Deploying a Jenkins server on Digital Ocean with Vagrant

One of the most popular Continuous Integration (CI) tools out there today is a nifty product named Jenkins. For those who want to take a look at how to utilize Jenkins, but you may not have had a place to try it out, I have just what you need!

You'll need a few things for the build:

1. Vagrant – http://www.vagrantup.com/downloads.html (Free)
2. Vagrant DigitalOcean plugin (https://github.com/smdahlen/vagrant-digitalocean)
3. A DigitalOcean account – If you don't have one yet, go here: http://discopos.se/GetDigitalOcean
4. Your DigitalOcean API Keys (both Client and Secret) and SSH key
5. Git – you'll have to check out my Github repo to deploy the system

Let's get started and show you get all the information you need and how to put these components together!

NOTE: This will create a server that will be a 1GB DigitalOcean Droplet which costs 10$ per month if left active continuously

Getting the GitHub repository

Depending on your system, you'll need to either pull down the code via the GitHub GUI, or the command line. The repository is easy to grab in the command line like so:

```
git clone https://github.com/discoposse/Vagrant-DigitalOcean-Jenkins.git
```

Now that you have that sorted out, you will need to head out to the command line and begin building our Vagrant configuration. Although we have the files needed to deploy our server, there is some

Getting your DigitalOcean Key info

You will have to log in to http://digitalocean.com and under your account page, you will find the API option in the left hand menu which shows your API key information. You will see the **Client ID** and your **API Key**.

In the event that you see your API Key as hidden (shown below) you will have to use the **Generate New Key** button at the top right. NOTE: Each time you regenerate an API key, it will invalidate your previous key so you should keep this key somewhere safe as it gives full access to your account.

Now that you have your API Key and Client ID, you also need your matching SSH key that was used during the setup of your account. For all the SSH key information, you can use this guide at DigitalOcean: https://www.digitalocean.com/community/articles/how-to-use-ssh-keys-with-digitalocea

n-droplets

Setting up your Vagrantfile

In the folder where you download there is a file named **Vagrantfile.sample** that you can copy, paste and rename to **Vagrantfile** with no file extension.

This is the folder where you will be storing your **id_rsa** and **id_rsa.pub**. Once you have those to SSH files there and your newly copied **Vagrantfile** the folder should look like this:

Inside your **Vagrantfile** you will be putting in your customer specific information which are your Client Key and your API Key. Just replace the text inside the **Vagrantfile** in the appropriate lines:

That's all the preparation we need for our file. Now let's make sure our Vagrant configuration is ready to deploy to DigitalOcean.

Installing the DigitalOcean Vagrant plugin

In your command prompt or shell, change directory to the folder where your GitHub repository was cloned. From there, you can also deploy (or confirm the deployment) of the Vagrant plugin for DigitalOcean (https://github.com/smdahlen/vagrant-digitalocean).

In the shell, run this command:

```
vagrant plugin install vagrant-digitalocean
```

Now we are ready to launch our DigitalOcean Droplet which will run our Jenkins server.

Deploying Jenkins with Vagrant and the DiscoPosse Quick Build

In the command shell, you have one simple command to run which is as follows:

```
vagrant up –provider=digital_ocean
```

This will trigger the deployment to your DigitalOcean account. The script will begin as shown here:

The script will run for about 5-10 minutes as it updates the instance with the Ubuntu updates and Jenkins server code. Once it is completed you will see a message similar to this indicating that your droplet is ready.

Now we will launch our browser to view the Jenkins application interface. First you will need the IP address of the instance you have launched.

To get the IP address, you can view your DigitalOcean control panel and under the Droplets menu you will see the instance information like the sample here:

Open a browser window using the the IP address with port 8080 to access Jenkins:

```
http://yourIPaddress:8080
```

The first thing that you will want to do is to add users and authentication requirements, so I recommend that you click the Manage Jenkins link to get that started:

Under the Jenkins management screen you can enable security in a number of ways. This will be up to you as to how much you choose to lock down your application. Just click the **Setup Security** button to do this:

And just like that you have your first Jenkins server on DigitalOcean. It is just that easy thanks to Vagrant!

Getting rid of your Jenkins instance

Once you get a chance to test things out, or if you want to rebuild your machine, you can easily remove the server using Vagrant with this command:

```
vagrant destroy -f
```

This makes it very simple for you to create and destroy your lab environment as needed. Especially if you are like me and have a number of projects on the go that don't always get full attention. By using vagrant destroy, you are no longer being charged for your instance.

I hope that you find this helpful and that it can help you on your journey to discover Jenkins.

Tech Field Day VFD3 – Spirent – the art of the test

It's another busy day with Tech Field Day VFD3 of exploring some great vendors! We were introduced to Spirent today which was a new experience for many of the delegates. Spirent has a broad set of offerings which you may already be familiar with though, as they actually have a strong presence in the GPS and mobile space using their testing products with NFV/SDN, and in the application test automation space also.

If you want to see an impressive portfolio of services, a quick over over the product list at Spirent will give you a hint as to how heavily leveraged they are in many areas:

Testing 1-2-3

The win for any environment that is under heavy change and development is the ability to apply the DevOps methodologies and move all aspects of their infrastructure into a continuous integration model. This is a paradigm shift for many organizations, but for folks who have had experience in application development environments, this will be a familiar and well-regarded way to achieve efficiency and consistency.

I am a fan of the automation and orchestration to reduce the variability when building and managing hardware and software solutions. One of the real challenges comes in the difficulty of testing network designs. We do our best to plan how to build our physical network topologies and test them in place before going live, but we also know how difficult it can be to do effective, repeatable testing.

Imagine how a minimally tested network infrastructure becomes exponentially more painful when it is running as an overlay network in the style of Software Defined Networking (SDN). What do we do for testing out a complex Layer 2-7 topology? And just imagine if it is in a dynamic, self-service environment such as using Cisco ACI or VMware NSX.

Enter Spirent

Network Virtualization (NV) and encapsulation create some very distinct challenges in how we can test them at design, at deployment, and then continuously during operation.

This diagram is a great depiction of the flow if implementation that was presented:

The soon to be released solution from Spirent is meant to answer the question of "How can I introduce continuous integration and testing into my network and NV platform?".

The continuous integration fan in me lit up at the thought of this

Spirent has a great history in application testing and L2-L7 testing already, but the addition of this new suite to push the testing into the NV space is very exciting. Dynamic network testing in virtual data centers with orchestration baked in really does feel like it could be a panacea, so for this reason

I will be eagerly watching for their announcement at Interop at the end of the month (http://www.interop.com/lasvegas/).

Automate All the Things

This is my style! I am a firm believer in the value of automation as you'll see in the theme of many of my posts and tweets. There is a reason of course, and that is the long term value for application, infrastructure, and network environments has been proven to be real when orchestration and testing has been integrated.

In the classic tagline: There's an app for that! Spirent showcased their iTest product (http://www.spirent.com/Products/iTest) and chatted lots about the advantages, both tangible (aka $$$$) and intangible (aka time-to-market reductions and smiling employees and customers).

There is definitely some great potential for the platform to be a part of many organizations, so if you would like to dig in a little deeper, make sure to reach out to the folks at Spirent (@Spirent on Twitter) and take a look at their wide array of offerings online at http://www.spirent.com for more info.

And if the theme hasn't been hammered home enough: Continuous Integration should be in your thoughts as you design your infrastructure, period.