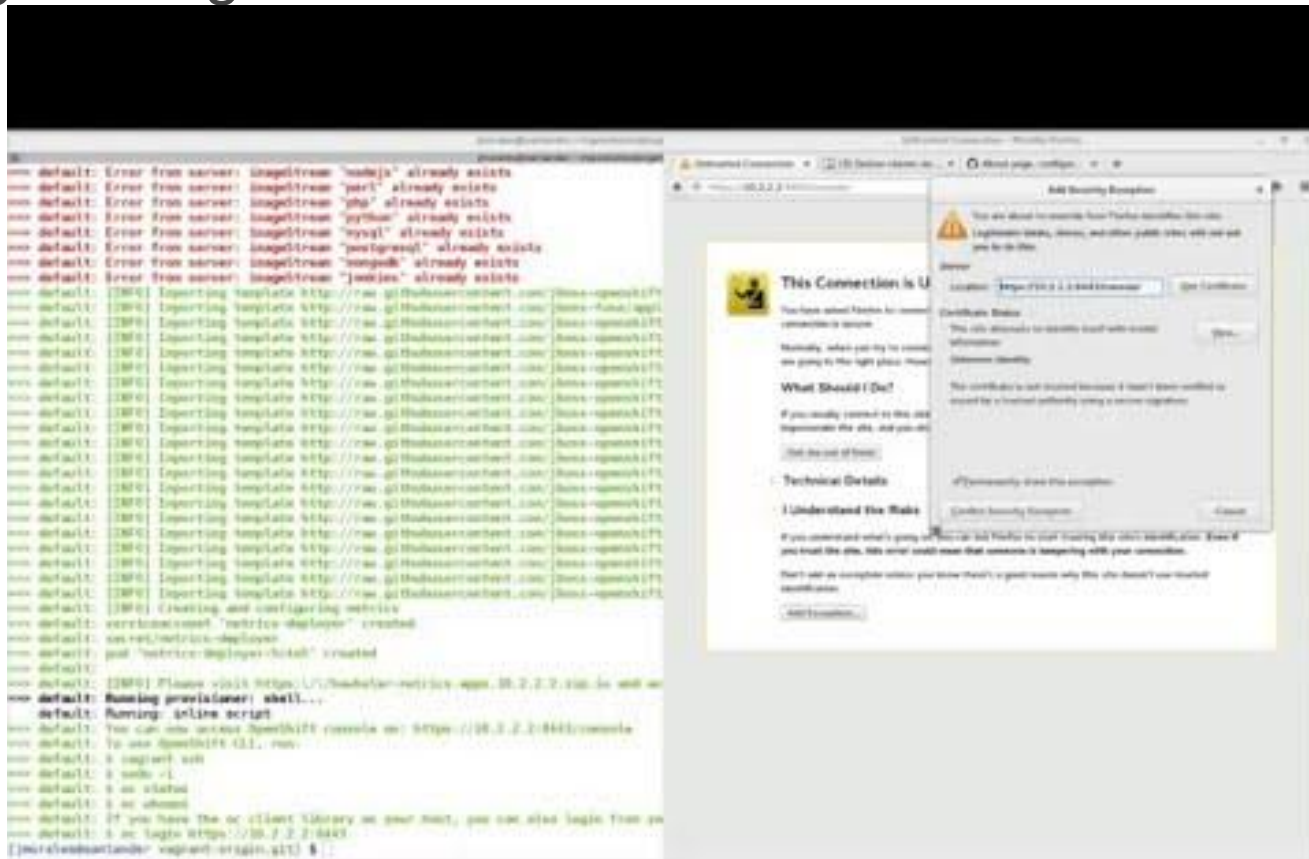
Creating an Origin VM from scratch

```
root@: total 27 (delta 0); reused 9 (delta 0); pack-reused 58
Uploading objects: 100% (80/80), done.
Checking connectivity... done.
[[mcrc@localhost ~]$ top -d vagrant-origin/
[[mcrc@localhost ~]$ vagrant up --provision --no-tty --no-shell
1.1.0.1 are available. 80498 and 80499 are available.
[[mcrc@localhost ~]$ vagrant up --provision --no-tty --no-shell
Current machine status:

default      not created (libvirt)

The libvirt domain has not been created. Run 'vagrant up' to create it.
[[mcrc@localhost ~]$ vagrant up --provision --no-tty --no-shell
Bringing machine 'default' up with 'libvirt' provider.
vagrant default: Creating image (consist of base box volume).
vagrant default: Creating domain with the following settings...
vagrant default: -- Name:                vagrant-origin_default
vagrant default: -- Domain type:                kvm
vagrant default: -- Cpus:                        2
vagrant default: -- Memory:                      8192M
vagrant default: -- Base box:                    Fedora-28
vagrant default: -- Storage pool:                default
vagrant default: -- Image:                       /var/lib/libvirt/images/vagrant-origin_default.img
vagrant default: -- Volume cache:                default
vagrant default: -- Kernel:                      default
vagrant default: -- Initrd:                      default
vagrant default: -- Graphics type:               vga
vagrant default: -- Graphics port:               5900
vagrant default: -- Graphics IP:                 127.0.0.1
vagrant default: -- Graphics password:           not defined
vagrant default: -- Video type:                  cirrus
vagrant default: -- Video VRAM:                  32M
vagrant default: -- Keymap:                      us
vagrant default: -- Command line:                --
vagrant default: Creating shared folders manually...
vagrant default: Starting domain.
vagrant default: Waiting for domain to get an IP address...
vagrant default: Waiting for SSH to become available...
vagrant default: Setting hostname...
vagrant default: Configuring and enabling network interfaces...
vagrant default: Syncing folder: /tmp/vagrant-origin/scripts/ => /scripts
vagrant default: Syncing folder: /tmp/vagrant-origin/utila/ => /utila
vagrant default: Running provisioner: shell...
vagrant default: Running: /tmp/vagrant-shell/20181216.0072-000000.sh
vagrant default: Last successful update: 2018-12-16 00:00:00 UTC. 13 days ago. For more info, see: https://vagrant.io/docs/2.2.0/upgrade-guide#2.2.0
```

Updating an Origin VM with a contributor's branch



Check it out:
YET TO BE RELEASED!!!

Use it!

<http://bit.ly/OpenshiftOriginEVGVagrant>

But don't kill me if you don't like it

Submit issues, PR, ...

Thank you for listening to me

or not