

Using Vagrant-Onap for local deployment of SDC project (Deprecated since Guilin)



This is a guide on how to work with the vagrant-onap as a local deployment vagrant.

Table of Contents

- [Prerequisites](#)
- [Initial Setup](#)
- [Using the vagrant](#)
- [Deploying local dockers to the vagrant](#)
- [Setting up the Webseal-Simulator](#)

Prerequisites

Before starting working with the vagrant-onap you'll need to following versions of vagrant and virtualbox:

1. **vagrant** version 2.0.1. Download it from here: <https://www.vagrantup.com/downloads.html>
2. **virtualbox** version 5.2.2 r119230 (Qt5.6.3). Download it from here: <https://www.virtualbox.org/wiki/Downloads>

Make sure to restart your computer in order for all the necessary configuration to be configured on your system.



Note: this versions were tested by us, newer versions may work as well but were not tested.

Initial Setup

1. Clone the integration project: `git clone http://gerrit.onap.org/r/integration/devtool`.
2. The next stages have to be done before trying to run the vagrant:
 - a. If you're going through proxy in your network you'll have to configure it as an environment variable under `http_proxy` and `https_proxy` (**What about no_proxy?**)
Restart your cmd for the changes to take affect. You can type `set <ENV_VAR_NAME>` and see if the environment variables were saved.
 - b. In the virtualbox you need to set the networking settings as followed: (**Need to add screenshots to each stage**)
 - i. Open oracle VM VirtualBox Manager and click on filePreferences
 - ii. Go to Network tab and there click on Host-only Networks tab
 - iii. Click on the icon for adding a new network adapter
 - iv. After the new adapter has been created click select in from the list, right-click->edit
 - v. Change the IPv4 Address to Be: 192.168.50.1
 - c. Go to the folder you've cloned the vagrant to and type in the command: `vagrant plugin install vagrant-proxyconf`. This is the plugin that will take care of the proxy configuration inside the vagrant.
 - d. The following files in the project needs to be changed from Windows style(CR LF) to Unix style(LF): **lib/_composed_functions**, **lib/_onap_functions**, **lib/commons**, **lib/functions**, **lib/sdc**.
You can do the following by using on windows for example by using notepad++ and right clicking on the lower right bottom where it writes Windows(CR LF) and select Unix(LF).
 - e. Copy the **scripts**, **environments** folders from the **sdc-os-chef** project in the sdc repository(What if they don't have it on their computer?) to **vagrant-onap/opt** folder
Also in the utils folder in sdc there is the **webseal-simulator** project. From there also copy the contents of the **scripts** folder to **vagrant-onap/opt/scripts**
Download the config.zip file from here and extract the contents to **vagrant-onap/opt/config**
[config.zip](#)
 - f. Edit the **Template.json** in **/sdc-os-chef/environments/Template.json** fill relevant fields for chef environment JSON.
Latest **Template.json** <https://gerrit.onap.org/r/gitweb?p=sdc.git;a=blob;f=sdc-os-chef/environments/Template.json;hb=HEAD>
3. The next changes needs to be applied to the code in order for the vagrant to work as we want it:
(Some of the changes here should be merged in the code in future updates so We'll update this guide accordingly)
 - a. In the Vagrantfile in the main folder in the bottom of the configuration clause change the next values to the following:

```
configuration = {
  'build_image' => 'False',
  'clone_repo' => 'False',
  'compile_repo' => 'False',
  'skip_get_images' => 'True',
  'skip_install' => 'False'
}
```

b. In the script file **lib\sdcc** the following changes should be applied:

i. In the **function install_sdcc** after the **call to init_data_folders** replace the cp and chmod commands block to the following:

```
cp /opt/environments/Template.json /data/environments
cp /opt/scripts/docker_run.sh /data/scripts
cp /opt/scripts/docker_health.sh /data/scripts
cp /opt/scripts/docker_login.sh /data/scripts
cp /opt/scripts/docker_clean.sh /data/scripts
cp /opt/scripts/simulator_docker_run.sh /data/scripts
chmod +x /data/scripts/docker_run.sh
chmod +x /data/scripts/docker_health.sh
chmod +x /data/scripts/docker_login.sh
chmod +x /data/scripts/docker_clean.sh
chmod +x /data/scripts/simulator_docker_run.sh
```

Using the vagrant

After you've finish configuring the vagrant rom the vagrant-onap folder you can run in the cmd: **vagrant up sdc** to start the booting sequence of the vagrant. It should take a couple of minutes.

When it's done you can connect to your machine at 192.168.50.5:22 with the credentials: **vagrant(both for username and password)**.

In order to run all the commands switch user to root by running: **su -**, password for root is also vagrant.

from the machine run the command: **docker**. to make sure that the docker engine has been installed successfully.

This is a link to list of common commands to run from the vagrant: [SDC Vagrant Common Commands \(Deprecated since Guilin\)](#)

In order to close the vagrant in safe state (saving all your deployed container and images) you need to open once again the oracle VM VirtualBox Manager, right click the sdc vagrant, move your cursor to the close option and there choose save state.

Next time you want to return to work on your vagrant you can type in the cmd: **vagrant resume sdc** or start it up again using the VirtualBox Manager.

If you want to start from scratch on the vagrant you can use the command: **vagrant destroy sdc**



Using the destroy command will results in complete deletion of your vagrant including container, images and any data you had on your machine.

Use at your own risk!

Deploying local dockers to the vagrant

In order to check our code from our local environment we'll compile our code locally and upload our local images to the vagrant

This is how to do so:

1. Add **DOCKER_HOST** to environment variables with the value: **tcp://192.168.50.5:2375**
(This is the ip of the vagrant, We're using private network settings in our vagrant to allow our dockers to comunicate with each other without using proxy configurations. The ip list can be found in the Vagrantfile int the nodes array)

2. Run **mvn clean install** and build the whole project.

```
mvn clean install -P docker
```



You'll have to make sure that your vagrant is up and running before doing so or else it won't work

3. After this process is done go to your vagrant and run **docker images** to check that your local images are on the machine
4. In order to deploy your local images without pulling the latest images from the nexus you'll need to run the docker_run script with the local flag like so:

```
/data/scripts/docker_run.sh -l
```

(This command and all other commands can be found here: [SDC Vagrant Common Commands \(Deprecated since Guilin\)](#))

Setting up the Webseal-Simulator

access to SDC in production is done from the ONAP portal,

a local envirement you will need to start our simulator to allow accses to the sdc user interface instructions can be found here: [SDC Simulator](#)