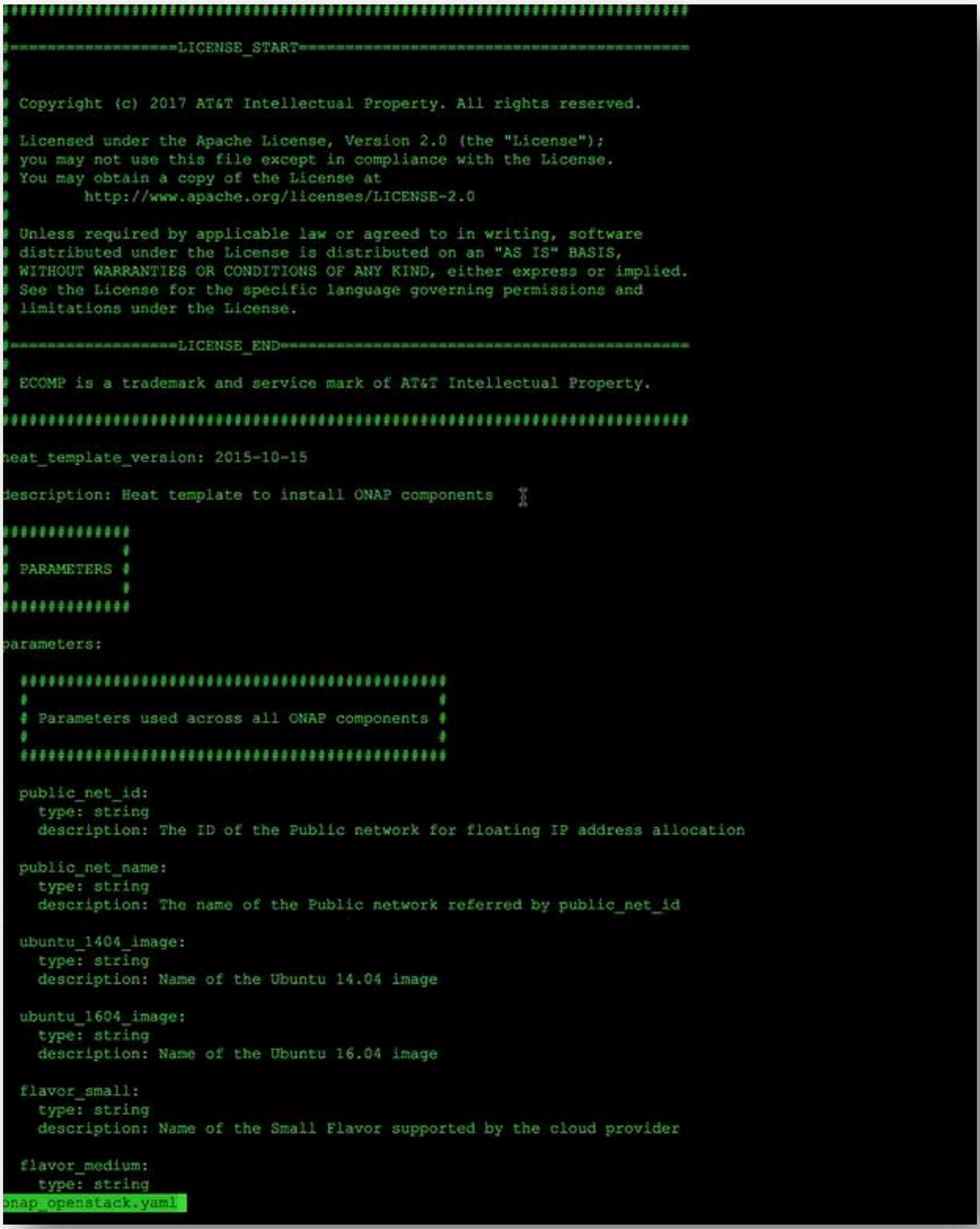
Tutorial on how to run DCAE Gen2 in release 1 (based on Lusheng's recording dcae-weekly-20171116.mp4):

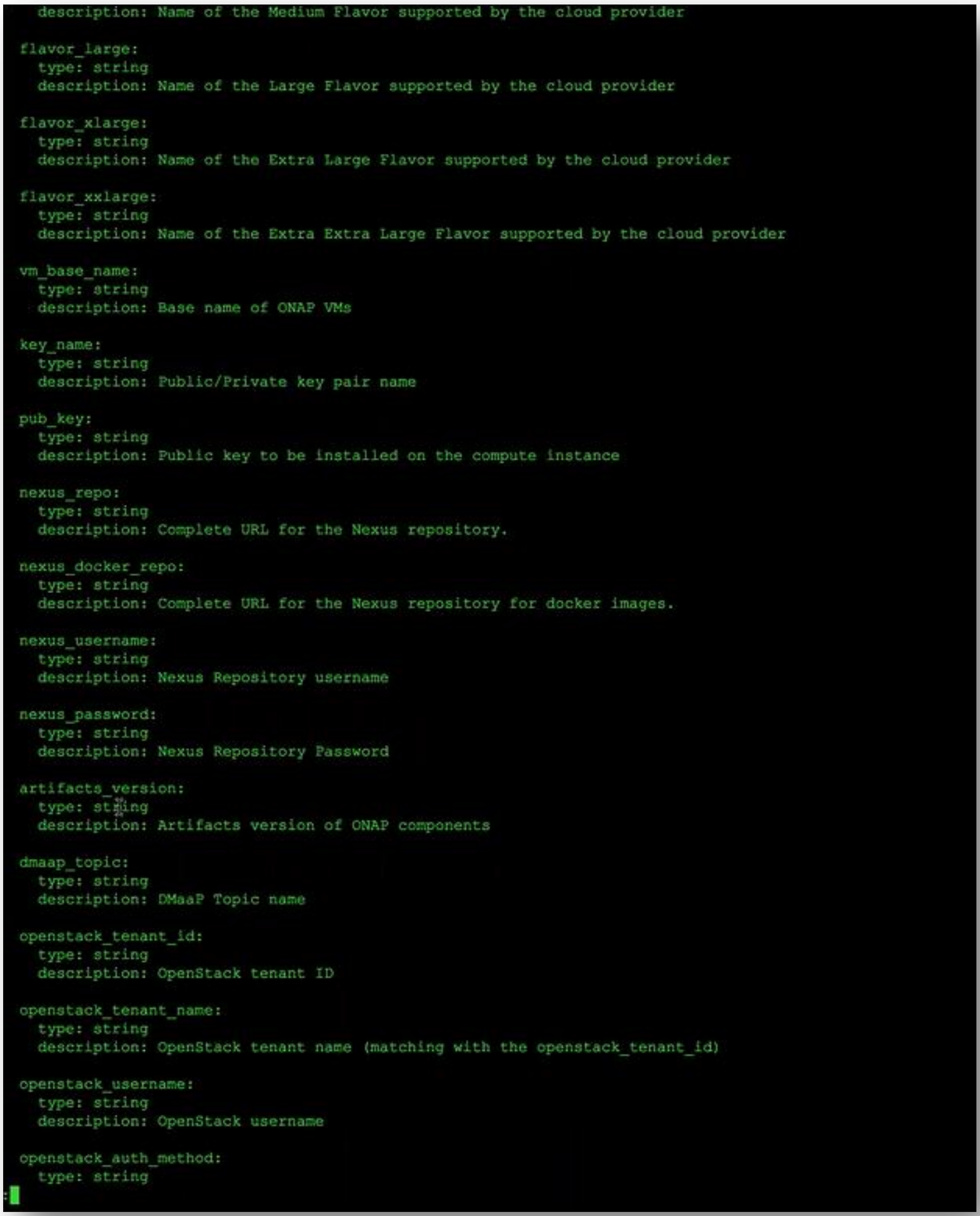
Author: Pawel Pawlak (pawel.pawlak3@orange.com)

ONAP R1 will be deployed through a Heat – special use of a yaml files, inside the file you can specify, describe how a virtual system looks like using Openstack terms, so you can describe what kind of virtual resources you need to you need Openstack to spin-up to finish your systems including networks, security groups - so key pairs and all the VMs. Also inside that you can also specify what kind of cloud in the scripts you want to run in each VM.

For people not familiar with Heat template you can see that there is a bunch of input parameters it requires and there is what kind of resources are needed,

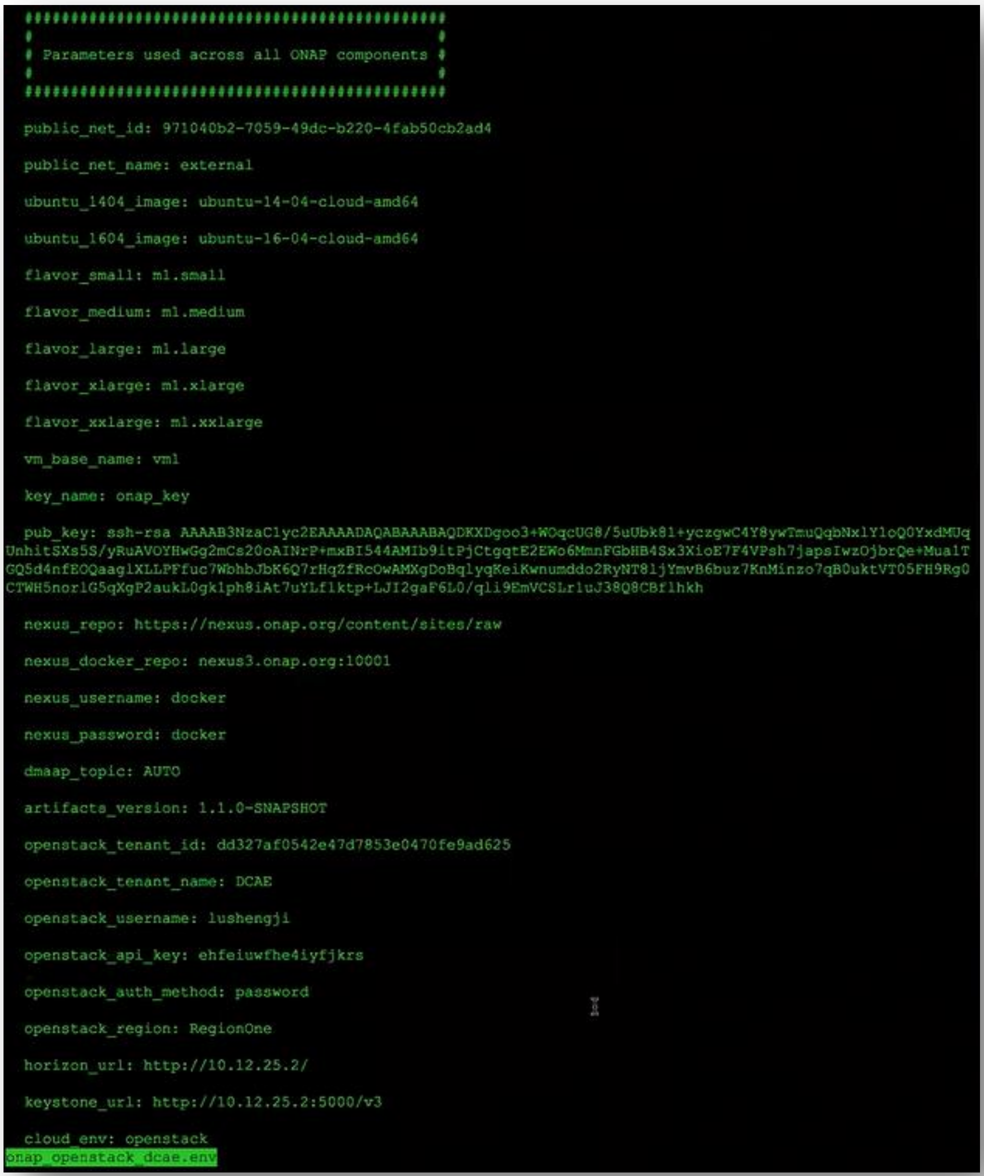


```
#################################################################################
#
#==================LICENSE_START==========================================
#
#
# Copyright (c) 2017 AT&T Intellectual Property. All rights reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#        http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
#==================LICENSE_END============================================
#
# ECOMP is a trademark and service mark of AT&T Intellectual Property.
#
#################################################################################

heat_template_version: 2015-10-15

description: Heat template to install ONAP components

##############
#            #
# PARAMETERS #
#            #
##############

parameters:

  ##############################################
  #                                            #
  # Parameters used across all ONAP components #
  #                                            #
  ##############################################

  public_net_id:
    type: string
    description: The ID of the Public network for floating IP address allocation

  public_net_name:
    type: string
    description: The name of the Public network referred by public_net_id

  ubuntu_1404_image:
    type: string
    description: Name of the Ubuntu 14.04 image

  ubuntu_1604_image:
    type: string
    description: Name of the Ubuntu 16.04 image

  flavor_small:
    type: string
    description: Name of the Small Flavor supported by the cloud provider

  flavor_medium:
    type: string
onap_openstack.yaml
```



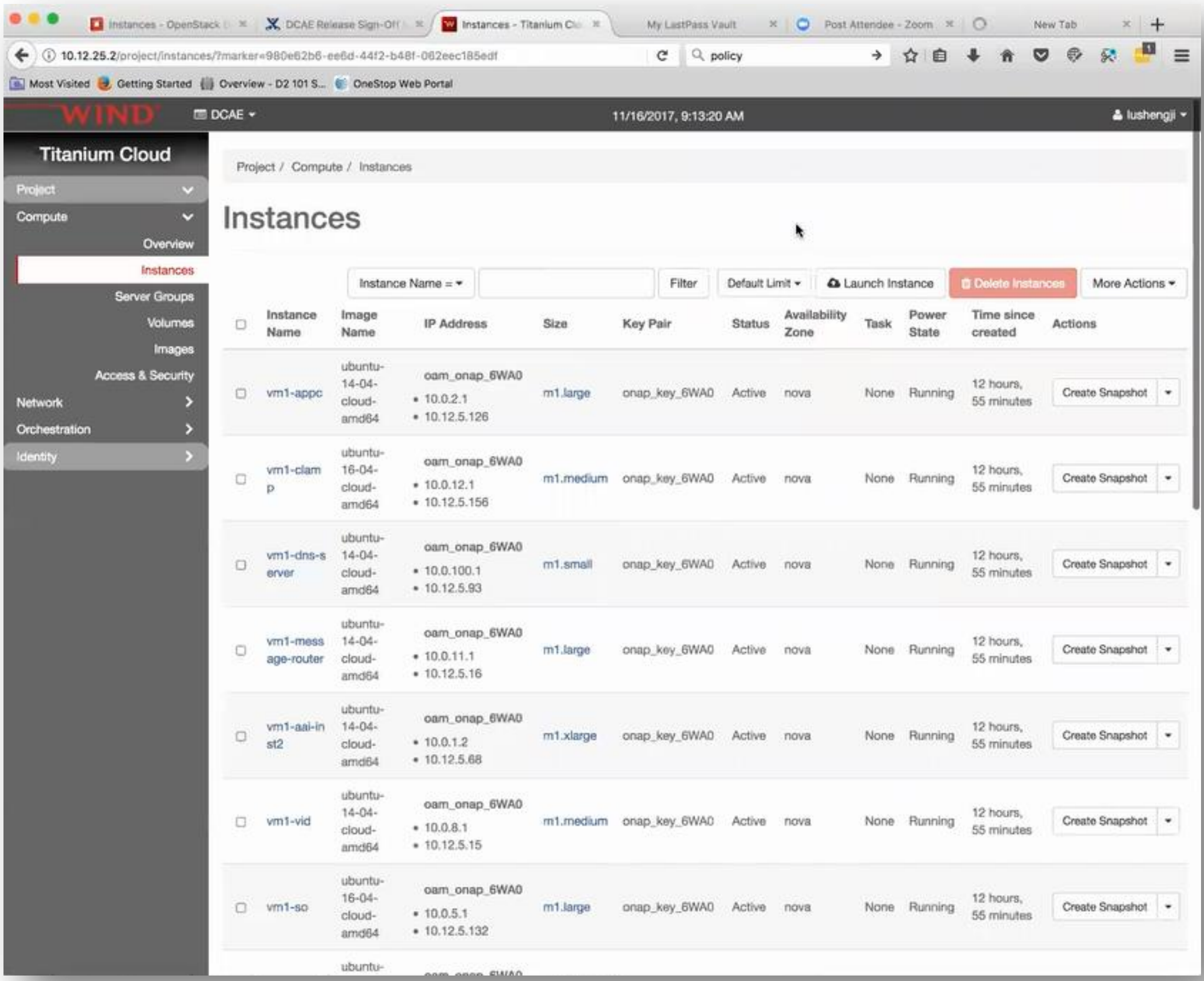
```
  description: Name of the Medium Flavor supported by the cloud provider

flavor_large:
  type: string
  description: Name of the Large Flavor supported by the cloud provider

flavor_xlarge:
  type: string
  description: Name of the Extra Large Flavor supported by the cloud provider

flavor_xxlarge:
  type: string
  description: Name of the Extra Extra Large Flavor supported by the cloud provider

vm_base_name:
  type: string
  description: Base name of ONAP VMs

key_name:
  type: string
  description: Public/Private key pair name

pub_key:
  type: string
  description: Public key to be installed on the compute instance

nexus_repo:
  type: string
  description: Complete URL for the Nexus repository.

nexus_docker_repo:
  type: string
  description: Complete URL for the Nexus repository for docker images.

nexus_username:
  type: string
  description: Nexus Repository username

nexus_password:
  type: string
  description: Nexus Repository Password

artifacts_version:
  type: string
  description: Artifacts version of ONAP components

dmaap_topic:
  type: string
  description: DMaaP Topic name

openstack_tenant_id:
  type: string
  description: OpenStack tenant ID

openstack_tenant_name:
  type: string
  description: OpenStack tenant name (matching with the openstack_tenant_id)

openstack_username:
  type: string
  description: OpenStack username

openstack_auth_method:
  type: string
:
```

when Marco or anybody from the Integration Team want to deploy a whole ONAP system, they just run Openstack and use a stack create command by spec giving command this heat template and well as input parameter file for example this is an environmental (extension .env) files are used for deploying into integration lab Pod25 there is a tenant called DCAE so this is an input parameters file



```
##############################################
#                                            #
# Parameters used across all ONAP components #
#                                            #
##############################################

  public_net_id: 971040b2-7059-49dc-b220-4fab50cb2ad4

  public_net_name: external

  ubuntu_1404_image: ubuntu-14-04-cloud-amd64

  ubuntu_1604_image: ubuntu-16-04-cloud-amd64

  flavor_small: m1.small

  flavor_medium: m1.medium

  flavor_large: m1.large

  flavor_xlarge: m1.xlarge

  flavor_xxlarge: m1.xxlarge

  vm_base_name: vm1

  key_name: onap_key

  pub_key: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDKXDgoo3+WOqcUG8/5uUbk81+yczgwC4Y8ywTmuQqbNx1Y1oQ0YxdMUq
UnhitSXs5S/yRuAVOYHwGg2mCs20oAINrP+mxBI544AMIb9itPjCtgqtE2EWo6MmnFGbHB4Sx3XioE7F4VPsh7japsIwzOjbrQe+Mua1T
GQ5d4nfEOQaag1XLLPFfuc7WbhbJbK6Q7rHqZfRcOwAMXgDoBqlyqKeiKwnumddo2RyNT8ljYmvB6buz7KnMinzo7qB0uktVT05FH9Rg0
CTWH5nor1G5qXgP2aukL0gk1ph8iAt7uYLf1ktp+LJI2gaF6L0/qli9EmVCSLr1uJ38Q8CBflhkh

  nexus_repo: https://nexus.onap.org/content/sites/raw

  nexus_docker_repo: nexus3.onap.org:10001

  nexus_username: docker

  nexus_password: docker

  dmaap_topic: AUTO

  artifacts_version: 1.1.0-SNAPSHOT

  openstack_tenant_id: dd327af0542e47d7853e0470fe9ad625

  openstack_tenant_name: DCAE

  openstack_username: lushengji

  openstack_api_key: ehfeiuwfhe4iyfjkrs

  openstack_auth_method: password

  openstack_region: RegionOne

  horizon_url: http://10.12.25.2/

  keystone_url: http://10.12.25.2:5000/v3

  cloud_env: openstack
onap_openstack_dcae.env
```

, so it contains parameters needed for any ONAP VMs + some of the special parameters for DCAE. Overall the process will be: when the stack is created it creates all ONAP VMs (14 or 15 of them) one of them is called DCAE Bootstrap – inside of this VM we run some preparation work such as verifying and creating DNS zone needed for DCAE and then inside of this VM it is gonna fetch the DCAE's bootstrap container, the container itself is going to run bootstrap script for DCAE, it is gonna interact with Openstack undernees and the launch the rest of the DCAE VMs and configure zone and launch all the necessary service components. So that is the overall process, let me connect to pod25.

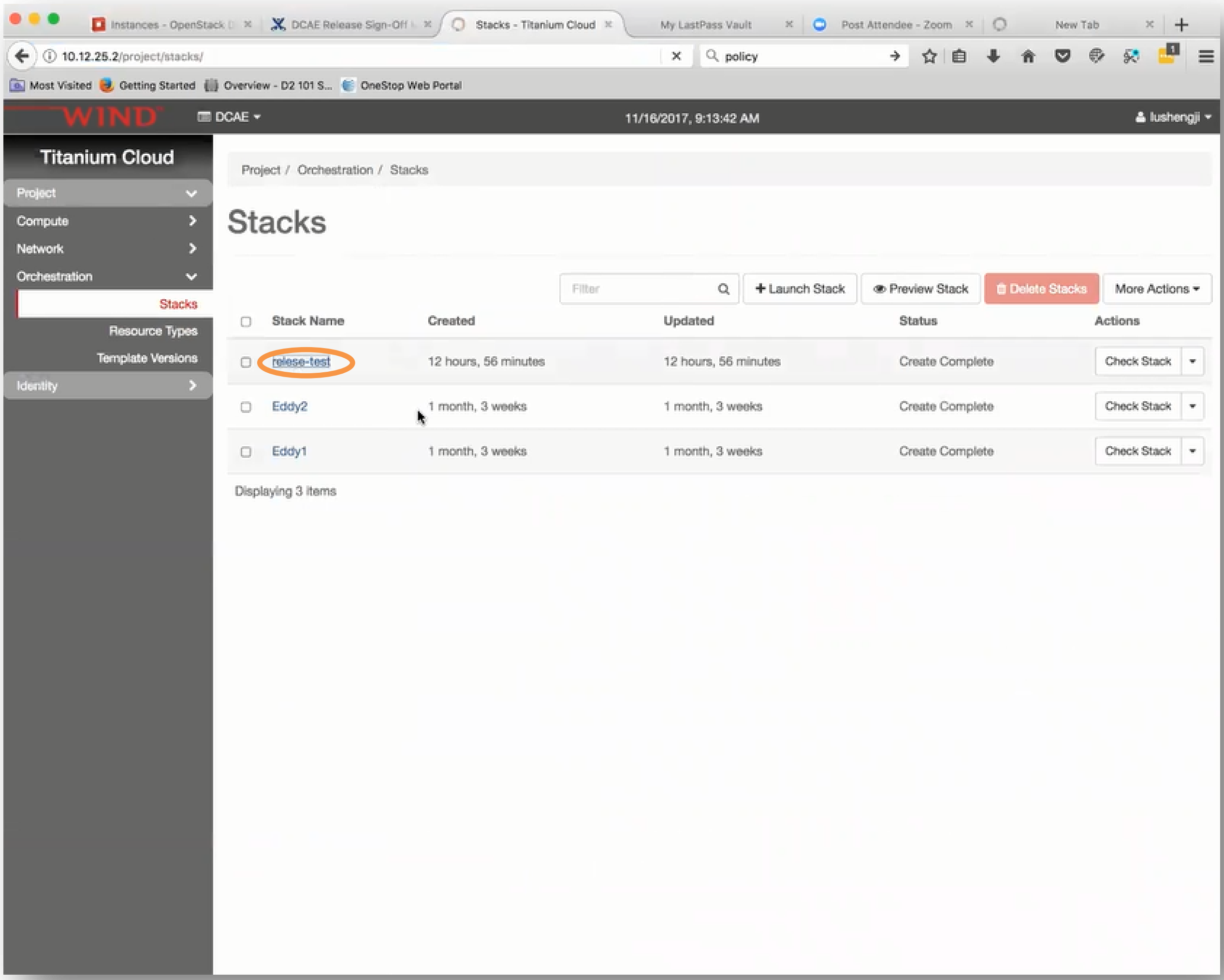


Project / Compute / Instances

Instances

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
vm1-appc	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.2.1 • 10.12.5.126	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 55 minutes	Create Snapshot
vm1-clamp	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.12.1 • 10.12.5.156	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 55 minutes	Create Snapshot
vm1-dns-server	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.100.1 • 10.12.5.93	m1.small	onap_key_6WA0	Active	nova	None	Running	12 hours, 55 minutes	Create Snapshot
vm1-message-router	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.11.1 • 10.12.5.16	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 55 minutes	Create Snapshot
vm1-aai-inst2	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.1.2 • 10.12.5.68	m1.xlarge	onap_key_6WA0	Active	nova	None	Running	12 hours, 55 minutes	Create Snapshot
vm1-vid	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.8.1 • 10.12.5.15	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 55 minutes	Create Snapshot
vm1-so	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.5.1 • 10.12.5.132	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 55 minutes	Create Snapshot

This is the dashboard for the Integration Lab. You can see that we deployed this last night after LF released all the dockers in the release registry. There is a full deployment of ONAP including DCAE. The starting point was deploying release-test stack using the template – here there are all the resources included in this stack,

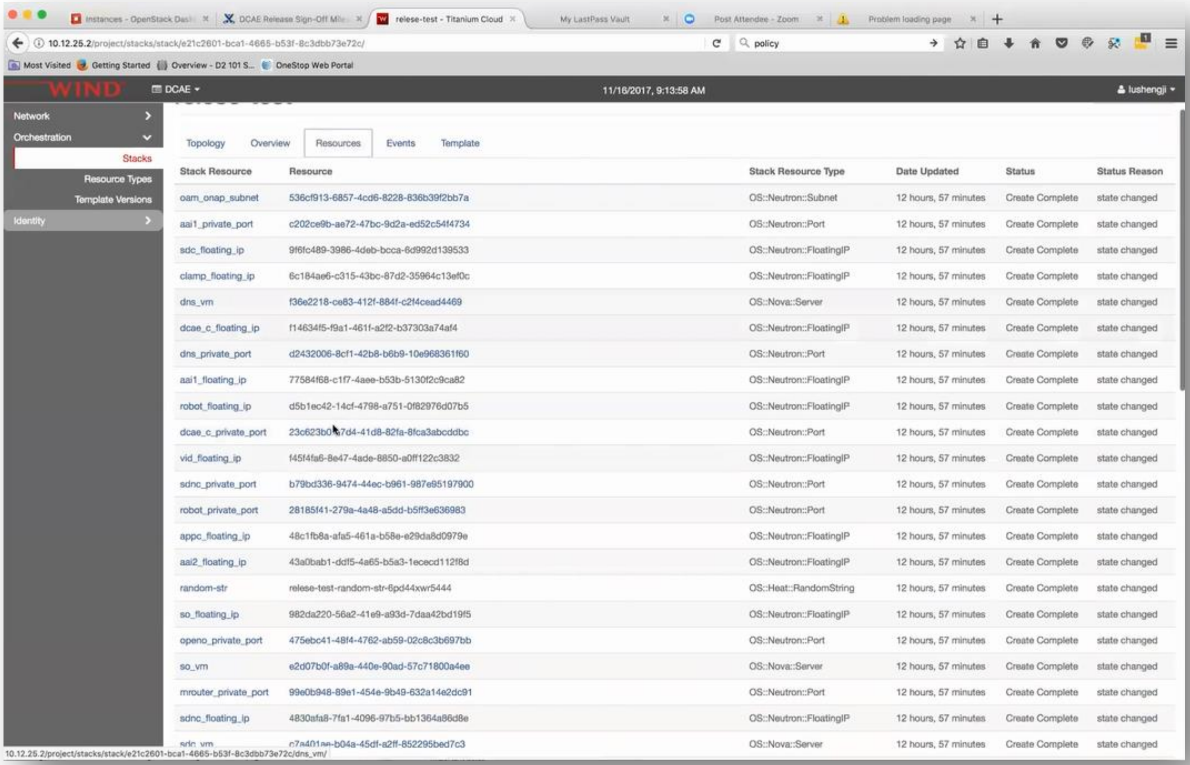


Stacks

	Stack Name	Created	Updated	Status	Actions
☐	relese-test	12 hours, 56 minutes	12 hours, 56 minutes	Create Complete	Check Stack
☐	Eddy2	1 month, 3 weeks	1 month, 3 weeks	Create Complete	Check Stack
☐	Eddy1	1 month, 3 weeks	1 month, 3 weeks	Create Complete	Check Stack

Displaying 3 items

all kind of probes, VMs.



10.12.25.2/project/stacks/stack/e21c2601-bca1-4665-b53f-8c3dbb73e72c/

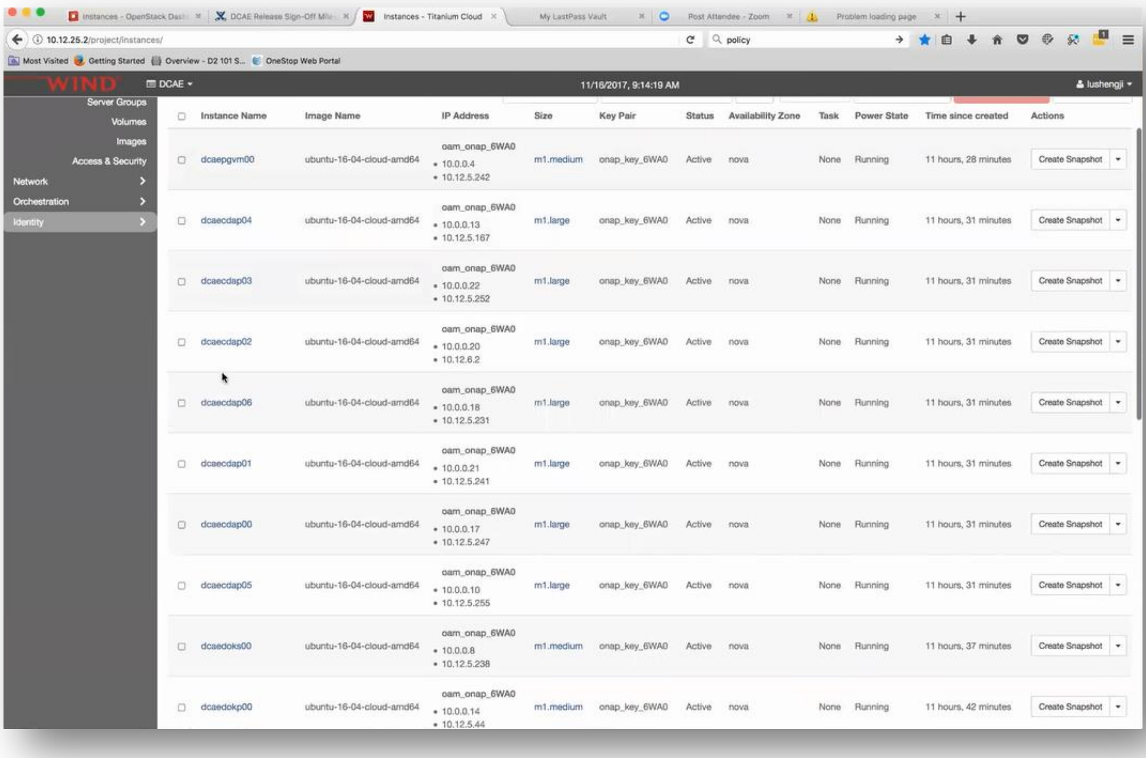
DCAE — 11/16/2017, 9:13:58 AM — lushengji

Topology | Overview | Resources | Events | Template

Stack Resource	Resource	Stack Resource Type	Date Updated	Status	Status Reason
oam_onap_subnet	536cf913-6857-4cd6-8228-836b39f2bb7a	OS::Neutron::Subnet	12 hours, 57 minutes	Create Complete	state changed
aai1_private_port	c202ce9b-ae72-47bc-9d2a-ed52c54f4734	OS::Neutron::Port	12 hours, 57 minutes	Create Complete	state changed
sdc_floating_ip	9f6fc489-3986-4deb-bcca-6d992d139533	OS::Neutron::FloatingIP	12 hours, 57 minutes	Create Complete	state changed
clamp_floating_ip	6c184ae6-c315-43bc-87d2-35964c13ef0c	OS::Neutron::FloatingIP	12 hours, 57 minutes	Create Complete	state changed
dns_vm	f36e2218-ce83-412f-884f-c2f4cead4469	OS::Nova::Server	12 hours, 57 minutes	Create Complete	state changed
dcae_c_floating_ip	f14634f5-f9a1-461f-a2f2-b37303a74af4	OS::Neutron::FloatingIP	12 hours, 57 minutes	Create Complete	state changed
dns_private_port	d2432006-8cf1-42b8-b6b9-10e968361f60	OS::Neutron::Port	12 hours, 57 minutes	Create Complete	state changed
aai1_floating_ip	77584f68-c1f7-4aee-b53b-5130f2c9ca82	OS::Neutron::FloatingIP	12 hours, 57 minutes	Create Complete	state changed
robot_floating_ip	d5b1ec42-14cf-4798-a751-0f82976d07b5	OS::Neutron::FloatingIP	12 hours, 57 minutes	Create Complete	state changed
dcae_c_private_port	23c623b0[illegible]7d4-41d8-82fa-8fca3abcddbc	OS::Neutron::Port	12 hours, 57 minutes	Create Complete	state changed
vid_floating_ip	f45f4fa6-8e47-4ade-8850-a0ff122c3832	OS::Neutron::FloatingIP	12 hours, 57 minutes	Create Complete	state changed
sdnc_private_port	b79bd336-9474-44ec-b961-987e95197900	OS::Neutron::Port	12 hours, 57 minutes	Create Complete	state changed
robot_private_port	28185f41-279a-4a48-a5dd-b5ff3e636983	OS::Neutron::Port	12 hours, 57 minutes	Create Complete	state changed
appc_floating_ip	48c1fb8a-afa5-461a-b58e-e29da8d0979e	OS::Neutron::FloatingIP	12 hours, 57 minutes	Create Complete	state changed
aai2_floating_ip	43a0bab1-ddf5-4a65-b5a3-1ececd112f8d	OS::Neutron::FloatingIP	12 hours, 57 minutes	Create Complete	state changed
random-str	relese-test-random-str-6pd44xwr5444	OS::Heat::RandomString	12 hours, 57 minutes	Create Complete	state changed
so_floating_ip	982da220-56a2-41e9-a93d-7daa42bd19f5	OS::Neutron::FloatingIP	12 hours, 57 minutes	Create Complete	state changed
openo_private_port	475ebc41-48f4-4762-ab59-02c8c3b697bb	OS::Neutron::Port	12 hours, 57 minutes	Create Complete	state changed
so_vm	e2d07b0f-a89a-440e-90ad-57c71800a4ee	OS::Nova::Server	12 hours, 57 minutes	Create Complete	state changed
mrouter_private_port	99e0b948-89e1-454e-9b49-632a14e2dc91	OS::Neutron::Port	12 hours, 57 minutes	Create Complete	state changed
sdnc_floating_ip	4830afa8-7fa1-4096-97b5-bb1364a86d8e	OS::Neutron::FloatingIP	12 hours, 57 minutes	Create Complete	state changed
sdc_vm	c7a401ae-b04a-45df-a2ff-852295bed7c3	OS::Nova::Server	12 hours, 57 minutes	Create Complete	state changed

10.12.25.2/project/stacks/stack/e21c2601-bca1-4665-b53f-8c3dbb73e72c/dns_vm/

Let’s go back to VM view (under instances) – the top ones are all DCAE VMs

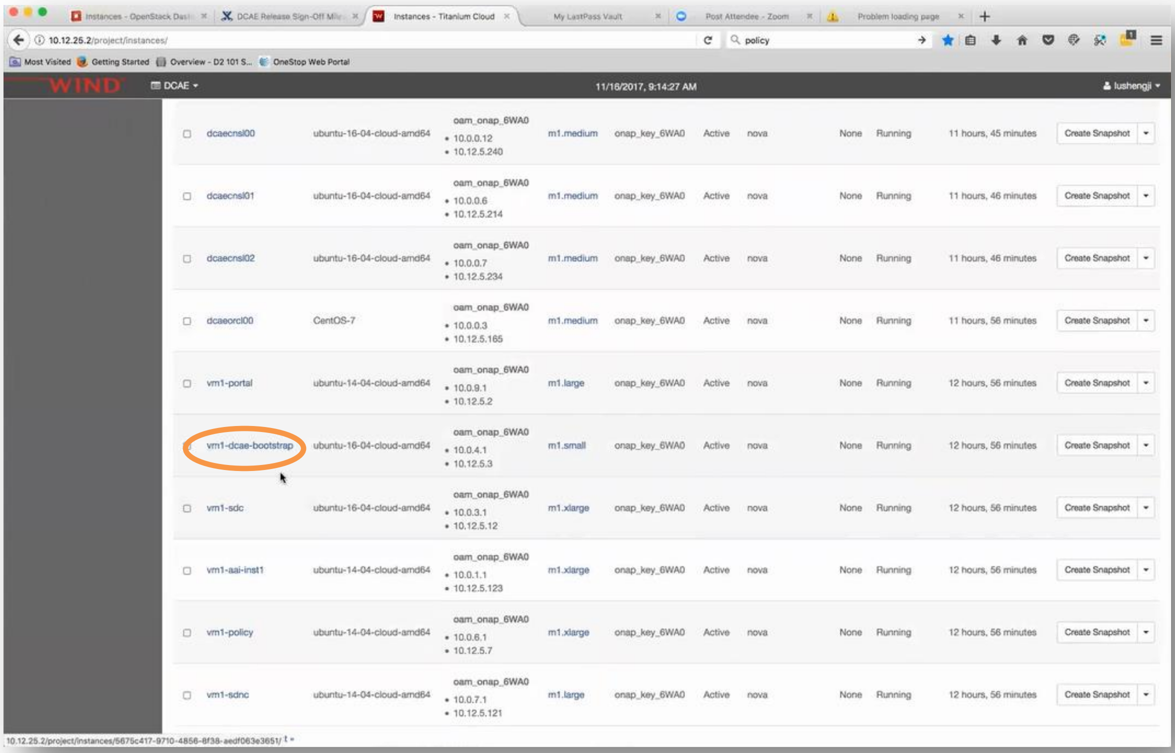


10.12.25.2/project/instances/

DCAE — 11/16/2017, 9:14:19 AM — lushengji

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
dcaepgvm00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.4 • 10.12.5.242	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 28 minutes	Create Snapshot
dcaecdap04	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.13 • 10.12.5.167	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 31 minutes	Create Snapshot
dcaecdap03	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.22 • 10.12.5.252	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 31 minutes	Create Snapshot
dcaecdap02	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.20 • 10.12.6.2	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 31 minutes	Create Snapshot
dcaecdap06	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.18 • 10.12.5.231	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 31 minutes	Create Snapshot
dcaecdap01	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.21 • 10.12.5.241	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 31 minutes	Create Snapshot
dcaecdap00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.17 • 10.12.5.247	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 31 minutes	Create Snapshot
dcaecdap05	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.10 • 10.12.5.255	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 31 minutes	Create Snapshot
dcaedoks00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.8 • 10.12.5.238	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 37 minutes	Create Snapshot
dcaedokp00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.14 • 10.12.5.44	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 42 minutes	Create Snapshot

span up by Bootstrap VM,

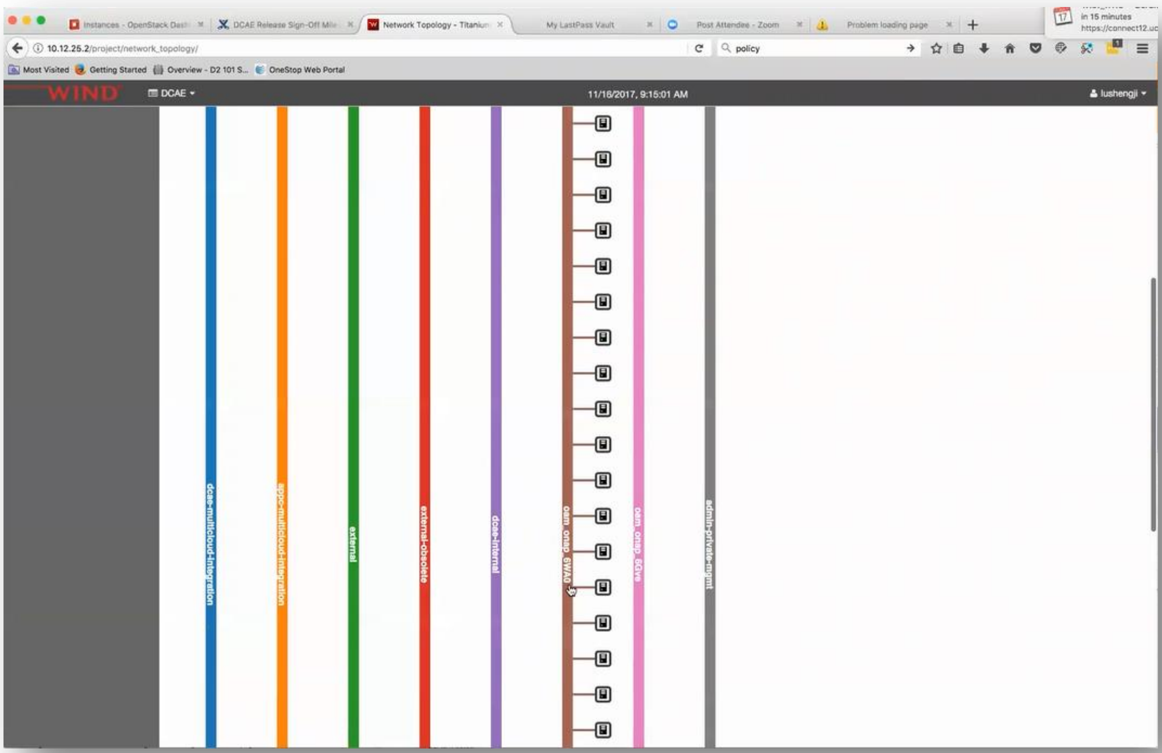

10.12.25.2/project/instances/

WIND · DCAE · 11/16/2017, 9:14:27 AM · lushengji

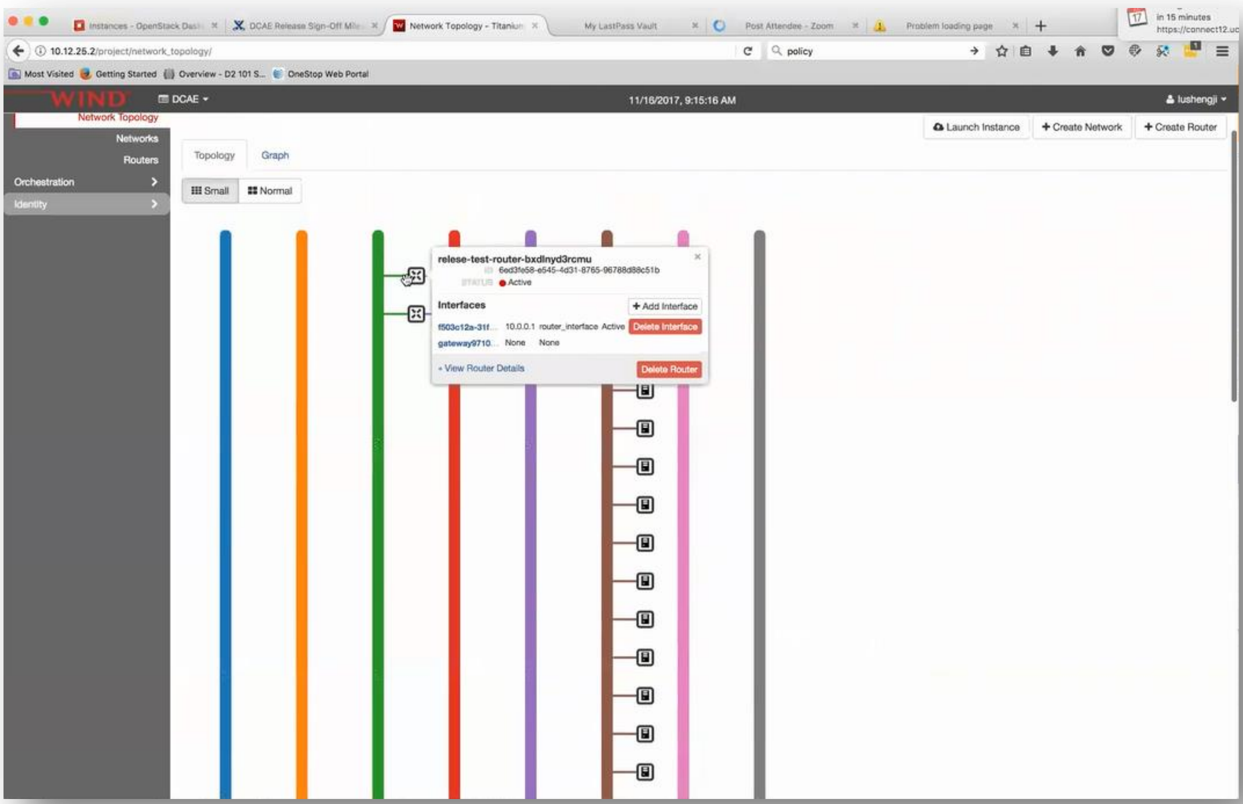
dcaecnsl00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.12 • 10.12.5.240	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 45 minutes	Create Snapshot
dcaecnsl01	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.6 • 10.12.5.214	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 46 minutes	Create Snapshot
dcaecnsl02	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.7 • 10.12.5.234	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 46 minutes	Create Snapshot
dcaeorcl00	CentOS-7	oam_onap_6WA0 • 10.0.0.3 • 10.12.5.165	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 56 minutes	Create Snapshot
vm1-portal	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.9.1 • 10.12.5.2	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 56 minutes	Create Snapshot
vm1-dcae-bootstrap	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.4.1 • 10.12.5.3	m1.small	onap_key_6WA0	Active	nova	None	Running	12 hours, 56 minutes	Create Snapshot
vm1-sdc	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.3.1 • 10.12.5.12	m1.xlarge	onap_key_6WA0	Active	nova	None	Running	12 hours, 56 minutes	Create Snapshot
vm1-aai-inst1	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.1.1 • 10.12.5.123	m1.xlarge	onap_key_6WA0	Active	nova	None	Running	12 hours, 56 minutes	Create Snapshot
vm1-policy	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.6.1 • 10.12.5.7	m1.xlarge	onap_key_6WA0	Active	nova	None	Running	12 hours, 56 minutes	Create Snapshot
vm1-sdnc	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.7.1 • 10.12.5.121	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 56 minutes	Create Snapshot

10.12.25.2/project/instances/5675c417-9710-4856-8f38-aedf063e3651/

the rest of them are ONAP VMs started by the heat template, you can see app-c, VNs, message router, there are all here, the networking of the system is such that we have the heat template will create the network – this is a pre-deployment network called OOM ONAP (random string here),



 it is gonna create and launch all the VMs described in a heat template attached to this network: for example this vm1 app-c, clamp, nso all attached to this network and the heat will also add the virtual router



relese-test-router-bxdlnyd3rcmu
ID 6ed3fe58-e545-4d31-8765-96788d88c51b
STATUS ● Active

Interfaces

f503c12a-31f...	10.0.0.1	router_interface	Active
gateway9710...	None	None	

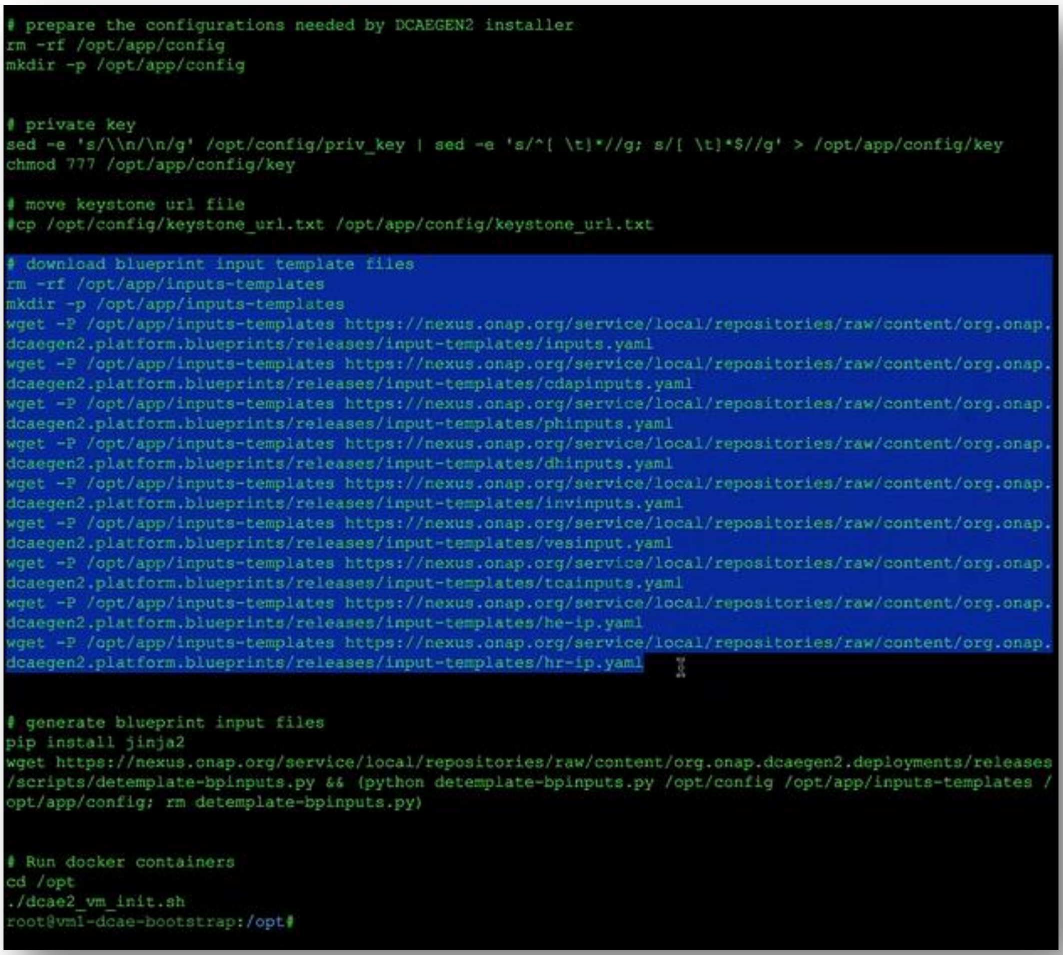
to connect this network to the external network so all the VMs on this network they can talk to the rest of the world and DCAE will launch its VMs in the same network so this way the intercommunication between all the ONAP VMs they can just use private addresses (it is described in the heat template) and also we have configured DNS services , so they can call to each other using host name as well. I will add more details regarding the DNS because it is a little bit complicated.

We can take a look at the details of how DCAE is started and can also show how to debug and check the status. This is the VM where the DCAE are from (vm1-dcae-bootstrap). I am connecting into that VM

ssh –i ~/.ssh/id_onap_dev ubuntu@10.12.5.3

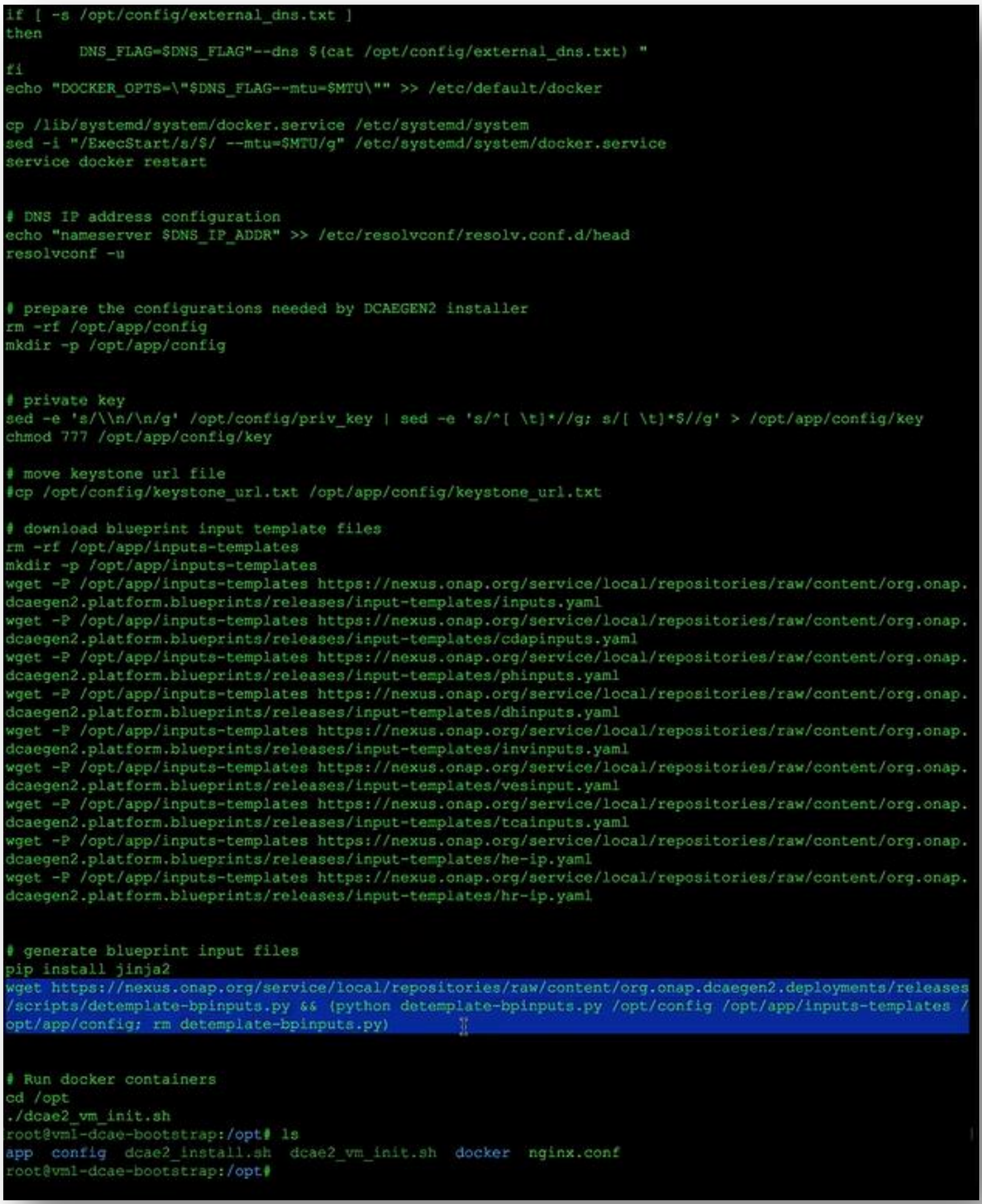
and under the opt you can see 2 scripts: dcae2_install.sh and dcae_vm_init.sh.

Those are scripts provided under the demo project. If you go to demo boot you will find those demo scripts. And the first dcae2_install.sh is where the cloud init of this VM it is going call this script it is going to install the SW, configure the DNS and for DCAE it is going to download all those input files for the blueprints,



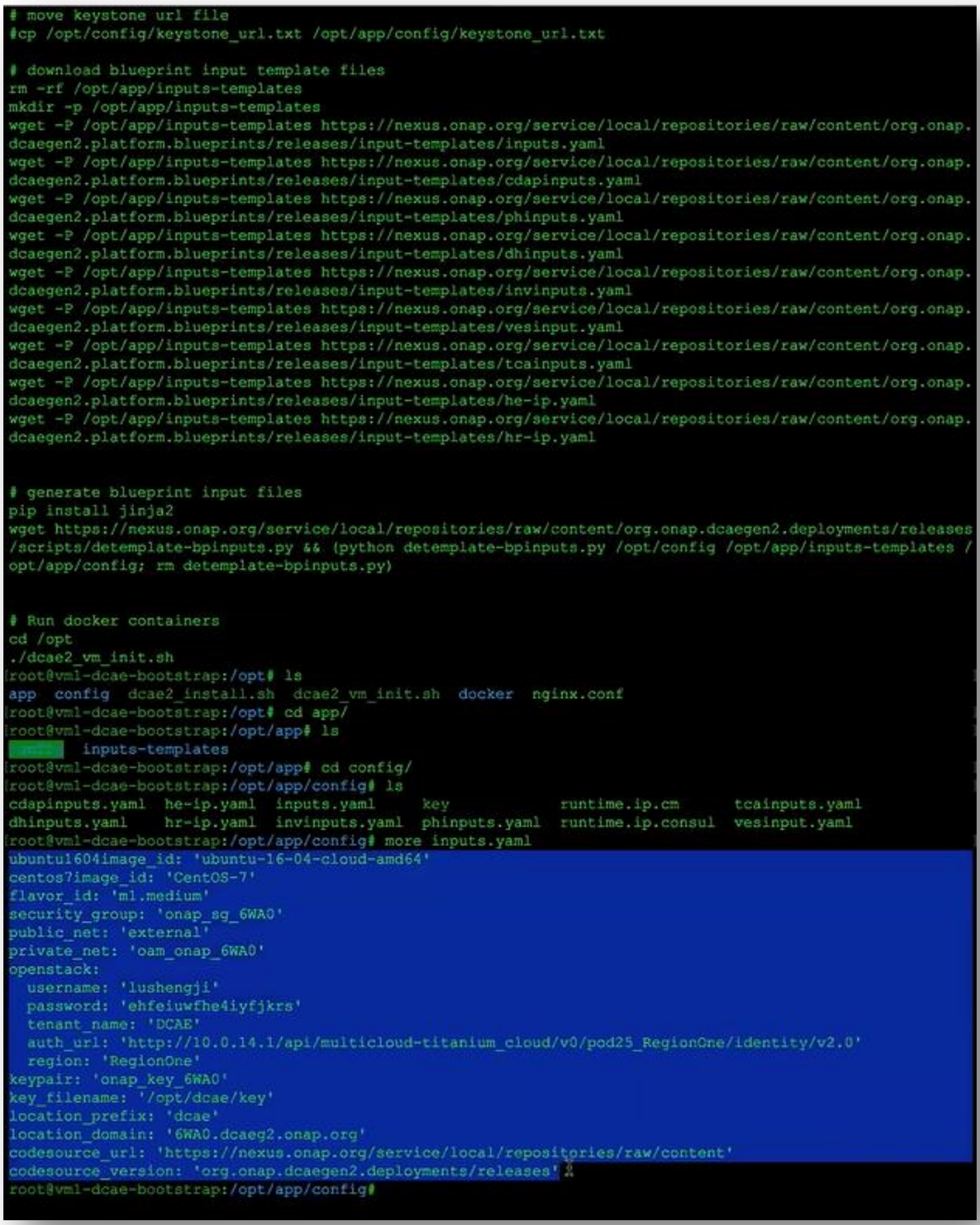
```
# prepare the configurations needed by DCAEGEN2 installer
rm -rf /opt/app/config
mkdir -p /opt/app/config


# private key
sed -e 's/\\n/\n/g' /opt/config/priv_key | sed -e 's/^[ \t]*//g; s/[ \t]*$//g' > /opt/app/config/key
chmod 777 /opt/app/config/key

# move keystone url file
#cp /opt/config/keystone_url.txt /opt/app/config/keystone_url.txt

# download blueprint input template files
rm -rf /opt/app/inputs-templates
mkdir -p /opt/app/inputs-templates
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/inputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/cdapinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/phinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/dhinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/invinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/vesinput.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/tcainputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/he-ip.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/hr-ip.yaml


# generate blueprint input files
pip install jinja2
wget https://nexus.onap.org/service/local/repositories/raw/content/org.onap.dcaegen2.deployments/releases
/scripts/detemplate-bpinputs.py && (python detemplate-bpinputs.py /opt/config /opt/app/inputs-templates /
opt/app/config; rm detemplate-bpinputs.py)


# Run docker containers
cd /opt
./dcae2_vm_init.sh
root@vm1-dcae-bootstrap:/opt#
```

those input files contain templated parameters because for example we need to know where is the keystone URL is all those it is only known at the deployment time, so those files are downloaded and detempletized



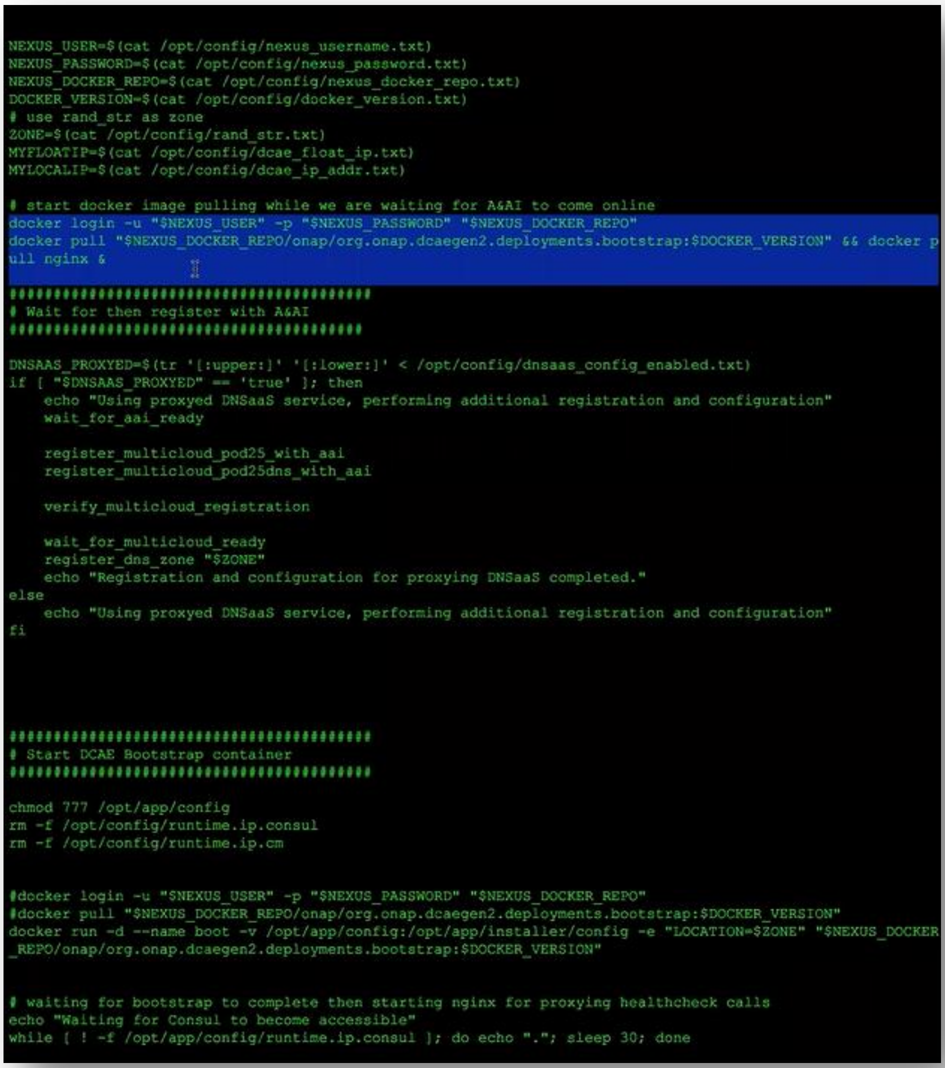
```
if [ -s /opt/config/external_dns.txt ]
then
        DNS_FLAG=$DNS_FLAG"--dns $(cat /opt/config/external_dns.txt) "
fi
echo "DOCKER_OPTS=\"$DNS_FLAG--mtu=$MTU\"" >> /etc/default/docker

cp /lib/systemd/system/docker.service /etc/systemd/system
sed -i "/ExecStart/s/$/ --mtu=$MTU/g" /etc/systemd/system/docker.service
service docker restart


# DNS IP address configuration
echo "nameserver $DNS_IP_ADDR" >> /etc/resolvconf/resolv.conf.d/head
resolvconf -u


# prepare the configurations needed by DCAEGEN2 installer
rm -rf /opt/app/config
mkdir -p /opt/app/config


# private key
sed -e 's/\\n/\n/g' /opt/config/priv_key | sed -e 's/^[ \t]*//g; s/[ \t]*$//g' > /opt/app/config/key
chmod 777 /opt/app/config/key

# move keystone url file
#cp /opt/config/keystone_url.txt /opt/app/config/keystone_url.txt

# download blueprint input template files
rm -rf /opt/app/inputs-templates
mkdir -p /opt/app/inputs-templates
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/inputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/cdapinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/phinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/dhinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/invinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/vesinput.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/tcainputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/he-ip.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/hr-ip.yaml


# generate blueprint input files
pip install jinja2
wget https://nexus.onap.org/service/local/repositories/raw/content/org.onap.dcaegen2.deployments/releases
/scripts/detemplate-bpinputs.py && (python detemplate-bpinputs.py /opt/config /opt/app/inputs-templates /
opt/app/config; rm detemplate-bpinputs.py)


# Run docker containers
cd /opt
./dcae2_vm_init.sh
root@vm1-dcae-bootstrap:/opt# ls
app  config  dcae2_install.sh  dcae2_vm_init.sh  docker  nginx.conf
root@vm1-dcae-bootstrap:/opt#
```

that is the jinja2 step for detempletizing that, we can take a look for example in the inputs.yaml file that is going to contain all those values which are localized for that particular ONAP deployment.



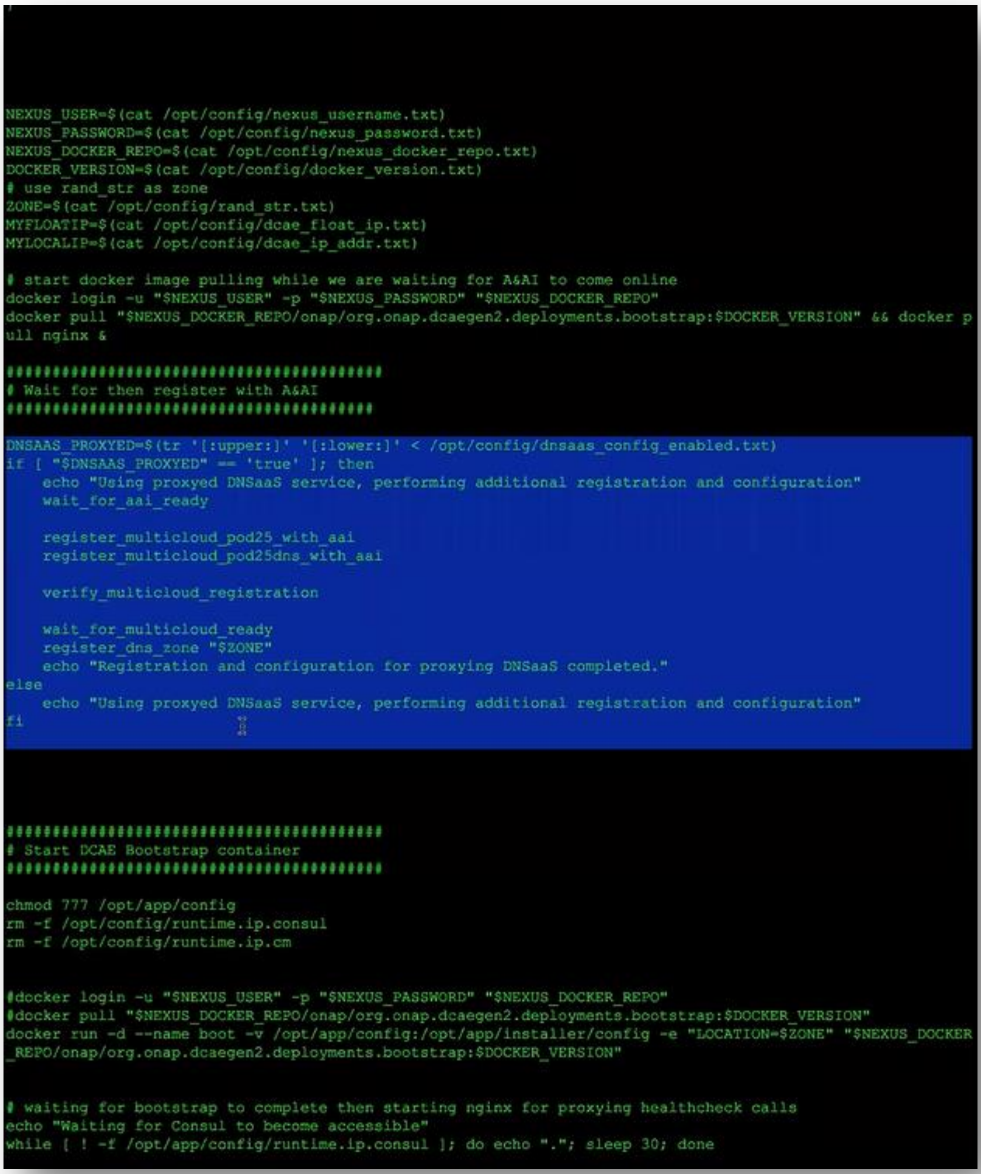
```
# move keystone url file
#cp /opt/config/keystone_url.txt /opt/app/config/keystone_url.txt

# download blueprint input template files
rm -rf /opt/app/inputs-templates
mkdir -p /opt/app/inputs-templates
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/inputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/cdapinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/phinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/dhinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/invinputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/vesinput.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/tcainputs.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/he-ip.yaml
wget -P /opt/app/inputs-templates https://nexus.onap.org/service/local/repositories/raw/content/org.onap.
dcaegen2.platform.blueprints/releases/input-templates/hr-ip.yaml


# generate blueprint input files
pip install jinja2
wget https://nexus.onap.org/service/local/repositories/raw/content/org.onap.dcaegen2.deployments/releases
/scripts/detemplate-bpinputs.py && (python detemplate-bpinputs.py /opt/config /opt/app/inputs-templates /
opt/app/config; rm detemplate-bpinputs.py)


# Run docker containers
cd /opt
./dcae2_vm_init.sh
root@vm1-dcae-bootstrap:/opt# ls
app  config  dcae2_install.sh  dcae2_vm_init.sh  docker  nginx.conf
root@vm1-dcae-bootstrap:/opt# cd app/
root@vm1-dcae-bootstrap:/opt/app# ls
[illegible]  inputs-templates
root@vm1-dcae-bootstrap:/opt/app# cd config/
root@vm1-dcae-bootstrap:/opt/app/config# ls
cdapinputs.yaml  he-ip.yaml  inputs.yaml     key             runtime.ip.cm      tcainputs.yaml
dhinputs.yaml    hr-ip.yaml  invinputs.yaml  phinputs.yaml   runtime.ip.consul  vesinput.yaml
root@vm1-dcae-bootstrap:/opt/app/config# more inputs.yaml
ubuntu1604image_id: 'ubuntu-16-04-cloud-amd64'
centos7image_id: 'CentOS-7'
flavor_id: 'm1.medium'
security_group: 'onap_sg_6WA0'
public_net: 'external'
private_net: 'oam_onap_6WA0'
openstack:
  username: 'lushengji'
  password: 'ehfeiuwfhe4iyfjkrs'
  tenant_name: 'DCAE'
  auth_url: 'http://10.0.14.1/api/multicloud-titanium_cloud/v0/pod25_RegionOne/identity/v2.0'
  region: 'RegionOne'
keypair: 'onap_key_6WA0'
key_filename: '/opt/dcae/key'
location_prefix: 'dcae'
location_domain: '6WA0.dcaeg2.onap.org'
codesource_url: 'https://nexus.onap.org/service/local/repositories/raw/content'
codesource_version: 'org.onap.dcaegen2.deployments/releases'
root@vm1-dcae-bootstrap:/opt/app/config#
```

The next step DCAE to install script gonna call is dcae_vm_init script inside here it is rather large but essentially what we do here is first we download the docker images for the bootstrap container with command:



```
NEXUS_USER=$(cat /opt/config/nexus_username.txt)
NEXUS_PASSWORD=$(cat /opt/config/nexus_password.txt)
NEXUS_DOCKER_REPO=$(cat /opt/config/nexus_docker_repo.txt)
DOCKER_VERSION=$(cat /opt/config/docker_version.txt)
# use rand_str as zone
ZONE=$(cat /opt/config/rand_str.txt)
MYFLOATIP=$(cat /opt/config/dcae_float_ip.txt)
MYLOCALIP=$(cat /opt/config/dcae_ip_addr.txt)

# start docker image pulling while we are waiting for A&AI to come online
docker login -u "$NEXUS_USER" -p "$NEXUS_PASSWORD" "$NEXUS_DOCKER_REPO"
docker pull "$NEXUS_DOCKER_REPO/onap/org.onap.dcaegen2.deployments.bootstrap:$DOCKER_VERSION" && docker p
ull nginx &

############################################
# Wait for then register with A&AI
############################################

DNSAAS_PROXYED=$(tr '[:upper:]' '[:lower:]' < /opt/config/dnsaas_config_enabled.txt)
if [ "$DNSAAS_PROXYED" == 'true' ]; then
    echo "Using proxyed DNSaaS service, performing additional registration and configuration"
    wait_for_aai_ready

    register_multicloud_pod25_with_aai
    register_multicloud_pod25dns_with_aai

    verify_multicloud_registration

    wait_for_multicloud_ready
    register_dns_zone "$ZONE"
    echo "Registration and configuration for proxying DNSaaS completed."
else
    echo "Using proxyed DNSaaS service, performing additional registration and configuration"
fi




############################################
# Start DCAE Bootstrap container
############################################

chmod 777 /opt/app/config
rm -f /opt/config/runtime.ip.consul
rm -f /opt/config/runtime.ip.cm


#docker login -u "$NEXUS_USER" -p "$NEXUS_PASSWORD" "$NEXUS_DOCKER_REPO"
#docker pull "$NEXUS_DOCKER_REPO/onap/org.onap.dcaegen2.deployments.bootstrap:$DOCKER_VERSION"
docker run -d --name boot -v /opt/app/config:/opt/app/installer/config -e "LOCATION=$ZONE" "$NEXUS_DOCKER
_REPO/onap/org.onap.dcaegen2.deployments.bootstrap:$DOCKER_VERSION"


# waiting for bootstrap to complete then starting nginx for proxying healthcheck calls
echo "Waiting for Consul to become accessible"
while [ ! -f /opt/app/config/runtime.ip.consul ]; do echo "."; sleep 30; done
```

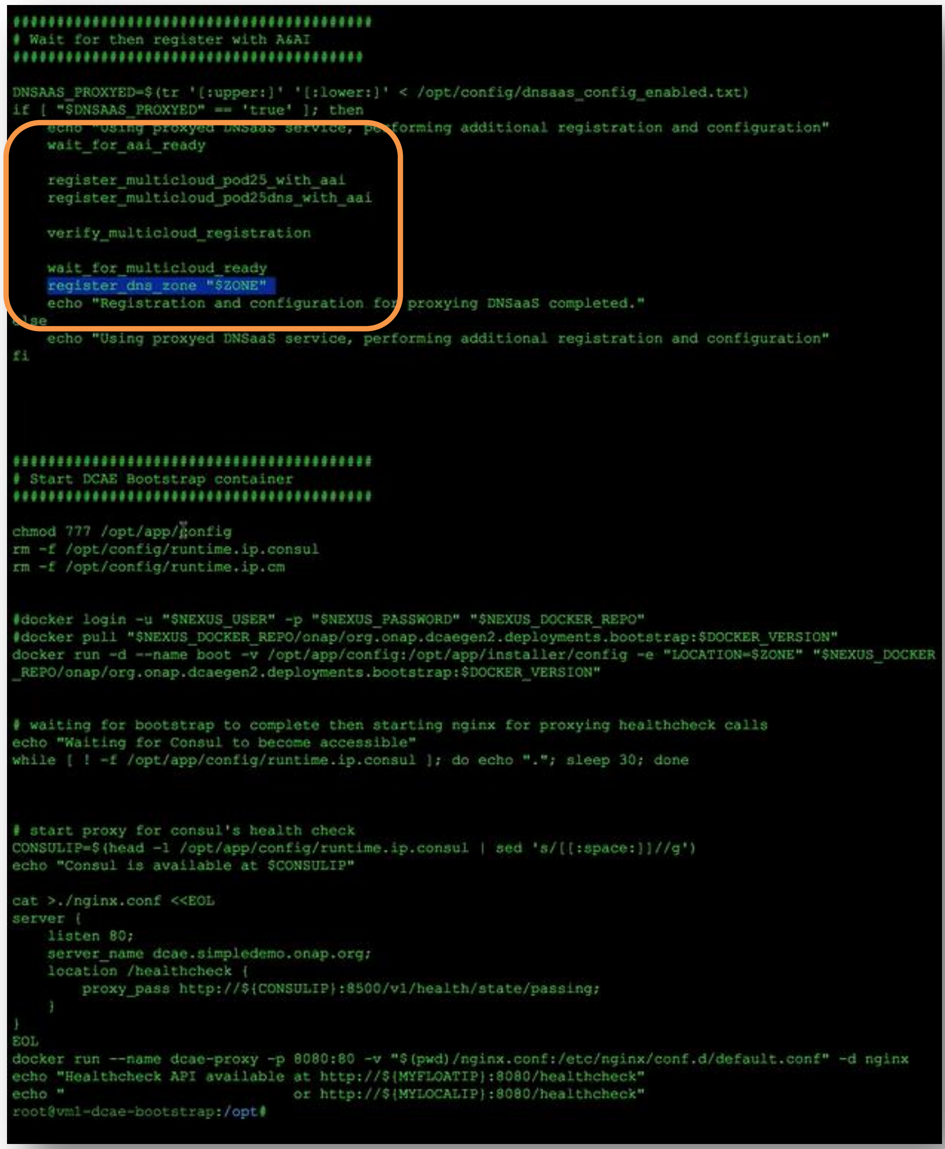
it may take some time depending on how fast your network is. While this is downloading we setup the DNS.



```
NEXUS_USER=$(cat /opt/config/nexus_username.txt)
NEXUS_PASSWORD=$(cat /opt/config/nexus_password.txt)
NEXUS_DOCKER_REPO=$(cat /opt/config/nexus_docker_repo.txt)
DOCKER_VERSION=$(cat /opt/config/docker_version.txt)
# use rand_str as zone
ZONE=$(cat /opt/config/rand_str.txt)
MYFLOATIP=$(cat /opt/config/dcae_float_ip.txt)
MYLOCALIP=$(cat /opt/config/dcae_ip_addr.txt)

# start docker image pulling while we are waiting for A&AI to come online
docker login -u "$NEXUS_USER" -p "$NEXUS_PASSWORD" "$NEXUS_DOCKER_REPO"
docker pull "$NEXUS_DOCKER_REPO/onap/org.onap.dcaegen2.deployments.bootstrap:$DOCKER_VERSION" && docker p
ull nginx &

##########################################
# Wait for then register with A&AI
#########################################

DNSAAS_PROXYED=$(tr '[:upper:]' '[:lower:]' < /opt/config/dnsaas_config_enabled.txt)
if [ "$DNSAAS_PROXYED" == 'true' ]; then
    echo "Using proxyed DNSaaS service, performing additional registration and configuration"
    wait_for_aai_ready

    register_multicloud_pod25_with_aai
    register_multicloud_pod25dns_with_aai

    verify_multicloud_registration

    wait_for_multicloud_ready
    register_dns_zone "$ZONE"
    echo "Registration and configuration for proxying DNSaaS completed."
else
    echo "Using proxyed DNSaaS service, performing additional registration and configuration"
fi



##########################################
# Start DCAE Bootstrap container
##########################################

chmod 777 /opt/app/config
rm -f /opt/config/runtime.ip.consul
rm -f /opt/config/runtime.ip.cm


#docker login -u "$NEXUS_USER" -p "$NEXUS_PASSWORD" "$NEXUS_DOCKER_REPO"
#docker pull "$NEXUS_DOCKER_REPO/onap/org.onap.dcaegen2.deployments.bootstrap:$DOCKER_VERSION"
docker run -d --name boot -v /opt/app/config:/opt/app/installer/config -e "LOCATION=$ZONE" "$NEXUS_DOCKER
_REPO/onap/org.onap.dcaegen2.deployments.bootstrap:$DOCKER_VERSION"


# waiting for bootstrap to complete then starting nginx for proxying healthcheck calls
echo "Waiting for Consul to become accessible"
while [ ! -f /opt/app/config/runtime.ip.consul ]; do echo "."; sleep 30; done
```

In the Integration Lab we use so called proxied DNS as a Service solution – the Pod25 Openstack installation does not natively have designate support instead has smaller another Openstack installation which has designate and we are really delegating all the DNS designated operations to that stack, so these are the steps needed for using this proxy DNS as a Service solution.

First we wait for A&AI become ready as the solution is using multicloud which requires A&AI to be ready so it can retrieve the information from the A&AI, so after A&AI is ready we can register those 2 blocks of information with A&AI – they will be used by multicloud to do the DNS proxy – designate proxy. Then after that we verify that the registrations are good, then we wait for multicloud to become ready, after that we register the DNS zone.

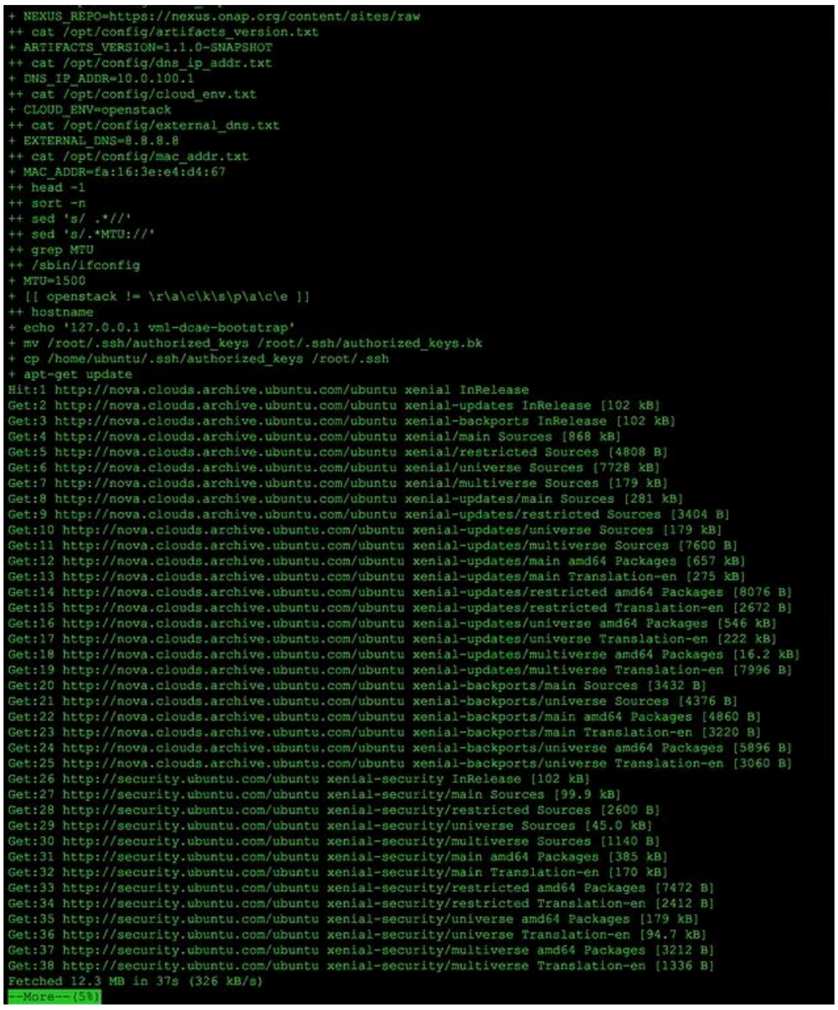


```
##########################################
# Wait for then register with A&AI
##########################################

DNSAAS_PROXYED=$(tr '[:upper:]' '[:lower:]' < /opt/config/dnsaas_config_enabled.txt)
if [ "$DNSAAS_PROXYED" == 'true' ]; then
    echo "Using proxyed DNSaaS service, performing additional registration and configuration"
    wait_for_aai_ready

    register_multicloud_pod25_with_aai
    register_multicloud_pod25dns_with_aai

    verify_multicloud_registration

    wait_for_multicloud_ready
    register_dns_zone "$ZONE"
    echo "Registration and configuration for proxying DNSaaS completed."
else
    echo "Using proxyed DNSaaS service, performing additional registration and configuration"
fi




##########################################
# Start DCAE Bootstrap container
##########################################

chmod 777 /opt/app/config
rm -f /opt/config/runtime.ip.consul
rm -f /opt/config/runtime.ip.cm


#docker login -u "$NEXUS_USER" -p "$NEXUS_PASSWORD" "$NEXUS_DOCKER_REPO"
#docker pull "$NEXUS_DOCKER_REPO/onap/org.onap.dcaegen2.deployments.bootstrap:$DOCKER_VERSION"
docker run -d --name boot -v /opt/app/config:/opt/app/installer/config -e "LOCATION=$ZONE" "$NEXUS_DOCKER
_REPO/onap/org.onap.dcaegen2.deployments.bootstrap:$DOCKER_VERSION"


# waiting for bootstrap to complete then starting nginx for proxying healthcheck calls
echo "Waiting for Consul to become accessible"
while [ ! -f /opt/app/config/runtime.ip.consul ]; do echo "."; sleep 30; done



# start proxy for consul's health check
CONSULIP=$(head -1 /opt/app/config/runtime.ip.consul | sed 's/[[:space:]]//g')
echo "Consul is available at $CONSULIP"

cat >./nginx.conf <<EOL
server {
    listen 80;
    server_name dcae.simpledemo.onap.org;
    location /healthcheck {
        proxy_pass http://${CONSULIP}:8500/v1/health/state/passing;
    }
}
EOL
docker run --name dcae-proxy -p 8080:80 -v "$(pwd)/nginx.conf:/etc/nginx/conf.d/default.conf" -d nginx
echo "Healthcheck API available at http://${MYFLOATIP}:8080/healthcheck"
echo "                                or http://${MYLOCALIP}:8080/healthcheck"
root@vm1-dcae-bootstrap:/opt#
```

So this is really for prepping DNS for the DCAE. After that we start to run the bootstrap container

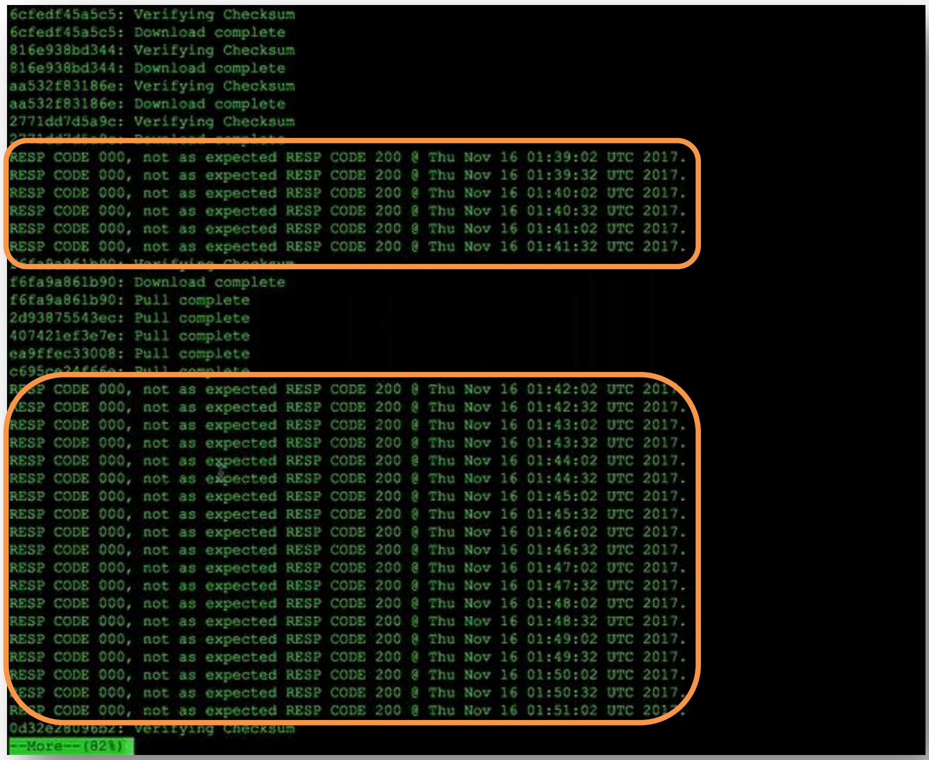
docker run –d --name boot –v /opt/app/config:/opt/app/installer/config –e "LOCATION-$ZONE" "$NEXUS_DOCKER_REPO/onap/org.onap.dcaegen2.deployments.bootstrap:$DOCKER_VERSION"

and then all the other interesting stuff happening inside of that bootstrap container, so there is a log file at tmp/dcae2_install.log where we can see that docker images were being pooled,



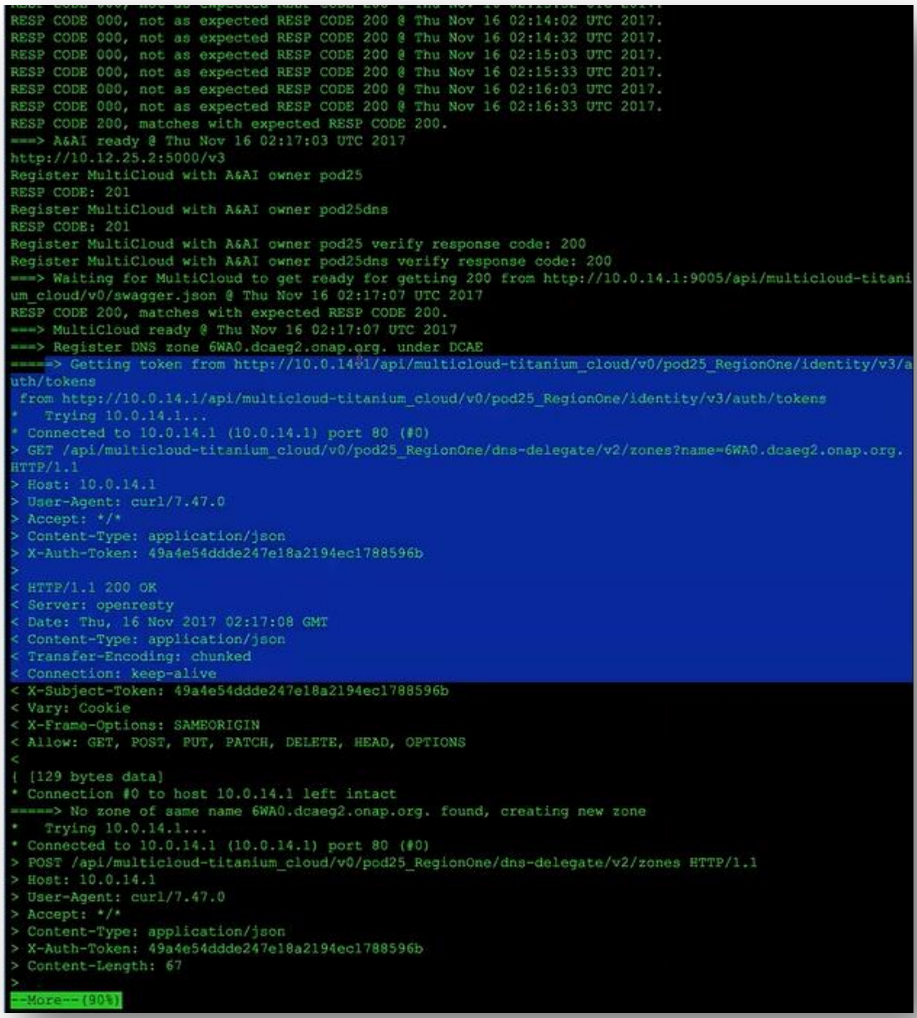
```
+ NEXUS_REPO=https://nexus.onap.org/content/sites/raw
++ cat /opt/config/artifacts_version.txt
+ ARTIFACTS_VERSION=1.1.0-SNAPSHOT
++ cat /opt/config/dns_ip_addr.txt
+ DNS_IP_ADDR=10.0.100.1
++ cat /opt/config/cloud_env.txt
+ CLOUD_ENV=openstack
++ cat /opt/config/external_dns.txt
+ EXTERNAL_DNS=8.8.8.8
++ cat /opt/config/mac_addr.txt
+ MAC_ADDR=fa:16:3e:e4:d4:67
++ head -1
++ sort -n
++ sed 's/ .*//'
++ sed 's/.*MTU://'
++ grep MTU
++ /sbin/ifconfig
+ MTU=1500
+ [[ openstack != \r\a\c\k\s\p\a\c\e ]]
++ hostname
+ echo '127.0.0.1 vm1-dcae-bootstrap'
+ mv /root/.ssh/authorized_keys /root/.ssh/authorized_keys.bk
+ cp /home/ubuntu/.ssh/authorized_keys /root/.ssh
+ apt-get update
Hit:1 http://nova.clouds.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Get:3 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Get:4 http://nova.clouds.archive.ubuntu.com/ubuntu xenial/main Sources [868 kB]
Get:5 http://nova.clouds.archive.ubuntu.com/ubuntu xenial/restricted Sources [4808 B]
Get:6 http://nova.clouds.archive.ubuntu.com/ubuntu xenial/universe Sources [7728 kB]
Get:7 http://nova.clouds.archive.ubuntu.com/ubuntu xenial/multiverse Sources [179 kB]
Get:8 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/main Sources [281 kB]
Get:9 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/restricted Sources [3404 B]
Get:10 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/universe Sources [179 kB]
Get:11 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/multiverse Sources [7600 B]
Get:12 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages [657 kB]
Get:13 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/main Translation-en [275 kB]
Get:14 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/restricted amd64 Packages [8076 B]
Get:15 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/restricted Translation-en [2672 B]
Get:16 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/universe amd64 Packages [546 kB]
Get:17 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/universe Translation-en [222 kB]
Get:18 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/multiverse amd64 Packages [16.2 kB]
Get:19 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-updates/multiverse Translation-en [7996 B]
Get:20 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-backports/main Sources [3432 B]
Get:21 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-backports/universe Sources [4376 B]
Get:22 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-backports/main amd64 Packages [4860 B]
Get:23 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-backports/main Translation-en [3220 B]
Get:24 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-backports/universe amd64 Packages [5896 B]
Get:25 http://nova.clouds.archive.ubuntu.com/ubuntu xenial-backports/universe Translation-en [3060 B]
Get:26 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Get:27 http://security.ubuntu.com/ubuntu xenial-security/main Sources [99.9 kB]
Get:28 http://security.ubuntu.com/ubuntu xenial-security/restricted Sources [2600 B]
Get:29 http://security.ubuntu.com/ubuntu xenial-security/universe Sources [45.0 kB]
Get:30 http://security.ubuntu.com/ubuntu xenial-security/multiverse Sources [1140 B]
Get:31 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [385 kB]
Get:32 http://security.ubuntu.com/ubuntu xenial-security/main Translation-en [170 kB]
Get:33 http://security.ubuntu.com/ubuntu xenial-security/restricted amd64 Packages [7472 B]
Get:34 http://security.ubuntu.com/ubuntu xenial-security/restricted Translation-en [2412 B]
Get:35 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages [179 kB]
Get:36 http://security.ubuntu.com/ubuntu xenial-security/universe Translation-en [94.7 kB]
Get:37 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 Packages [3212 B]
Get:38 http://security.ubuntu.com/ubuntu xenial-security/multiverse Translation-en [1336 B]
Fetched 12.3 MB in 37s (326 kB/s)
--More--(5%)
```

you can see lots of response code 000 – that is a step we are waiting for A&AI to become ready



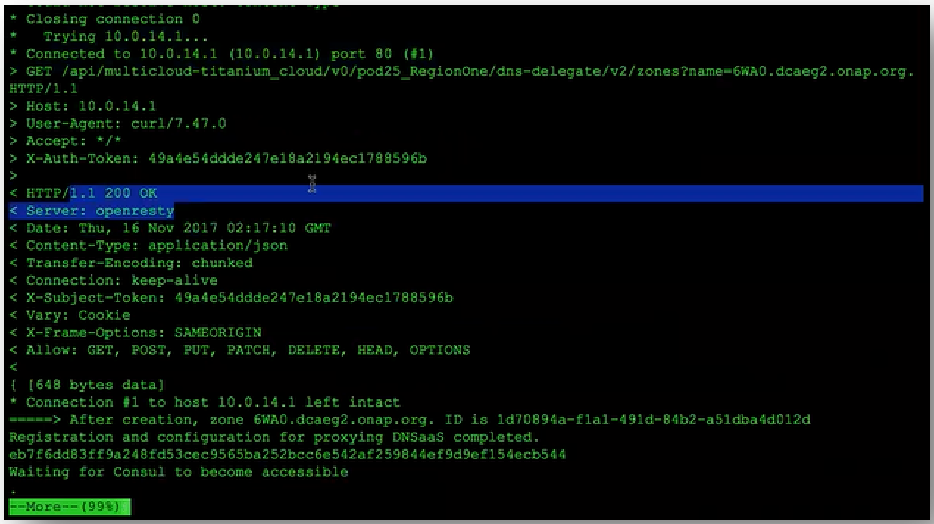
```
6cfedf45a5c5: Verifying Checksum
6cfedf45a5c5: Download complete
816e938bd344: Verifying Checksum
816e938bd344: Download complete
aa532f83186e: Verifying Checksum
aa532f83186e: Download complete
2771dd7d5a9c: Verifying Checksum
2771dd7d5a9c: Download complete
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:39:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:39:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:40:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:40:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:41:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:41:32 UTC 2017.
f6fa9a861b90: Verifying Checksum
f6fa9a861b90: Download complete
f6fa9a861b90: Pull complete
2d93875543ec: Pull complete
407421ef3e7e: Pull complete
ea9ffec33008: Pull complete
c695ce24f66e: Pull complete
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:42:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:42:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:43:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:43:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:44:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:44:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:45:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:45:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:46:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:46:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:47:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:47:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:48:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:48:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:49:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:49:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:50:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:50:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 01:51:02 UTC 2017.
0d32e28096b2: Verifying Checksum
--More--(82%)
```

– it takes like 30 - 40 minutes for A&AI to become ready. So than after that you can see the DNS registering keystone token from the delegate designate OpenStack, then it lists all the current zones registered there,



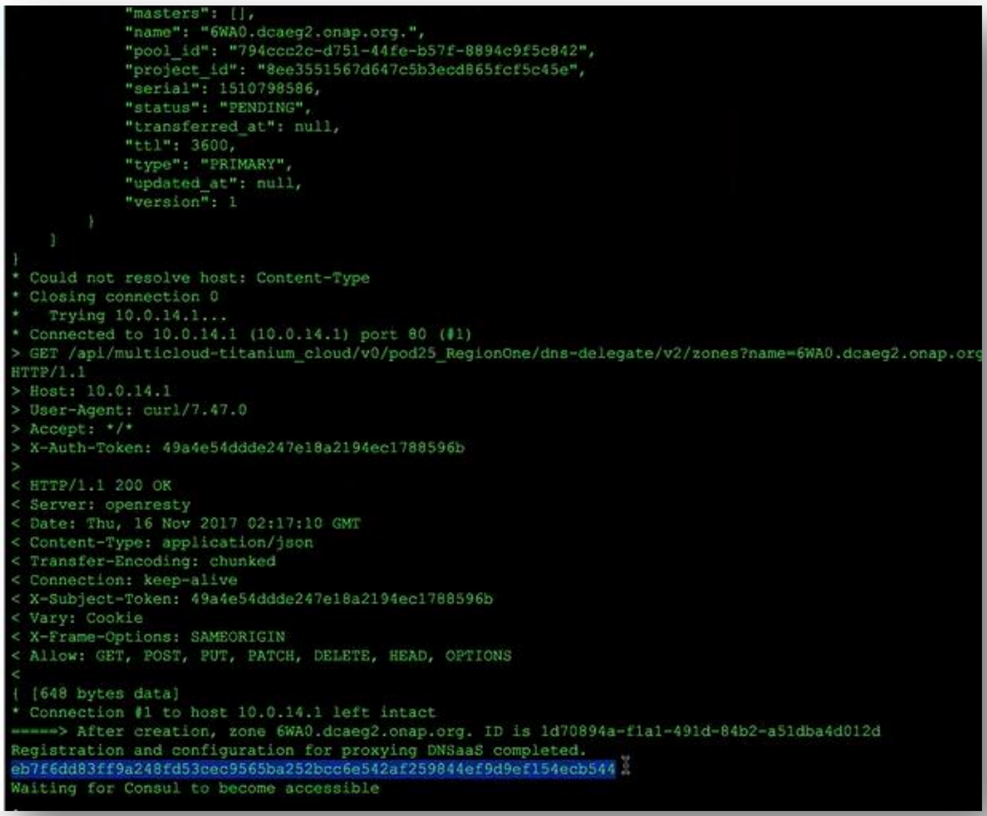
```
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 02:14:02 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 02:14:32 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 02:15:03 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 02:15:33 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 02:16:03 UTC 2017.
RESP CODE 000, not as expected RESP CODE 200 @ Thu Nov 16 02:16:33 UTC 2017.
RESP CODE 200, matches with expected RESP CODE 200.
===> A&AI ready @ Thu Nov 16 02:17:03 UTC 2017
http://10.12.25.2:5000/v3
Register MultiCloud with A&AI owner pod25
RESP CODE: 201
Register MultiCloud with A&AI owner pod25dns
RESP CODE: 201
Register MultiCloud with A&AI owner pod25 verify response code: 200
Register MultiCloud with A&AI owner pod25dns verify response code: 200
===> Waiting for MultiCloud to get ready for getting 200 from http://10.0.14.1:9005/api/multicloud-titani
um_cloud/v0/swagger.json @ Thu Nov 16 02:17:07 UTC 2017
RESP CODE 200, matches with expected RESP CODE 200.
===> MultiCloud ready @ Thu Nov 16 02:17:07 UTC 2017
===> Register DNS zone 6WA0.dcaeg2.onap.org. under DCAE
=====> Getting token from http://10.0.14.1/api/multicloud-titanium_cloud/v0/pod25_RegionOne/identity/v3/a
uth/tokens
 from http://10.0.14.1/api/multicloud-titanium_cloud/v0/pod25_RegionOne/identity/v3/auth/tokens
*   Trying 10.0.14.1...
* Connected to 10.0.14.1 (10.0.14.1) port 80 (#0)
> GET /api/multicloud-titanium_cloud/v0/pod25_RegionOne/dns-delegate/v2/zones?name=6WA0.dcaeg2.onap.org.
HTTP/1.1
> Host: 10.0.14.1
> User-Agent: curl/7.47.0
> Accept: */*
> Content-Type: application/json
> X-Auth-Token: 49a4e54ddde247e18a2194ec1788596b
>
< HTTP/1.1 200 OK
< Server: openresty
< Date: Thu, 16 Nov 2017 02:17:08 GMT
< Content-Type: application/json
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Subject-Token: 49a4e54ddde247e18a2194ec1788596b
< Vary: Cookie
< X-Frame-Options: SAMEORIGIN
< Allow: GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS
<
{ [129 bytes data]
* Connection #0 to host 10.0.14.1 left intact
=====> No zone of same name 6WA0.dcaeg2.onap.org. found, creating new zone
*   Trying 10.0.14.1...
* Connected to 10.0.14.1 (10.0.14.1) port 80 (#0)
> POST /api/multicloud-titanium_cloud/v0/pod25_RegionOne/dns-delegate/v2/zones HTTP/1.1
> Host: 10.0.14.1
> User-Agent: curl/7.47.0
> Accept: */*
> Content-Type: application/json
> X-Auth-Token: 49a4e54ddde247e18a2194ec1788596b
> Content-Length: 67
>
--More--(90%)
```

finally it will register the new zone for the DNS for this particular ONAP installation



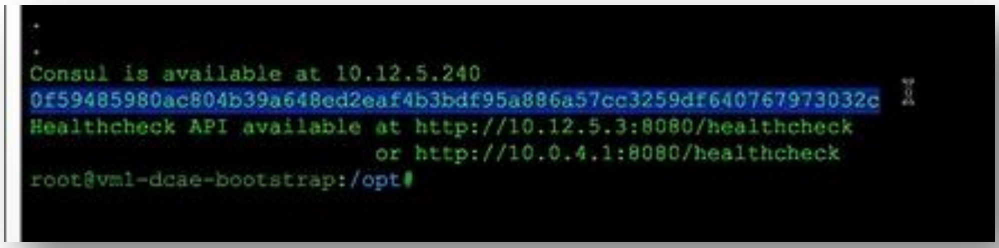
```
* Closing connection 0
*   Trying 10.0.14.1...
* Connected to 10.0.14.1 (10.0.14.1) port 80 (#1)
> GET /api/multicloud-titanium_cloud/v0/pod25_RegionOne/dns-delegate/v2/zones?name=6WA0.dcaeg2.onap.org.
HTTP/1.1
> Host: 10.0.14.1
> User-Agent: curl/7.47.0
> Accept: */*
> X-Auth-Token: 49a4e54ddde247e18a2194ec1788596b
>
< HTTP/1.1 200 OK
< Server: openresty
< Date: Thu, 16 Nov 2017 02:17:10 GMT
< Content-Type: application/json
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Subject-Token: 49a4e54ddde247e18a2194ec1788596b
< Vary: Cookie
< X-Frame-Options: SAMEORIGIN
< Allow: GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS
<
{ [648 bytes data]
* Connection #1 to host 10.0.14.1 left intact
=====> After creation, zone 6WA0.dcaeg2.onap.org. ID is 1d70894a-f1a1-491d-84b2-a51dba4d012d
Registration and configuration for proxying DNSaaS completed.
eb7f6dd83ff9a248fd53cec9565ba252bcc6e542af259844ef9d9ef154ecb544
Waiting for Consul to become accessible
.
--More--(99%)
```

then we move on here: the docker is started and it is waiting for docker to finish its job,



```
            "masters": [],
            "name": "6WA0.dcaeg2.onap.org.",
            "pool_id": "794ccc2c-d751-44fe-b57f-8894c9f5c842",
            "project_id": "8ee3551567d647c5b3ecd865fcf5c45e",
            "serial": 1510798586,
            "status": "PENDING",
            "transferred_at": null,
            "ttl": 3600,
            "type": "PRIMARY",
            "updated_at": null,
            "version": 1
        }
    ]
}
* Could not resolve host: Content-Type
* Closing connection 0
*   Trying 10.0.14.1...
* Connected to 10.0.14.1 (10.0.14.1) port 80 (#1)
> GET /api/multicloud-titanium_cloud/v0/pod25_RegionOne/dns-delegate/v2/zones?name=6WA0.dcaeg2.onap.org
HTTP/1.1
> Host: 10.0.14.1
> User-Agent: curl/7.47.0
> Accept: */*
> X-Auth-Token: 49a4e54ddde247e18a2194ec1788596b
>
< HTTP/1.1 200 OK
< Server: openresty
< Date: Thu, 16 Nov 2017 02:17:10 GMT
< Content-Type: application/json
< Transfer-Encoding: chunked
< Connection: keep-alive
< X-Subject-Token: 49a4e54ddde247e18a2194ec1788596b
< Vary: Cookie
< X-Frame-Options: SAMEORIGIN
< Allow: GET, POST, PUT, PATCH, DELETE, HEAD, OPTIONS
<
{ [648 bytes data]
* Connection #1 to host 10.0.14.1 left intact
=====> After creation, zone 6WA0.dcaeg2.onap.org. ID is 1d70894a-f1a1-491d-84b2-a51dba4d012d
Registration and configuration for proxying DNSaaS completed.
eb7f6dd83ff9a248fd53cec9565ba252bcc6e542af259844ef9d9ef154ecb544
Waiting for Consul to become accessible
```

 then it is end - the docker is done, we start the second docker image, docker container which is and Enginx which provides the health check API for the robot testing framework.

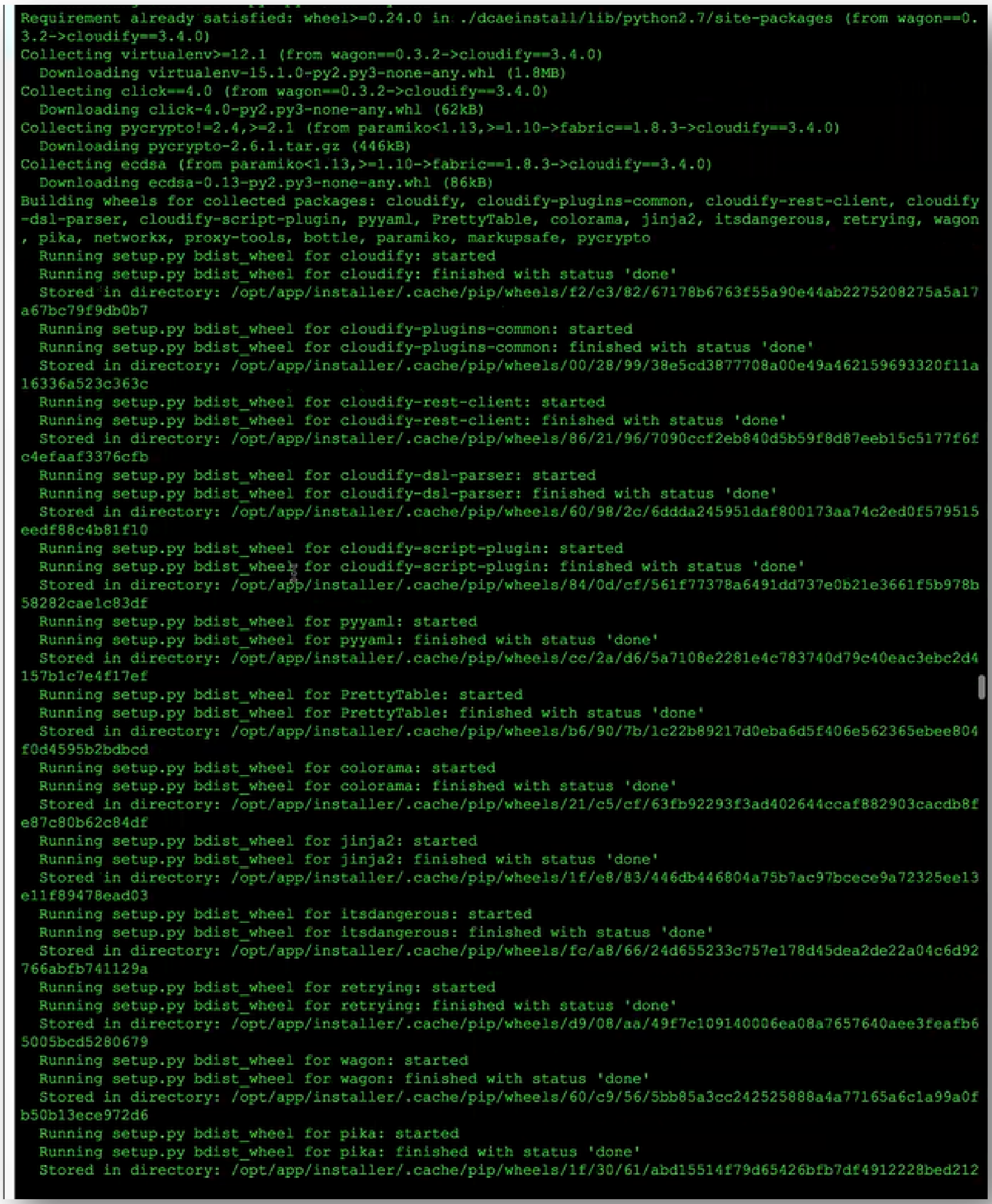


```
.
.
Consul is available at 10.12.5.240
0f59485980ac804b39a648ed2eaf4b3bdf95a886a57cc3259df640767973032c
Healthcheck API available at http://10.12.5.3:8080/healthcheck
                          or http://10.0.4.1:8080/healthcheck
root@vm1-dcae-bootstrap:/opt#
```

The reason being the robot it is problem with an IP address and URL to pool for health tests status but for DCAE because such health tests information is available at the console cluster and console cluster IP address is dynamically assigned by DHCP, so we do not know beforehand, therefore we setting up this proxy basically within this bootstrap VM at the end of deploying of DCAE we would know where the console is and we setup this Enginx proxy, so the robot can always call bootstrap VM which has a fixed static IP in the heat template. The robot can always pool this address, this URL (http://10.12.3.3:8080/healthcheck and (http://10.0.4.1:8080/healthcheck) for the health check for the whole DCAE. All those vertical dots are executions inside of the docker so you can see that there are 2 docker containers running. This is the bootstrap container so we can do logs:

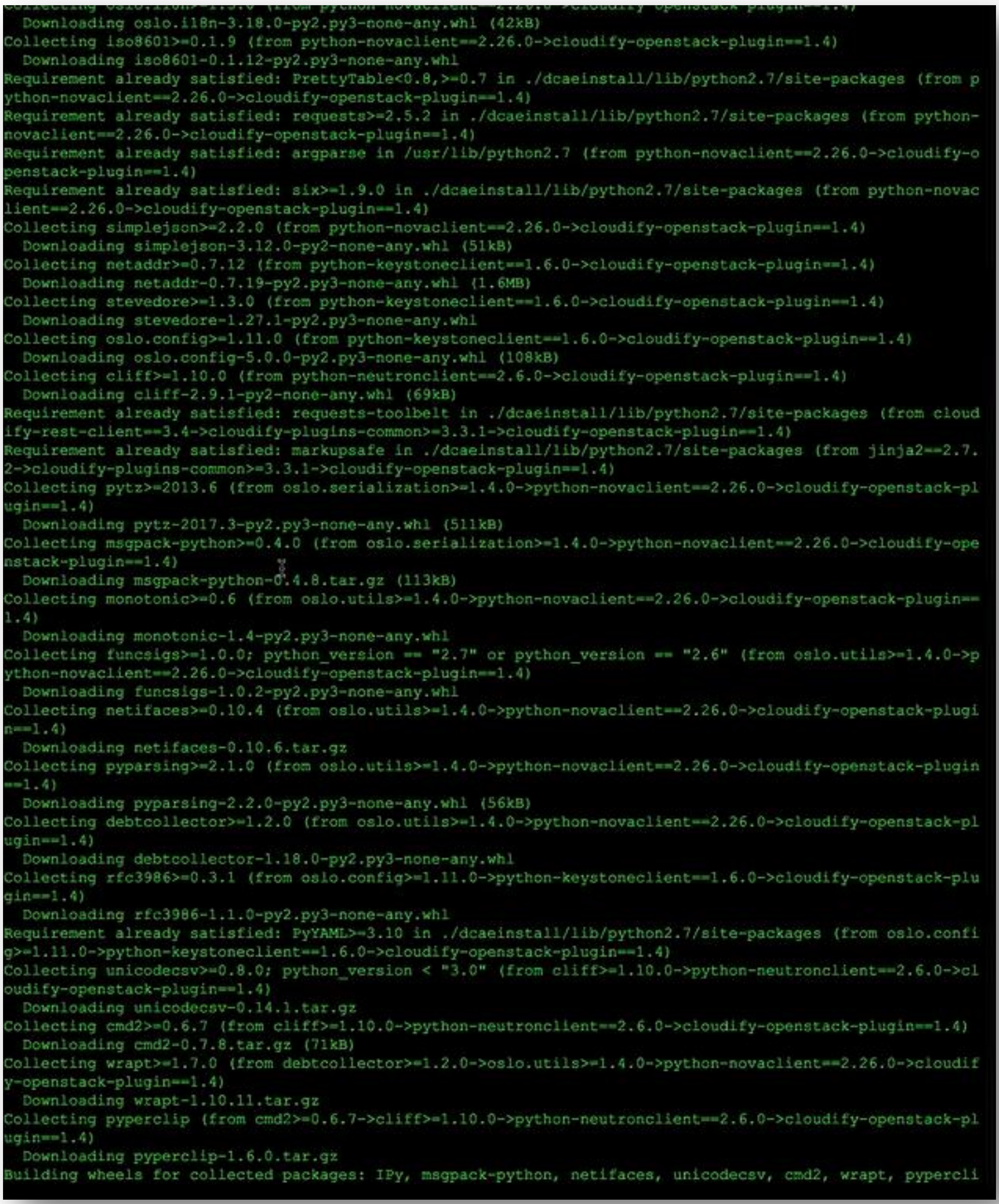
Docker logs –f boot

These were the steps it went through to install, to spin-up all the DCAE VMs and install service items and those VMs, this is very detailed logs. That is the beginning of containers log,



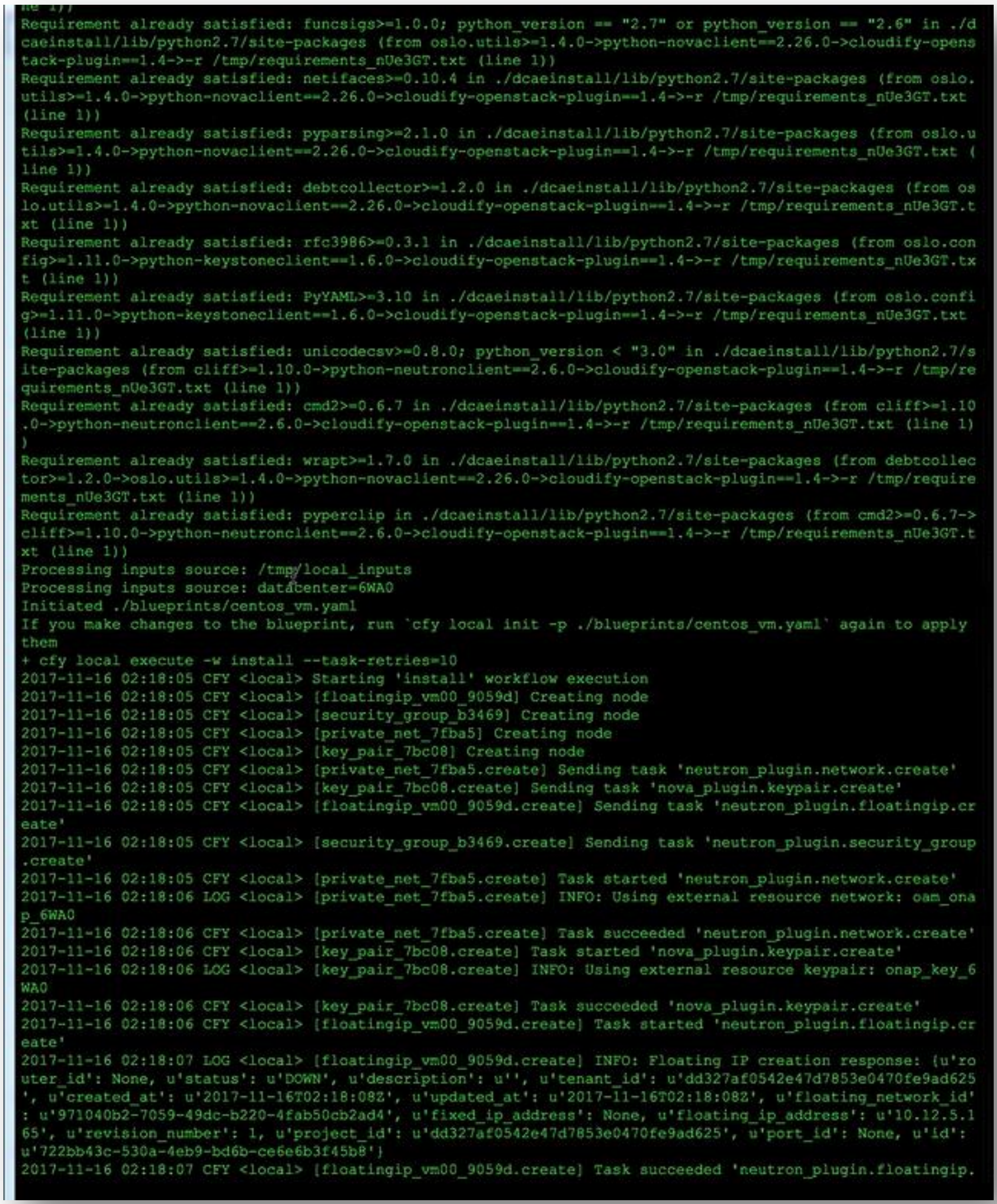
```
Requirement already satisfied: wheel>=0.24.0 in ./dcaeinstall/lib/python2.7/site-packages (from wagon==0.
3.2->cloudify==3.4.0)
Collecting virtualenv>=12.1 (from wagon==0.3.2->cloudify==3.4.0)
  Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
Collecting click==4.0 (from wagon==0.3.2->cloudify==3.4.0)
  Downloading click-4.0-py2.py3-none-any.whl (62kB)
Collecting pycrypto!=2.4,>=2.1 (from paramiko<1.13,>=1.10->fabric==1.8.3->cloudify==3.4.0)
  Downloading pycrypto-2.6.1.tar.gz (446kB)
Collecting ecdsa (from paramiko<1.13,>=1.10->fabric==1.8.3->cloudify==3.4.0)
  Downloading ecdsa-0.13-py2.py3-none-any.whl (86kB)
Building wheels for collected packages: cloudify, cloudify-plugins-common, cloudify-rest-client, cloudify
-dsl-parser, cloudify-script-plugin, pyyaml, PrettyTable, colorama, jinja2, itsdangerous, retrying, wagon
, pika, networkx, proxy-tools, bottle, paramiko, markupsafe, pycrypto
  Running setup.py bdist_wheel for cloudify: started
  Running setup.py bdist_wheel for cloudify: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/f2/c3/82/67178b6763f55a90e44ab2275208275a5a17
a67bc79f9db0b7
  Running setup.py bdist_wheel for cloudify-plugins-common: started
  Running setup.py bdist_wheel for cloudify-plugins-common: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/00/28/99/38e5cd3877708a00e49a462159693320f11a
16336a523c363c
  Running setup.py bdist_wheel for cloudify-rest-client: started
  Running setup.py bdist_wheel for cloudify-rest-client: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/86/21/96/7090ccf2eb840d5b59f8d87eeb15c5177f6f
c4efaaf3376cfb
  Running setup.py bdist_wheel for cloudify-dsl-parser: started
  Running setup.py bdist_wheel for cloudify-dsl-parser: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/60/98/2c/6ddda245951daf800173aa74c2ed0f579515
eedf88c4b81f10
  Running setup.py bdist_wheel for cloudify-script-plugin: started
  Running setup.py bdist_wheel for cloudify-script-plugin: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/84/0d/cf/561f77378a6491dd737e0b21e3661f5b978b
58282cae1c83df
  Running setup.py bdist_wheel for pyyaml: started
  Running setup.py bdist_wheel for pyyaml: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/cc/2a/d6/5a7108e2281e4c783740d79c40eac3ebc2d4
157b1c7e4f17ef
  Running setup.py bdist_wheel for PrettyTable: started
  Running setup.py bdist_wheel for PrettyTable: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/b6/90/7b/1c22b89217d0eba6d5f406e562365ebee804
f0d4595b2bdbcd
  Running setup.py bdist_wheel for colorama: started
  Running setup.py bdist_wheel for colorama: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/21/c5/cf/63fb92293f3ad402644ccaf882903cacdb8f
e87c80b62c84df
  Running setup.py bdist_wheel for jinja2: started
  Running setup.py bdist_wheel for jinja2: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/1f/e8/83/446db446804a75b7ac97bcece9a72325ee13
e11f89478ead03
  Running setup.py bdist_wheel for itsdangerous: started
  Running setup.py bdist_wheel for itsdangerous: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/fc/a8/66/24d655233c757e178d45dea2de22a04c6d92
766abfb741129a
  Running setup.py bdist_wheel for retrying: started
  Running setup.py bdist_wheel for retrying: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/d9/08/aa/49f7c109140006ea08a7657640aee3feafb6
5005bcd5280679
  Running setup.py bdist_wheel for wagon: started
  Running setup.py bdist_wheel for wagon: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/60/c9/56/5bb85a3cc242525888a4a77165a6c1a99a0f
b50b13ece972d6
  Running setup.py bdist_wheel for pika: started
  Running setup.py bdist_wheel for pika: finished with status 'done'
  Stored in directory: /opt/app/installer/.cache/pip/wheels/1f/30/61/abd15514f79d65426bfb7df4912228bed212
```

the first thing it does is to install some software locally then it was pin-up one VM to install the Cloudify Manager,



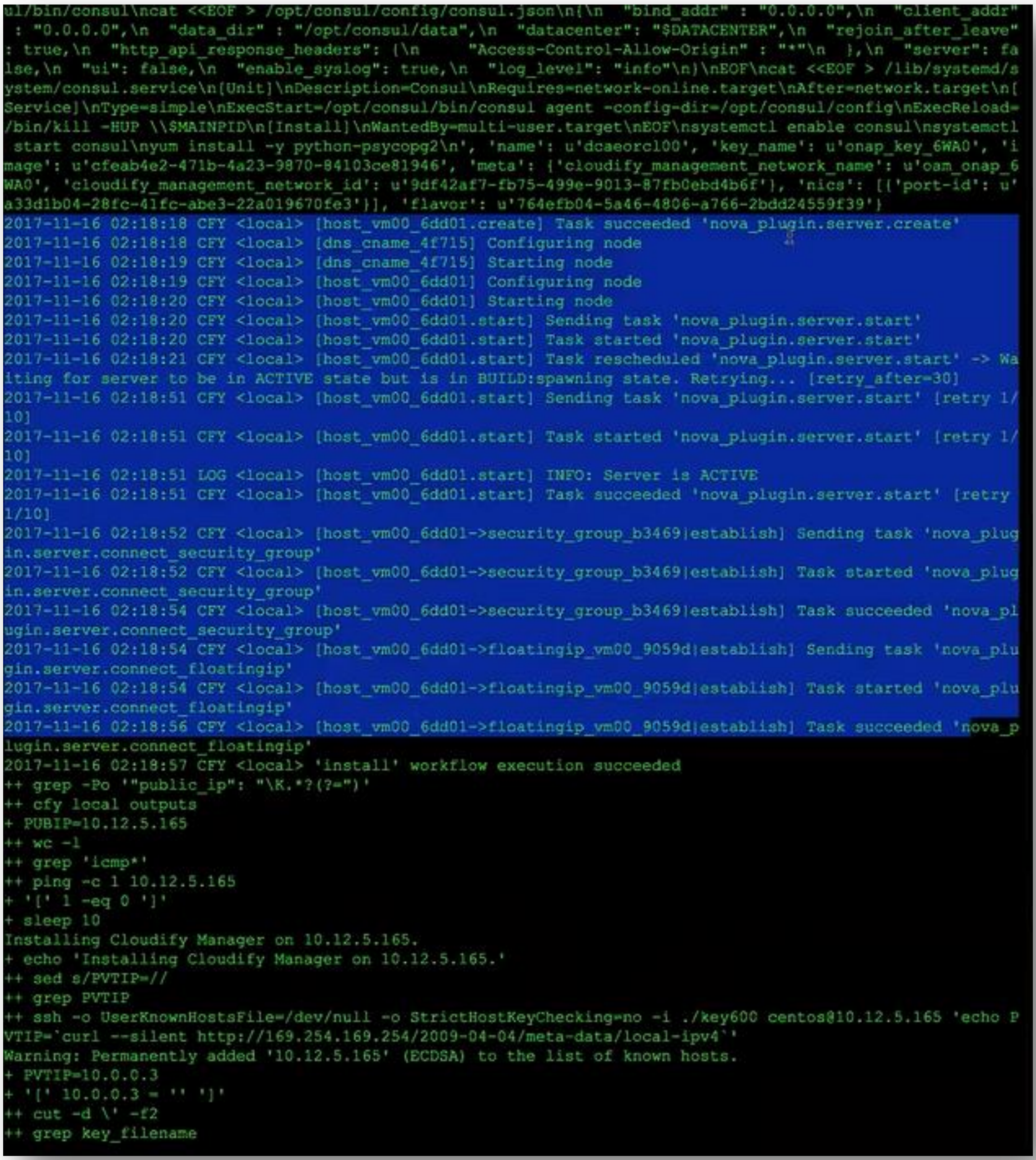
```
  Downloading oslo.i18n-3.18.0-py2.py3-none-any.whl (42kB)
Collecting iso8601>=0.1.9 (from python-novaclient==2.26.0->cloudify-openstack-plugin==1.4)
  Downloading iso8601-0.1.12-py2.py3-none-any.whl
Requirement already satisfied: PrettyTable<0.8,>=0.7 in ./dcaeinstall/lib/python2.7/site-packages (from p
ython-novaclient==2.26.0->cloudify-openstack-plugin==1.4)
Requirement already satisfied: requests>=2.5.2 in ./dcaeinstall/lib/python2.7/site-packages (from python-
novaclient==2.26.0->cloudify-openstack-plugin==1.4)
Requirement already satisfied: argparse in /usr/lib/python2.7 (from python-novaclient==2.26.0->cloudify-o
penstack-plugin==1.4)
Requirement already satisfied: six>=1.9.0 in ./dcaeinstall/lib/python2.7/site-packages (from python-novac
lient==2.26.0->cloudify-openstack-plugin==1.4)
Collecting simplejson>=2.2.0 (from python-novaclient==2.26.0->cloudify-openstack-plugin==1.4)
  Downloading simplejson-3.12.0-py2-none-any.whl (51kB)
Collecting netaddr>=0.7.12 (from python-keystoneclient==1.6.0->cloudify-openstack-plugin==1.4)
  Downloading netaddr-0.7.19-py2.py3-none-any.whl (1.6MB)
Collecting stevedore>=1.3.0 (from python-keystoneclient==1.6.0->cloudify-openstack-plugin==1.4)
  Downloading stevedore-1.27.1-py2.py3-none-any.whl
Collecting oslo.config>=1.11.0 (from python-keystoneclient==1.6.0->cloudify-openstack-plugin==1.4)
  Downloading oslo.config-5.0.0-py2.py3-none-any.whl (108kB)
Collecting cliff>=1.10.0 (from python-neutronclient==2.6.0->cloudify-openstack-plugin==1.4)
  Downloading cliff-2.9.1-py2-none-any.whl (69kB)
Requirement already satisfied: requests-toolbelt in ./dcaeinstall/lib/python2.7/site-packages (from cloud
ify-rest-client==3.4->cloudify-plugins-common>=3.3.1->cloudify-openstack-plugin==1.4)
Requirement already satisfied: markupsafe in ./dcaeinstall/lib/python2.7/site-packages (from jinja2==2.7.
2->cloudify-plugins-common>=3.3.1->cloudify-openstack-plugin==1.4)
Collecting pytz>=2013.6 (from oslo.serialization>=1.4.0->python-novaclient==2.26.0->cloudify-openstack-pl
ugin==1.4)
  Downloading pytz-2017.3-py2.py3-none-any.whl (511kB)
Collecting msgpack-python>=0.4.0 (from oslo.serialization>=1.4.0->python-novaclient==2.26.0->cloudify-ope
nstack-plugin==1.4)
  Downloading msgpack-python-0.4.8.tar.gz (113kB)
Collecting monotonic>=0.6 (from oslo.utils>=1.4.0->python-novaclient==2.26.0->cloudify-openstack-plugin==
1.4)
  Downloading monotonic-1.4-py2.py3-none-any.whl
Collecting funcsigs>=1.0.0; python_version == "2.7" or python_version == "2.6" (from oslo.utils>=1.4.0->p
ython-novaclient==2.26.0->cloudify-openstack-plugin==1.4)
  Downloading funcsigs-1.0.2-py2.py3-none-any.whl
Collecting netifaces>=0.10.4 (from oslo.utils>=1.4.0->python-novaclient==2.26.0->cloudify-openstack-plugi
n==1.4)
  Downloading netifaces-0.10.6.tar.gz
Collecting pyparsing>=2.1.0 (from oslo.utils>=1.4.0->python-novaclient==2.26.0->cloudify-openstack-plugin
==1.4)
  Downloading pyparsing-2.2.0-py2.py3-none-any.whl (56kB)
Collecting debtcollector>=1.2.0 (from oslo.utils>=1.4.0->python-novaclient==2.26.0->cloudify-openstack-pl
ugin==1.4)
  Downloading debtcollector-1.18.0-py2.py3-none-any.whl
Collecting rfc3986>=0.3.1 (from oslo.config>=1.11.0->python-keystoneclient==1.6.0->cloudify-openstack-plu
gin==1.4)
  Downloading rfc3986-1.1.0-py2.py3-none-any.whl
Requirement already satisfied: PyYAML>=3.10 in ./dcaeinstall/lib/python2.7/site-packages (from oslo.confi
g>=1.11.0->python-keystoneclient==1.6.0->cloudify-openstack-plugin==1.4)
Collecting unicodecsv>=0.8.0; python_version < "3.0" (from cliff>=1.10.0->python-neutronclient==2.6.0->cl
oudify-openstack-plugin==1.4)
  Downloading unicodecsv-0.14.1.tar.gz
Collecting cmd2>=0.6.7 (from cliff>=1.10.0->python-neutronclient==2.6.0->cloudify-openstack-plugin==1.4)
  Downloading cmd2-0.7.8.tar.gz (71kB)
Collecting wrapt>=1.7.0 (from debtcollector>=1.2.0->oslo.utils>=1.4.0->python-novaclient==2.26.0->cloudif
y-openstack-plugin==1.4)
  Downloading wrapt-1.10.11.tar.gz
Collecting pyperclip (from cmd2>=0.6.7->cliff>=1.10.0->python-neutronclient==2.6.0->cloudify-openstack-pl
ugin==1.4)
  Downloading pyperclip-1.6.0.tar.gz
Building wheels for collected packages: IPy, msgpack-python, netifaces, unicodecsv, cmd2, wrapt, pypercli
```

so this is all installation of the software locally.



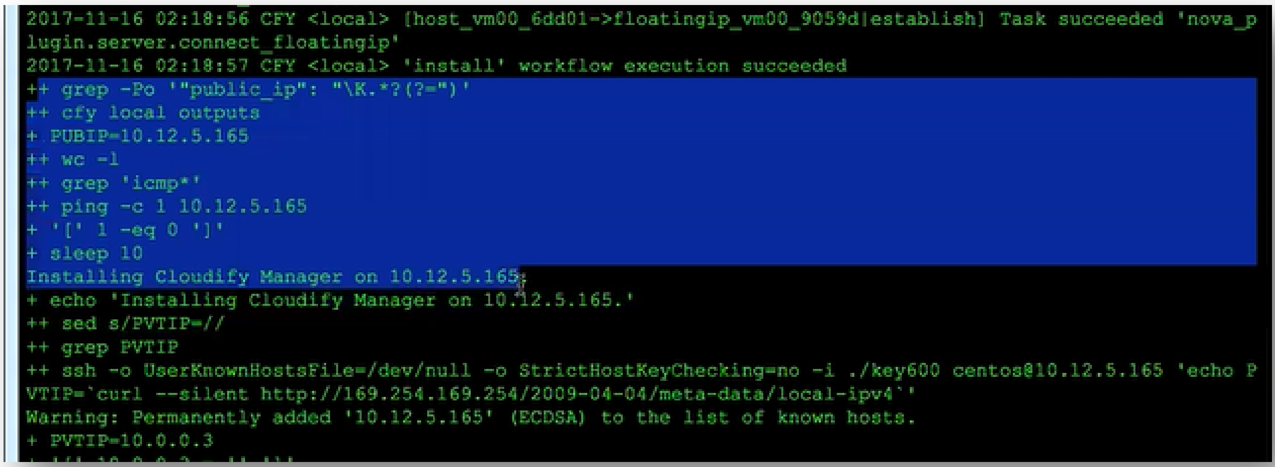
```
ne 1))
Requirement already satisfied: funcsigs>=1.0.0; python_version == "2.7" or python_version == "2.6" in ./d
caeinstall/lib/python2.7/site-packages (from oslo.utils>=1.4.0->python-novaclient==2.26.0->cloudify-opens
tack-plugin==1.4->-r /tmp/requirements_nUe3GT.txt (line 1))
Requirement already satisfied: netifaces>=0.10.4 in ./dcaeinstall/lib/python2.7/site-packages (from oslo.
utils>=1.4.0->python-novaclient==2.26.0->cloudify-openstack-plugin==1.4->-r /tmp/requirements_nUe3GT.txt
(line 1))
Requirement already satisfied: pyparsing>=2.1.0 in ./dcaeinstall/lib/python2.7/site-packages (from oslo.u
tils>=1.4.0->python-novaclient==2.26.0->cloudify-openstack-plugin==1.4->-r /tmp/requirements_nUe3GT.txt (
line 1))
Requirement already satisfied: debtcollector>=1.2.0 in ./dcaeinstall/lib/python2.7/site-packages (from os
lo.utils>=1.4.0->python-novaclient==2.26.0->cloudify-openstack-plugin==1.4->-r /tmp/requirements_nUe3GT.t
xt (line 1))
Requirement already satisfied: rfc3986>=0.3.1 in ./dcaeinstall/lib/python2.7/site-packages (from oslo.con
fig>=1.11.0->python-keystoneclient==1.6.0->cloudify-openstack-plugin==1.4->-r /tmp/requirements_nUe3GT.tx
t (line 1))
Requirement already satisfied: PyYAML>=3.10 in ./dcaeinstall/lib/python2.7/site-packages (from oslo.confi
g>=1.11.0->python-keystoneclient==1.6.0->cloudify-openstack-plugin==1.4->-r /tmp/requirements_nUe3GT.txt
(line 1))
Requirement already satisfied: unicodecsv>=0.8.0; python_version < "3.0" in ./dcaeinstall/lib/python2.7/s
ite-packages (from cliff>=1.10.0->python-neutronclient==2.6.0->cloudify-openstack-plugin==1.4->-r /tmp/re
quirements_nUe3GT.txt (line 1))
Requirement already satisfied: cmd2>=0.6.7 in ./dcaeinstall/lib/python2.7/site-packages (from cliff>=1.10
.0->python-neutronclient==2.6.0->cloudify-openstack-plugin==1.4->-r /tmp/requirements_nUe3GT.txt (line 1)
)
Requirement already satisfied: wrapt>=1.7.0 in ./dcaeinstall/lib/python2.7/site-packages (from debtcollec
tor>=1.2.0->oslo.utils>=1.4.0->python-novaclient==2.26.0->cloudify-openstack-plugin==1.4->-r /tmp/require
ments_nUe3GT.txt (line 1))
Requirement already satisfied: pyperclip in ./dcaeinstall/lib/python2.7/site-packages (from cmd2>=0.6.7->
cliff>=1.10.0->python-neutronclient==2.6.0->cloudify-openstack-plugin==1.4->-r /tmp/requirements_nUe3GT.t
xt (line 1))
Processing inputs source: /tmp/local_inputs
Processing inputs source: datacenter=6WA0
Initiated ./blueprints/centos_vm.yaml
If you make changes to the blueprint, run `cfy local init -p ./blueprints/centos_vm.yaml` again to apply
them
+ cfy local execute -w install --task-retries=10
2017-11-16 02:18:05 CFY <local> Starting 'install' workflow execution
2017-11-16 02:18:05 CFY <local> [floatingip_vm00_9059d] Creating node
2017-11-16 02:18:05 CFY <local> [security_group_b3469] Creating node
2017-11-16 02:18:05 CFY <local> [private_net_7fba5] Creating node
2017-11-16 02:18:05 CFY <local> [key_pair_7bc08] Creating node
2017-11-16 02:18:05 CFY <local> [private_net_7fba5.create] Sending task 'neutron_plugin.network.create'
2017-11-16 02:18:05 CFY <local> [key_pair_7bc08.create] Sending task 'nova_plugin.keypair.create'
2017-11-16 02:18:05 CFY <local> [floatingip_vm00_9059d.create] Sending task 'neutron_plugin.floatingip.cr
eate'
2017-11-16 02:18:05 CFY <local> [security_group_b3469.create] Sending task 'neutron_plugin.security_group
.create'
2017-11-16 02:18:05 CFY <local> [private_net_7fba5.create] Task started 'neutron_plugin.network.create'
2017-11-16 02:18:06 LOG <local> [private_net_7fba5.create] INFO: Using external resource network: oam_ona
p_6WA0
2017-11-16 02:18:06 CFY <local> [private_net_7fba5.create] Task succeeded 'neutron_plugin.network.create'
2017-11-16 02:18:06 CFY <local> [key_pair_7bc08.create] Task started 'nova_plugin.keypair.create'
2017-11-16 02:18:06 LOG <local> [key_pair_7bc08.create] INFO: Using external resource keypair: onap_key_6
WA0
2017-11-16 02:18:06 CFY <local> [key_pair_7bc08.create] Task succeeded 'nova_plugin.keypair.create'
2017-11-16 02:18:06 CFY <local> [floatingip_vm00_9059d.create] Task started 'neutron_plugin.floatingip.cr
eate'
2017-11-16 02:18:07 LOG <local> [floatingip_vm00_9059d.create] INFO: Floating IP creation response: {u'ro
uter_id': None, u'status': u'DOWN', u'description': u'', u'tenant_id': u'dd327af0542e47d7853e0470fe9ad625
', u'created_at': u'2017-11-16T02:18:08Z', u'updated_at': u'2017-11-16T02:18:08Z', u'floating_network_id'
: u'971040b2-7059-49dc-b220-4fab50cb2ad4', u'fixed_ip_address': None, u'floating_ip_address': u'10.12.5.1
65', u'revision_number': 1, u'project_id': u'dd327af0542e47d7853e0470fe9ad625', u'port_id': None, u'id':
u'722bb43c-530a-4eb9-bd6b-ce6e6b3f45b8'}
2017-11-16 02:18:07 CFY <local> [floatingip_vm00_9059d.create] Task succeeded 'neutron_plugin.floatingip.
```

For the future we can pack all those steps into the bootstrap container itself, it does not inflate that particular container but to improve boot-up time quite a lot as we need to do this every time. OK, software is now installed, then it is asking to create a new server to install Cloudify Manager,



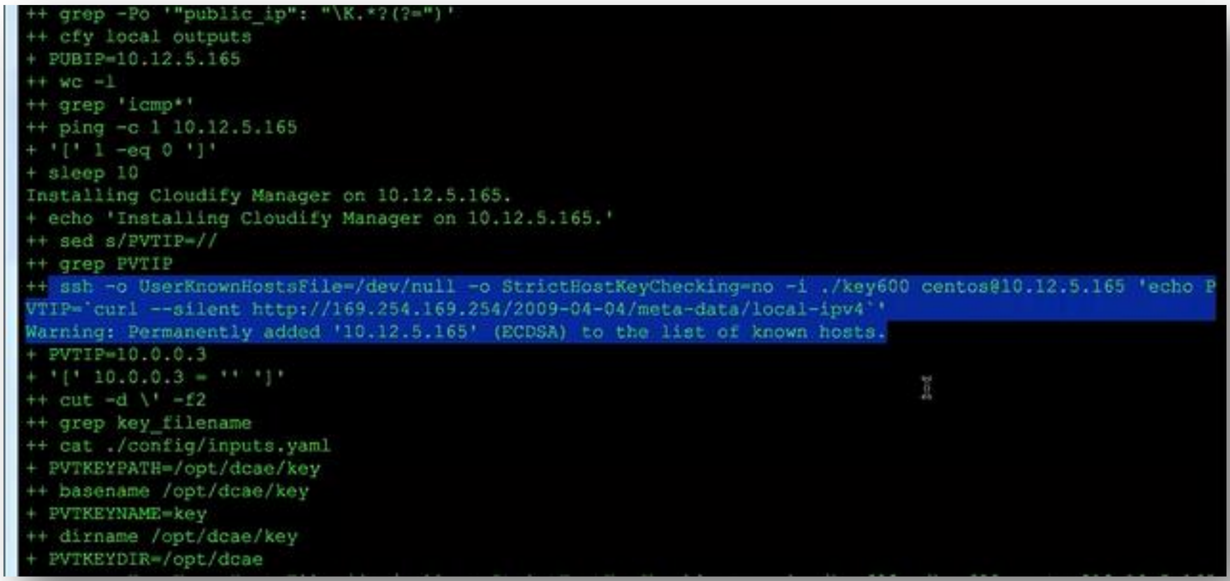
```
ul/bin/consul\ncat <<EOF > /opt/consul/config/consul.json\n{\n  "bind_addr" : "0.0.0.0",\n  "client_addr"
 : "0.0.0.0",\n  "data_dir" : "/opt/consul/data",\n  "datacenter": "$DATACENTER",\n  "rejoin_after_leave"
: true,\n  "http_api_response_headers": {\n     "Access-Control-Allow-Origin" : "*"\n  },\n  "server": fa
lse,\n  "ui": false,\n  "enable_syslog": true,\n  "log_level": "info"\n}\nEOF\ncat <<EOF > /lib/systemd/s
ystem/consul.service\n[Unit]\nDescription=Consul\nRequires=network-online.target\nAfter=network.target\n[
Service]\nType=simple\nExecStart=/opt/consul/bin/consul agent -config-dir=/opt/consul/config\nExecReload=
/bin/kill -HUP \\$MAINPID\n[Install]\nWantedBy=multi-user.target\nEOF\nsystemctl enable consul\nsystemctl
 start consul\nyum install -y python-psycopg2\n', 'name': u'dcaeorcl00', 'key_name': u'onap_key_6WA0', 'i
mage': u'cfeab4e2-471b-4a23-9870-84103ce81946', 'meta': {'cloudify_management_network_name': u'oam_onap_6
WA0', 'cloudify_management_network_id': u'9df42af7-fb75-499e-9013-87fb0ebd4b6f'}, 'nics': [{'port-id': u'
a33d1b04-28fc-41fc-abe3-22a019670fe3'}], 'flavor': u'764efb04-5a46-4806-a766-2bdd24559f39'}
2017-11-16 02:18:18 CFY <local> [host_vm00_6dd01.create] Task succeeded 'nova_plugin.server.create'
2017-11-16 02:18:18 CFY <local> [dns_cname_4f715] Configuring node
2017-11-16 02:18:19 CFY <local> [dns_cname_4f715] Starting node
2017-11-16 02:18:19 CFY <local> [host_vm00_6dd01] Configuring node
2017-11-16 02:18:20 CFY <local> [host_vm00_6dd01] Starting node
2017-11-16 02:18:20 CFY <local> [host_vm00_6dd01.start] Sending task 'nova_plugin.server.start'
2017-11-16 02:18:20 CFY <local> [host_vm00_6dd01.start] Task started 'nova_plugin.server.start'
2017-11-16 02:18:21 CFY <local> [host_vm00_6dd01.start] Task rescheduled 'nova_plugin.server.start' -> Wa
iting for server to be in ACTIVE state but is in BUILD:spawning state. Retrying... [retry_after=30]
2017-11-16 02:18:51 CFY <local> [host_vm00_6dd01.start] Sending task 'nova_plugin.server.start' [retry 1/
10]
2017-11-16 02:18:51 CFY <local> [host_vm00_6dd01.start] Task started 'nova_plugin.server.start' [retry 1/
10]
2017-11-16 02:18:51 LOG <local> [host_vm00_6dd01.start] INFO: Server is ACTIVE
2017-11-16 02:18:51 CFY <local> [host_vm00_6dd01.start] Task succeeded 'nova_plugin.server.start' [retry
1/10]
2017-11-16 02:18:52 CFY <local> [host_vm00_6dd01->security_group_b3469|establish] Sending task 'nova_plug
in.server.connect_security_group'
2017-11-16 02:18:52 CFY <local> [host_vm00_6dd01->security_group_b3469|establish] Task started 'nova_plug
in.server.connect_security_group'
2017-11-16 02:18:54 CFY <local> [host_vm00_6dd01->security_group_b3469|establish] Task succeeded 'nova_pl
ugin.server.connect_security_group'
2017-11-16 02:18:54 CFY <local> [host_vm00_6dd01->floatingip_vm00_9059d|establish] Sending task 'nova_plu
gin.server.connect_floatingip'
2017-11-16 02:18:54 CFY <local> [host_vm00_6dd01->floatingip_vm00_9059d|establish] Task started 'nova_plu
gin.server.connect_floatingip'
2017-11-16 02:18:56 CFY <local> [host_vm00_6dd01->floatingip_vm00_9059d|establish] Task succeeded 'nova_p
lugin.server.connect_floatingip'
2017-11-16 02:18:57 CFY <local> 'install' workflow execution succeeded
++ grep -Po '"public_ip": "\K.*?(?=")'
++ cfy local outputs
+ PUBIP=10.12.5.165
++ wc -l
++ grep 'icmp*'
++ ping -c 1 10.12.5.165
+ '[' 1 -eq 0 ']'
+ sleep 10
Installing Cloudify Manager on 10.12.5.165.
+ echo 'Installing Cloudify Manager on 10.12.5.165.'
++ sed s/PVTIP=//
++ grep PVTIP
++ ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -i ./key600 centos@10.12.5.165 'echo P
VTIP=`curl --silent http://169.254.169.254/2009-04-04/meta-data/local-ipv4`'
Warning: Permanently added '10.12.5.165' (ECDSA) to the list of known hosts.
+ PVTIP=10.0.0.3
+ '[' 10.0.0.3 = '' ']'
++ cut -d \' -f2
++ grep key_filename
```

then it does wait for that VM to come up,



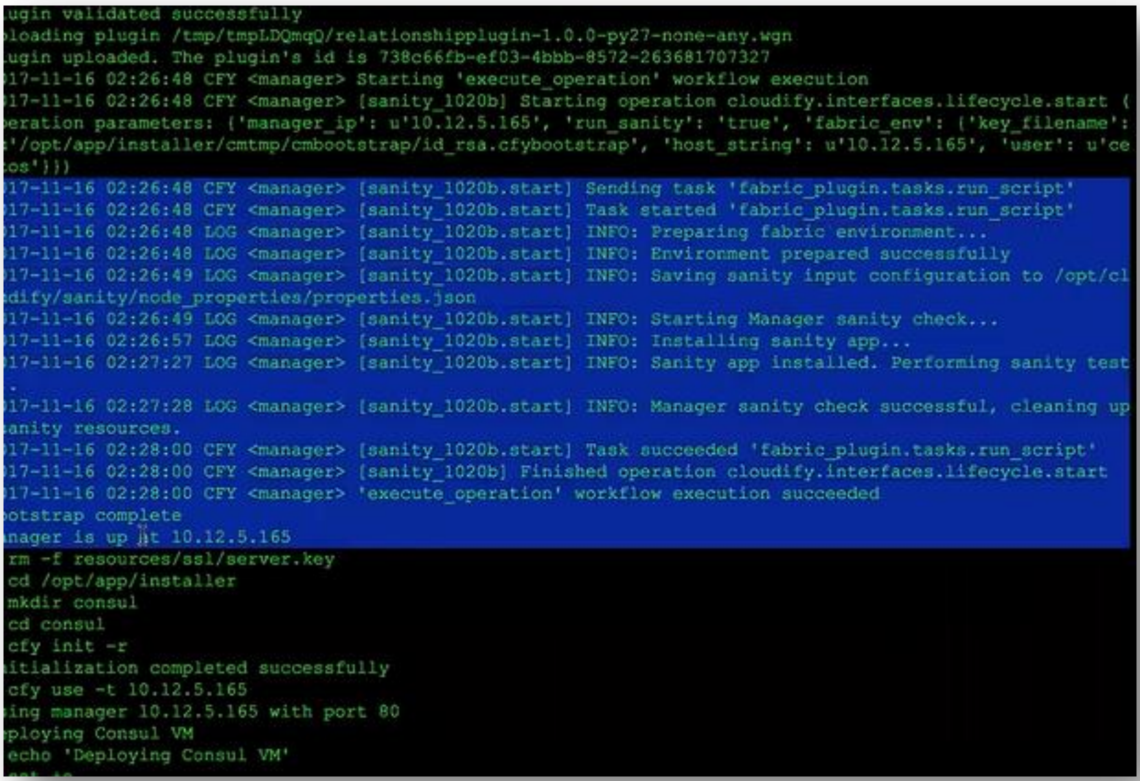
```
2017-11-16 02:18:56 CFY <local> [host_vm00_6dd01->floatingip_vm00_9059d|establish] Task succeeded 'nova_p
lugin.server.connect_floatingip'
2017-11-16 02:18:57 CFY <local> 'install' workflow execution succeeded
++ grep -Po '"public_ip": "\K.*?(?=")'
++ cfy local outputs
+ PUBIP=10.12.5.165
++ wc -l
++ grep 'icmp*'
++ ping -c 1 10.12.5.165
+ '[' 1 -eq 0 ']'
+ sleep 10
Installing Cloudify Manager on 10.12.5.165.
+ echo 'Installing Cloudify Manager on 10.12.5.165.'
++ sed s/PVTIP=//
++ grep PVTIP
++ ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -i ./key600 centos@10.12.5.165 'echo P
VTIP=`curl --silent http://169.254.169.254/2009-04-04/meta-data/local-ipv4`'
Warning: Permanently added '10.12.5.165' (ECDSA) to the list of known hosts.
+ PVTIP=10.0.0.3
```

after it is coming up, then it will perform an installation work on that particular VM using SSH



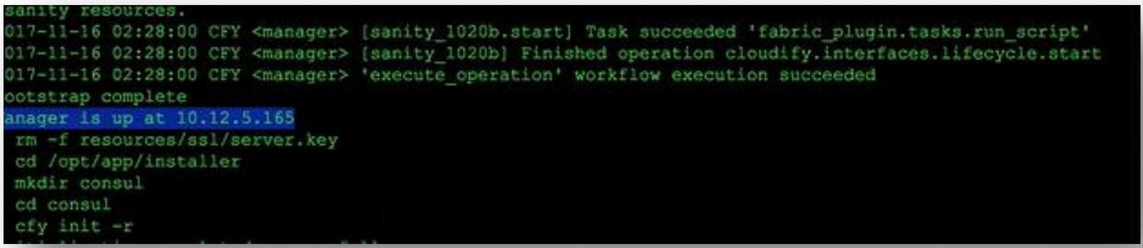
```
++ grep -Po '"public_ip": "\K.*?(?=")'
++ cfy local outputs
+ PUBIP=10.12.5.165
++ wc -l
++ grep 'icmp*'
++ ping -c 1 10.12.5.165
+ '[' 1 -eq 0 ']'
+ sleep 10
Installing Cloudify Manager on 10.12.5.165.
+ echo 'Installing Cloudify Manager on 10.12.5.165.'
++ sed s/PVTIP=//
++ grep PVTIP
++ ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no -i ./key600 centos@10.12.5.165 'echo P
VTIP=`curl --silent http://169.254.169.254/2009-04-04/meta-data/local-ipv4`'
Warning: Permanently added '10.12.5.165' (ECDSA) to the list of known hosts.
+ PVTIP=10.0.0.3
+ '[' 10.0.0.3 = '' ']'
++ cut -d \' -f2
++ grep key_filename
++ cat ./config/inputs.yaml
+ PVTKEYPATH=/opt/dcae/key
++ basename /opt/dcae/key
+ PVTKEYNAME=key
++ dirname /opt/dcae/key
+ PVTKEYDIR=/opt/dcae
```

and here is all those things happening in the Cloudify Manager view – quite a lot logs to go through, and at the end



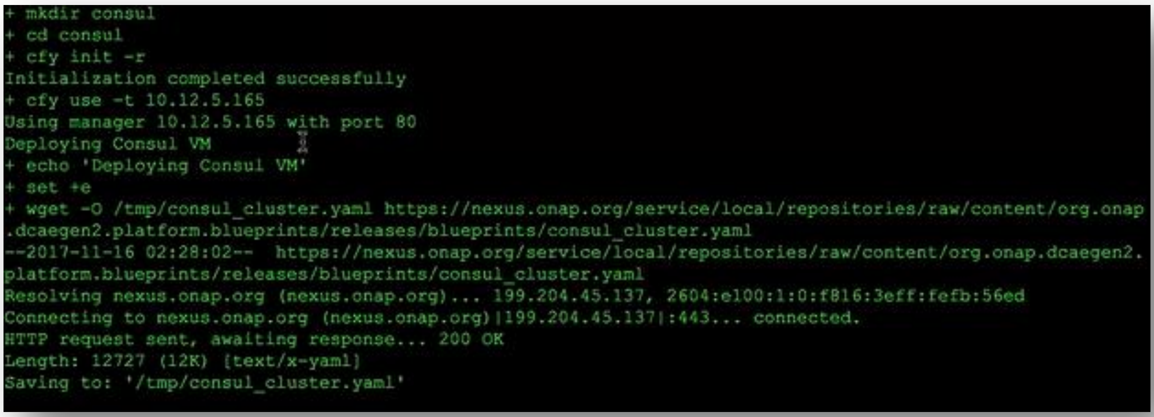
```
ugin validated successfully
ploading plugin /tmp/tmpLDQmqQ/relationshipplugin-1.0.0-py27-none-any.wgn
ugin uploaded. The plugin's id is 738c66fb-ef03-4bbb-8572-263681707327
)17-11-16 02:26:48 CFY <manager> Starting 'execute_operation' workflow execution
)17-11-16 02:26:48 CFY <manager> [sanity_1020b] Starting operation cloudify.interfaces.lifecycle.start (
eration parameters: {'manager_ip': u'10.12.5.165', 'run_sanity': 'true', 'fabric_env': {'key_filename':
:'/opt/app/installer/cmtmp/cmbootstrap/id_rsa.cfybootstrap', 'host_string': u'10.12.5.165', 'user': u'ce
tos'}})
)17-11-16 02:26:48 CFY <manager> [sanity_1020b.start] Sending task 'fabric_plugin.tasks.run_script'
)17-11-16 02:26:48 CFY <manager> [sanity_1020b.start] Task started 'fabric_plugin.tasks.run_script'
)17-11-16 02:26:48 LOG <manager> [sanity_1020b.start] INFO: Preparing fabric environment...
)17-11-16 02:26:48 LOG <manager> [sanity_1020b.start] INFO: Environment prepared successfully
)17-11-16 02:26:49 LOG <manager> [sanity_1020b.start] INFO: Saving sanity input configuration to /opt/cl
dify/sanity/node_properties/properties.json
)17-11-16 02:26:49 LOG <manager> [sanity_1020b.start] INFO: Starting Manager sanity check...
)17-11-16 02:26:57 LOG <manager> [sanity_1020b.start] INFO: Installing sanity app...
)17-11-16 02:27:27 LOG <manager> [sanity_1020b.start] INFO: Sanity app installed. Performing sanity test
.
)17-11-16 02:27:28 LOG <manager> [sanity_1020b.start] INFO: Manager sanity check successful, cleaning up
anity resources.
)17-11-16 02:28:00 CFY <manager> [sanity_1020b.start] Task succeeded 'fabric_plugin.tasks.run_script'
)17-11-16 02:28:00 CFY <manager> [sanity_1020b] Finished operation cloudify.interfaces.lifecycle.start
)17-11-16 02:28:00 CFY <manager> 'execute_operation' workflow execution succeeded
otstrap complete
nager is up at 10.12.5.165
 rm -f resources/ssl/server.key
 cd /opt/app/installer
 mkdir consul
 cd consul
 cfy init -r
itialization completed successfully
 cfy use -t 10.12.5.165
sing manager 10.12.5.165 with port 80
ploying Consul VM
 echo 'Deploying Consul VM'
```

of that after Cloudify Manager is up we can see (Manager is up at 10.12.5.165),



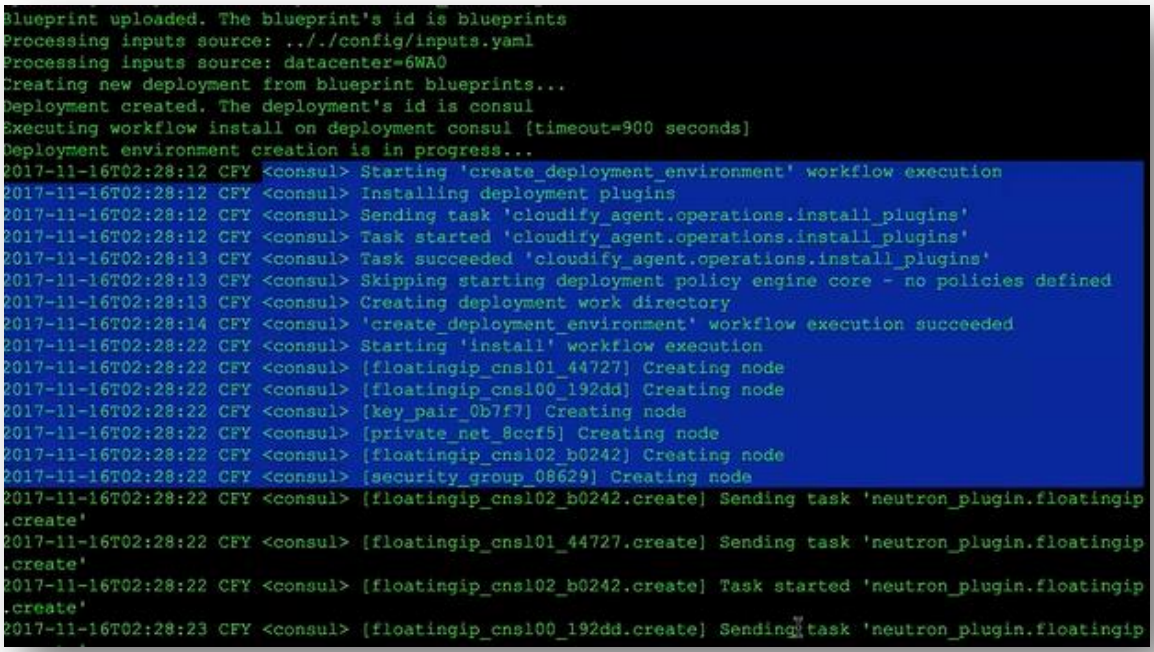
```
sanity resources.
017-11-16 02:28:00 CFY <manager> [sanity_1020b.start] Task succeeded 'fabric_plugin.tasks.run_script'
017-11-16 02:28:00 CFY <manager> [sanity_1020b] Finished operation cloudify.interfaces.lifecycle.start
017-11-16 02:28:00 CFY <manager> 'execute_operation' workflow execution succeeded
ootstrap complete
anager is up at 10.12.5.165
 rm -f resources/ssl/server.key
 cd /opt/app/installer
 mkdir consul
 cd consul
 cfy init -r
```

the next step is to deploy a console cluster



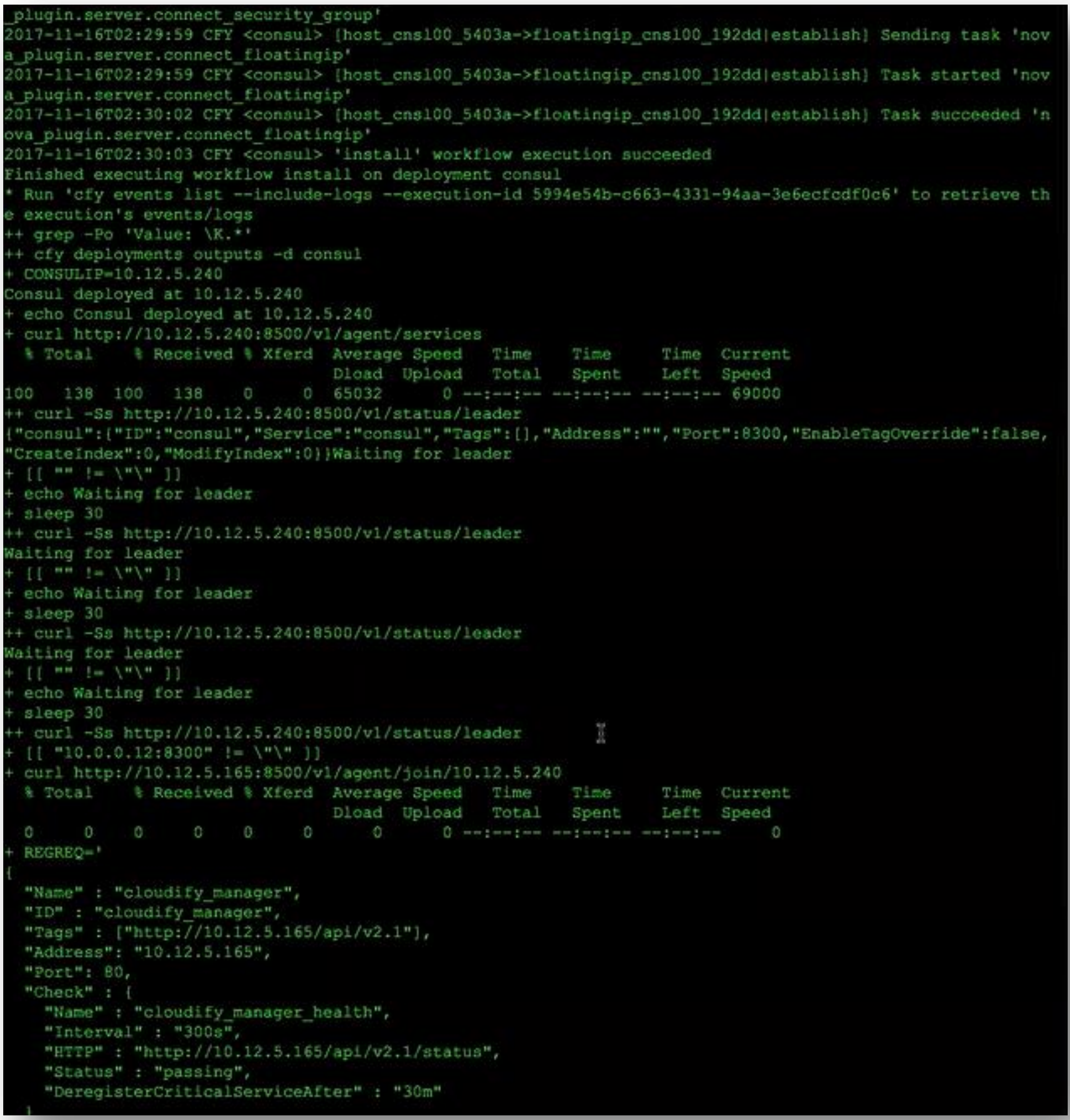
```
+ mkdir consul
+ cd consul
+ cfy init -r
Initialization completed successfully
+ cfy use -t 10.12.5.165
Using manager 10.12.5.165 with port 80
Deploying Consul VM
+ echo 'Deploying Consul VM'
+ set +e
+ wget -O /tmp/consul_cluster.yaml https://nexus.onap.org/service/local/repositories/raw/content/org.onap
.dcaegen2.platform.blueprints/releases/blueprints/consul_cluster.yaml
--2017-11-16 02:28:02--  https://nexus.onap.org/service/local/repositories/raw/content/org.onap.dcaegen2.
platform.blueprints/releases/blueprints/consul_cluster.yaml
Resolving nexus.onap.org (nexus.onap.org)... 199.204.45.137, 2604:e100:1:0:f816:3eff:fefb:56ed
Connecting to nexus.onap.org (nexus.onap.org)|199.204.45.137|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12727 (12K) [text/x-yaml]
Saving to: '/tmp/consul_cluster.yaml'
```

and these are the steps to deploy a console cluster: again asking the Openstack to create VMs,



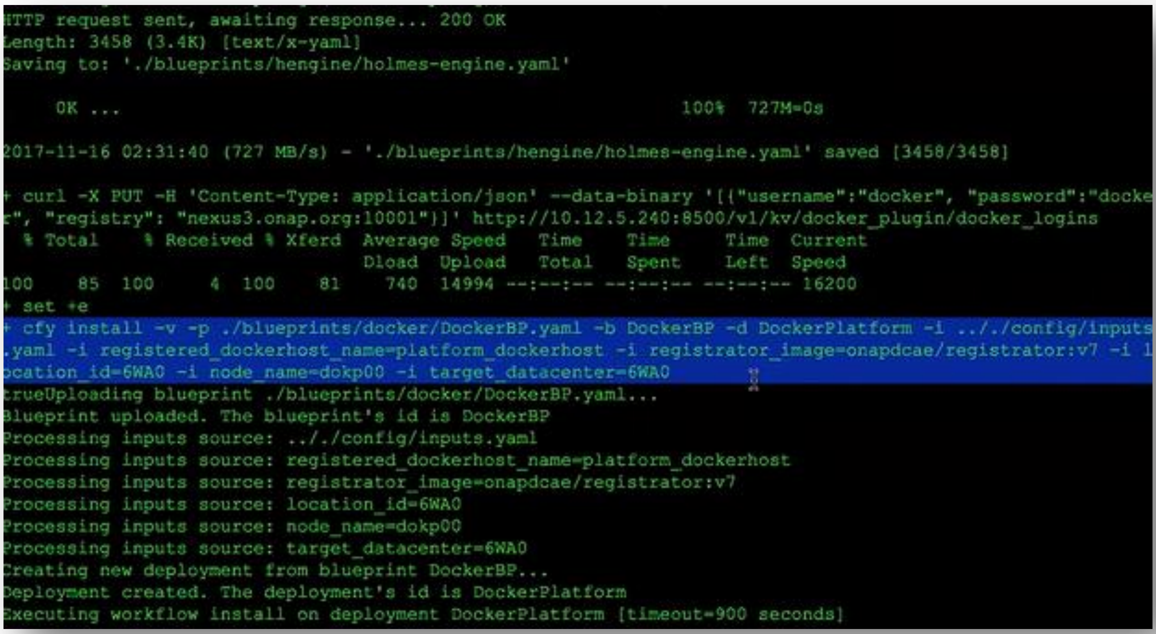
```
Blueprint uploaded. The blueprint's id is blueprints
Processing inputs source: ../../config/inputs.yaml
Processing inputs source: datacenter=6WA0
Creating new deployment from blueprint blueprints...
Deployment created. The deployment's id is consul
Executing workflow install on deployment consul [timeout=900 seconds]
Deployment environment creation is in progress...
2017-11-16T02:28:12 CFY <consul> Starting 'create_deployment_environment' workflow execution
2017-11-16T02:28:12 CFY <consul> Installing deployment plugins
2017-11-16T02:28:12 CFY <consul> Sending task 'cloudify_agent.operations.install_plugins'
2017-11-16T02:28:12 CFY <consul> Task started 'cloudify_agent.operations.install_plugins'
2017-11-16T02:28:13 CFY <consul> Task succeeded 'cloudify_agent.operations.install_plugins'
2017-11-16T02:28:13 CFY <consul> Skipping starting deployment policy engine core - no policies defined
2017-11-16T02:28:13 CFY <consul> Creating deployment work directory
2017-11-16T02:28:14 CFY <consul> 'create_deployment_environment' workflow execution succeeded
2017-11-16T02:28:22 CFY <consul> Starting 'install' workflow execution
2017-11-16T02:28:22 CFY <consul> [floatingip_cnsl01_44727] Creating node
2017-11-16T02:28:22 CFY <consul> [floatingip_cnsl00_192dd] Creating node
2017-11-16T02:28:22 CFY <consul> [key_pair_0b7f7] Creating node
2017-11-16T02:28:22 CFY <consul> [private_net_8ccf5] Creating node
2017-11-16T02:28:22 CFY <consul> [floatingip_cnsl02_b0242] Creating node
2017-11-16T02:28:22 CFY <consul> [security_group_08629] Creating node
2017-11-16T02:28:22 CFY <consul> [floatingip_cnsl02_b0242.create] Sending task 'neutron_plugin.floatingip
.create'
2017-11-16T02:28:22 CFY <consul> [floatingip_cnsl01_44727.create] Sending task 'neutron_plugin.floatingip
.create'
2017-11-16T02:28:22 CFY <consul> [floatingip_cnsl02_b0242.create] Task started 'neutron_plugin.floatingip
.create'
2017-11-16T02:28:23 CFY <consul> [floatingip_cnsl00_192dd.create] Sending task 'neutron_plugin.floatingip
```

then installation of the software. Console cluster is 3 VM cluster – it is designed for HA reasons, you can also distribute the members of this cluster,



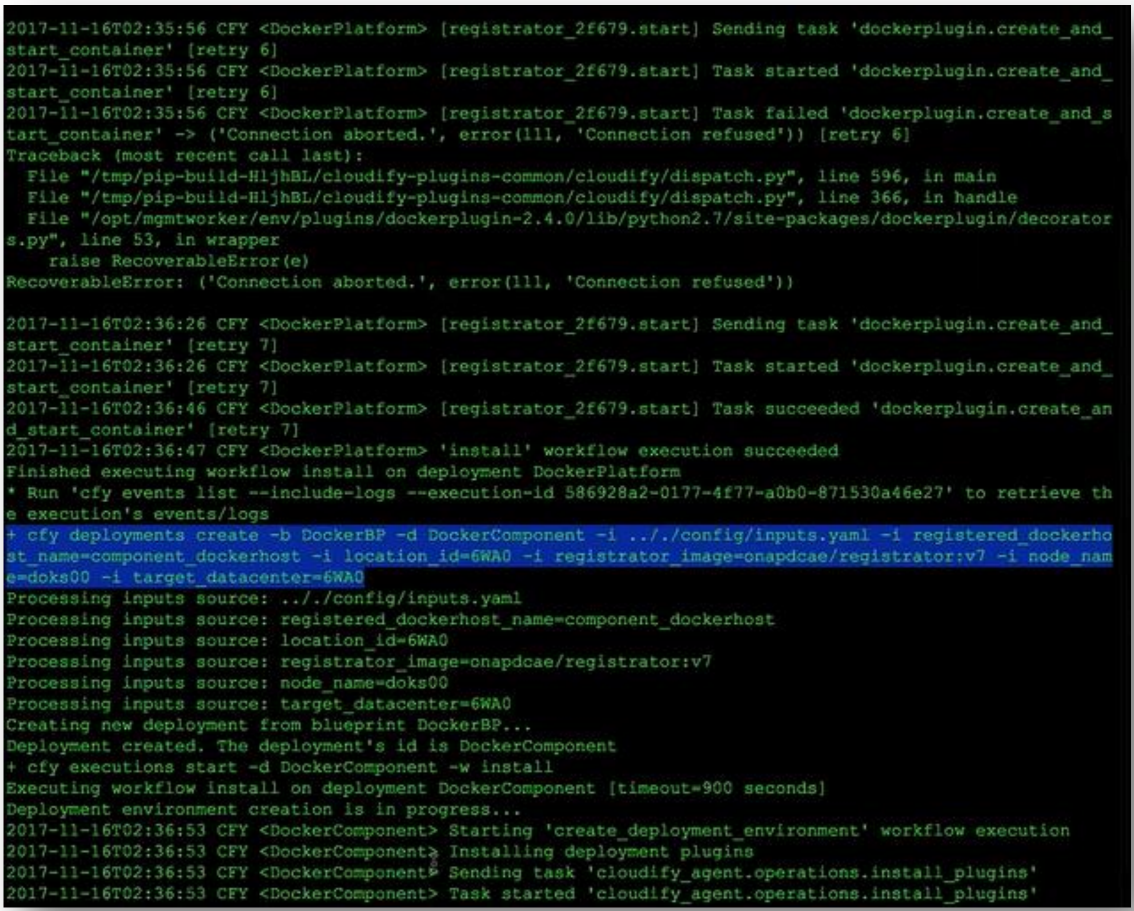
```
_plugin.server.connect_security_group'
2017-11-16T02:29:59 CFY <consul> [host_cns100_5403a->floatingip_cns100_192dd|establish] Sending task 'nov
a_plugin.server.connect_floatingip'
2017-11-16T02:29:59 CFY <consul> [host_cns100_5403a->floatingip_cns100_192dd|establish] Task started 'nov
a_plugin.server.connect_floatingip'
2017-11-16T02:30:02 CFY <consul> [host_cns100_5403a->floatingip_cns100_192dd|establish] Task succeeded 'n
ova_plugin.server.connect_floatingip'
2017-11-16T02:30:03 CFY <consul> 'install' workflow execution succeeded
Finished executing workflow install on deployment consul
* Run 'cfy events list --include-logs --execution-id 5994e54b-c663-4331-94aa-3e6ecfcdf0c6' to retrieve th
e execution's events/logs
++ grep -Po 'Value: \K.*'
++ cfy deployments outputs -d consul
+ CONSULIP=10.12.5.240
Consul deployed at 10.12.5.240
+ echo Consul deployed at 10.12.5.240
+ curl http://10.12.5.240:8500/v1/agent/services
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   138  100   138    0     0  65032      0 --:--:-- --:--:-- --:--:-- 69000
++ curl -Ss http://10.12.5.240:8500/v1/status/leader
{"consul":{"ID":"consul","Service":"consul","Tags":[],"Address":"","Port":8300,"EnableTagOverride":false,
"CreateIndex":0,"ModifyIndex":0}}Waiting for leader
+ [[ "" != \"\" ]]
+ echo Waiting for leader
+ sleep 30
++ curl -Ss http://10.12.5.240:8500/v1/status/leader
Waiting for leader
+ [[ "" != \"\" ]]
+ echo Waiting for leader
+ sleep 30
++ curl -Ss http://10.12.5.240:8500/v1/status/leader
Waiting for leader
+ [[ "" != \"\" ]]
+ echo Waiting for leader
+ sleep 30
++ curl -Ss http://10.12.5.240:8500/v1/status/leader
+ [[ "10.0.0.12:8300" != \"\" ]]
+ curl http://10.12.5.165:8500/v1/agent/join/10.12.5.240
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
+ REGREQ='
{
  "Name" : "cloudify_manager",
  "ID" : "cloudify_manager",
  "Tags" : ["http://10.12.5.165/api/v2.1"],
  "Address": "10.12.5.165",
  "Port": 80,
  "Check" : {
    "Name" : "cloudify_manager_health",
    "Interval" : "300s",
    "HTTP" : "http://10.12.5.165/api/v2.1/status",
    "Status" : "passing",
    "DeregisterCriticalServiceAfter" : "30m"
```

they perform leader election, perform synchronization, so using this cluster you can really have a very wide coverage of the information that is provided though a console. All these are installing console. After the console is done, the next step is where we are installing the docker host.



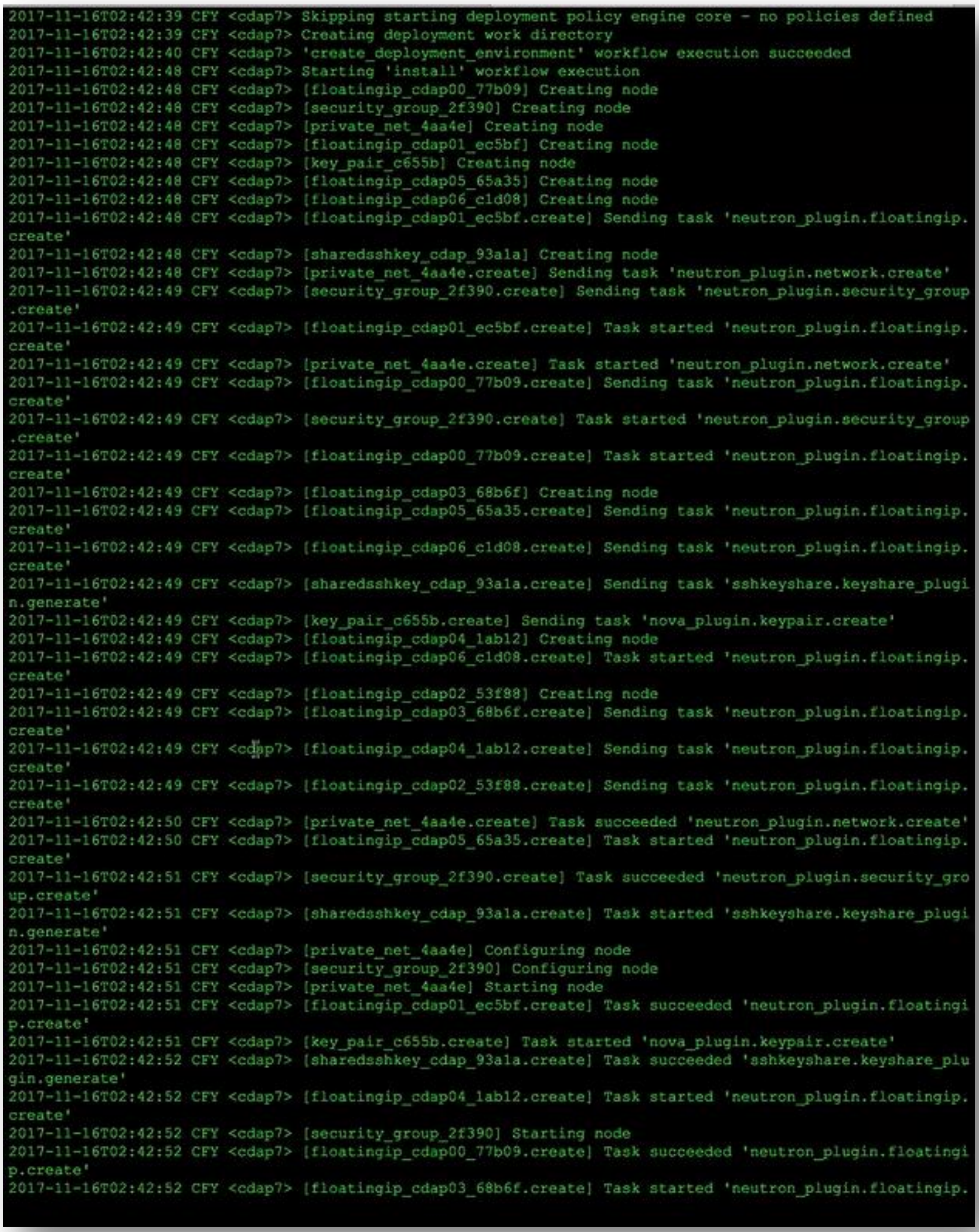
```
HTTP request sent, awaiting response... 200 OK
Length: 3458 (3.4K) [text/x-yaml]
Saving to: './blueprints/hengine/holmes-engine.yaml'

     OK ...                                                    100%  727M=0s

2017-11-16 02:31:40 (727 MB/s) - './blueprints/hengine/holmes-engine.yaml' saved [3458/3458]

+ curl -X PUT -H 'Content-Type: application/json' --data-binary '[{"username":"docker", "password":"docke
r", "registry": "nexus3.onap.org:10001"}]' http://10.12.5.240:8500/v1/kv/docker_plugin/docker_logins
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    85  100     4  100    81    740  14994 --:--:-- --:--:-- --:--:-- 16200
+ set +e
+ cfy install -v -p ./blueprints/docker/DockerBP.yaml -b DockerBP -d DockerPlatform -i ../../config/inputs
.yaml -i registered_dockerhost_name=platform_dockerhost -i registrator_image=onapdcae/registrator:v7 -i l
ocation_id=6WA0 -i node_name=dokp00 -i target_datacenter=6WA0
trueUploading blueprint ./blueprints/docker/DockerBP.yaml...
Blueprint uploaded. The blueprint's id is DockerBP
Processing inputs source: ../../config/inputs.yaml
Processing inputs source: registered_dockerhost_name=platform_dockerhost
Processing inputs source: registrator_image=onapdcae/registrator:v7
Processing inputs source: location_id=6WA0
Processing inputs source: node_name=dokp00
Processing inputs source: target_datacenter=6WA0
Creating new deployment from blueprint DockerBP...
Deployment created. The deployment's id is DockerPlatform
Executing workflow install on deployment DockerPlatform [timeout=900 seconds]
```

This docker host is used for installing DCAE platform components such as the Policy Handler, such as Deploy Handler – those are already docker containerized and there will be running on that docker host platform so that is installing docker platform. And after that the second docker host is launched – it is used for installing the service components,



```
2017-11-16T02:35:56 CFY <DockerPlatform> [registrator_2f679.start] Sending task 'dockerplugin.create_and_
start_container' [retry 6]
2017-11-16T02:35:56 CFY <DockerPlatform> [registrator_2f679.start] Task started 'dockerplugin.create_and_
start_container' [retry 6]
2017-11-16T02:35:56 CFY <DockerPlatform> [registrator_2f679.start] Task failed 'dockerplugin.create_and_s
tart_container' -> ('Connection aborted.', error(111, 'Connection refused')) [retry 6]
Traceback (most recent call last):
  File "/tmp/pip-build-HljhBL/cloudify-plugins-common/cloudify/dispatch.py", line 596, in main
  File "/tmp/pip-build-HljhBL/cloudify-plugins-common/cloudify/dispatch.py", line 366, in handle
  File "/opt/mgmtworker/env/plugins/dockerplugin-2.4.0/lib/python2.7/site-packages/dockerplugin/decorator
s.py", line 53, in wrapper
    raise RecoverableError(e)
RecoverableError: ('Connection aborted.', error(111, 'Connection refused'))

2017-11-16T02:36:26 CFY <DockerPlatform> [registrator_2f679.start] Sending task 'dockerplugin.create_and_
start_container' [retry 7]
2017-11-16T02:36:26 CFY <DockerPlatform> [registrator_2f679.start] Task started 'dockerplugin.create_and_
start_container' [retry 7]
2017-11-16T02:36:46 CFY <DockerPlatform> [registrator_2f679.start] Task succeeded 'dockerplugin.create_an
d_start_container' [retry 7]
2017-11-16T02:36:47 CFY <DockerPlatform> 'install' workflow execution succeeded
Finished executing workflow install on deployment DockerPlatform
* Run 'cfy events list --include-logs --execution-id 586928a2-0177-4f77-a0b0-871530a46e27' to retrieve th
e execution's events/logs
+ cfy deployments create -b DockerBP -d DockerComponent -i ../config/inputs.yaml -i registered_dockerho
st_name=component_dockerhost -i location_id=6WA0 -i registrator_image=onapdcae/registrator:v7 -i node_nam
e=doks00 -i target_datacenter=6WA0
Processing inputs source: ../config/inputs.yaml
Processing inputs source: registered_dockerhost_name=component_dockerhost
Processing inputs source: location_id=6WA0
Processing inputs source: registrator_image=onapdcae/registrator:v7
Processing inputs source: node_name=doks00
Processing inputs source: target_datacenter=6WA0
Creating new deployment from blueprint DockerBP...
Deployment created. The deployment's id is DockerComponent
+ cfy executions start -d DockerComponent -w install
Executing workflow install on deployment DockerComponent [timeout=900 seconds]
Deployment environment creation is in progress...
2017-11-16T02:36:53 CFY <DockerComponent> Starting 'create_deployment_environment' workflow execution
2017-11-16T02:36:53 CFY <DockerComponent> Installing deployment plugins
2017-11-16T02:36:53 CFY <DockerComponent> Sending task 'cloudify_agent.operations.install_plugins'
2017-11-16T02:36:53 CFY <DockerComponent> Task started 'cloudify_agent.operations.install_plugins'
```

so for example the VES collector is a service component – it is a docker container and it will be installed on this docker host. So there are 2 docker hosts. Then after that there is a big a CDAP cluster. CDAP cluster consists of 7 VMs – the reason is in production environment per the Hadoop technology provider hoping works in this case they only trust VM cluster of certain size because only that they can guarantee certain service level – they can say – ok we can handle even 3 VMs down for example, that is the service level they want to provide and therefore the large size of the cluster.

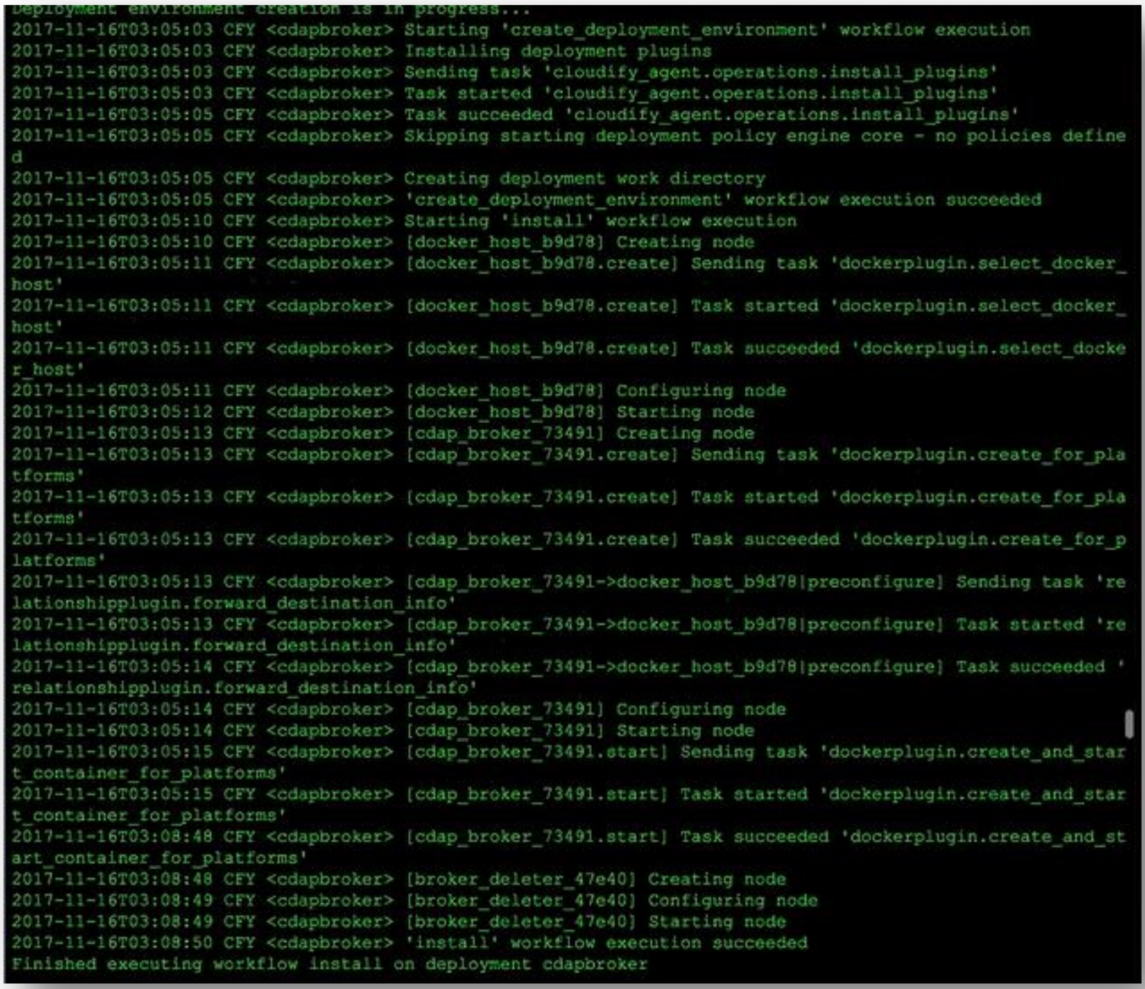


```
2017-11-16T02:42:39 CFY <cdap7> Skipping starting deployment policy engine core - no policies defined
2017-11-16T02:42:39 CFY <cdap7> Creating deployment work directory
2017-11-16T02:42:40 CFY <cdap7> 'create_deployment_environment' workflow execution succeeded
2017-11-16T02:42:48 CFY <cdap7> Starting 'install' workflow execution
2017-11-16T02:42:48 CFY <cdap7> [floatingip_cdap00_77b09] Creating node
2017-11-16T02:42:48 CFY <cdap7> [security_group_2f390] Creating node
2017-11-16T02:42:48 CFY <cdap7> [private_net_4aa4e] Creating node
2017-11-16T02:42:48 CFY <cdap7> [floatingip_cdap01_ec5bf] Creating node
2017-11-16T02:42:48 CFY <cdap7> [key_pair_c655b] Creating node
2017-11-16T02:42:48 CFY <cdap7> [floatingip_cdap05_65a35] Creating node
2017-11-16T02:42:48 CFY <cdap7> [floatingip_cdap06_c1d08] Creating node
2017-11-16T02:42:48 CFY <cdap7> [floatingip_cdap01_ec5bf.create] Sending task 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:48 CFY <cdap7> [sharedsshkey_cdap_93a1a] Creating node
2017-11-16T02:42:48 CFY <cdap7> [private_net_4aa4e.create] Sending task 'neutron_plugin.network.create'
2017-11-16T02:42:49 CFY <cdap7> [security_group_2f390.create] Sending task 'neutron_plugin.security_group
.create'
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap01_ec5bf.create] Task started 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:49 CFY <cdap7> [private_net_4aa4e.create] Task started 'neutron_plugin.network.create'
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap00_77b09.create] Sending task 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:49 CFY <cdap7> [security_group_2f390.create] Task started 'neutron_plugin.security_group
.create'
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap00_77b09.create] Task started 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap03_68b6f] Creating node
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap05_65a35.create] Sending task 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap06_c1d08.create] Sending task 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:49 CFY <cdap7> [sharedsshkey_cdap_93a1a.create] Sending task 'sshkeyshare.keyshare_plugi
n.generate'
2017-11-16T02:42:49 CFY <cdap7> [key_pair_c655b.create] Sending task 'nova_plugin.keypair.create'
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap04_1ab12] Creating node
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap06_c1d08.create] Task started 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap02_53f88] Creating node
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap03_68b6f.create] Sending task 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap04_1ab12.create] Sending task 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:49 CFY <cdap7> [floatingip_cdap02_53f88.create] Sending task 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:50 CFY <cdap7> [private_net_4aa4e.create] Task succeeded 'neutron_plugin.network.create'
2017-11-16T02:42:50 CFY <cdap7> [floatingip_cdap05_65a35.create] Task started 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:51 CFY <cdap7> [security_group_2f390.create] Task succeeded 'neutron_plugin.security_gro
up.create'
2017-11-16T02:42:51 CFY <cdap7> [sharedsshkey_cdap_93a1a.create] Task started 'sshkeyshare.keyshare_plugi
n.generate'
2017-11-16T02:42:51 CFY <cdap7> [private_net_4aa4e] Configuring node
2017-11-16T02:42:51 CFY <cdap7> [security_group_2f390] Configuring node
2017-11-16T02:42:51 CFY <cdap7> [private_net_4aa4e] Starting node
2017-11-16T02:42:51 CFY <cdap7> [floatingip_cdap01_ec5bf.create] Task succeeded 'neutron_plugin.floatingi
p.create'
2017-11-16T02:42:51 CFY <cdap7> [key_pair_c655b.create] Task started 'nova_plugin.keypair.create'
2017-11-16T02:42:52 CFY <cdap7> [sharedsshkey_cdap_93a1a.create] Task succeeded 'sshkeyshare.keyshare_plu
gin.generate'
2017-11-16T02:42:52 CFY <cdap7> [floatingip_cdap04_1ab12.create] Task started 'neutron_plugin.floatingip.
create'
2017-11-16T02:42:52 CFY <cdap7> [security_group_2f390] Starting node
2017-11-16T02:42:52 CFY <cdap7> [floatingip_cdap00_77b09.create] Task succeeded 'neutron_plugin.floatingi
p.create'
2017-11-16T02:42:52 CFY <cdap7> [floatingip_cdap03_68b6f.create] Task started 'neutron_plugin.floatingip.
```

There's all the CDAPs steps.

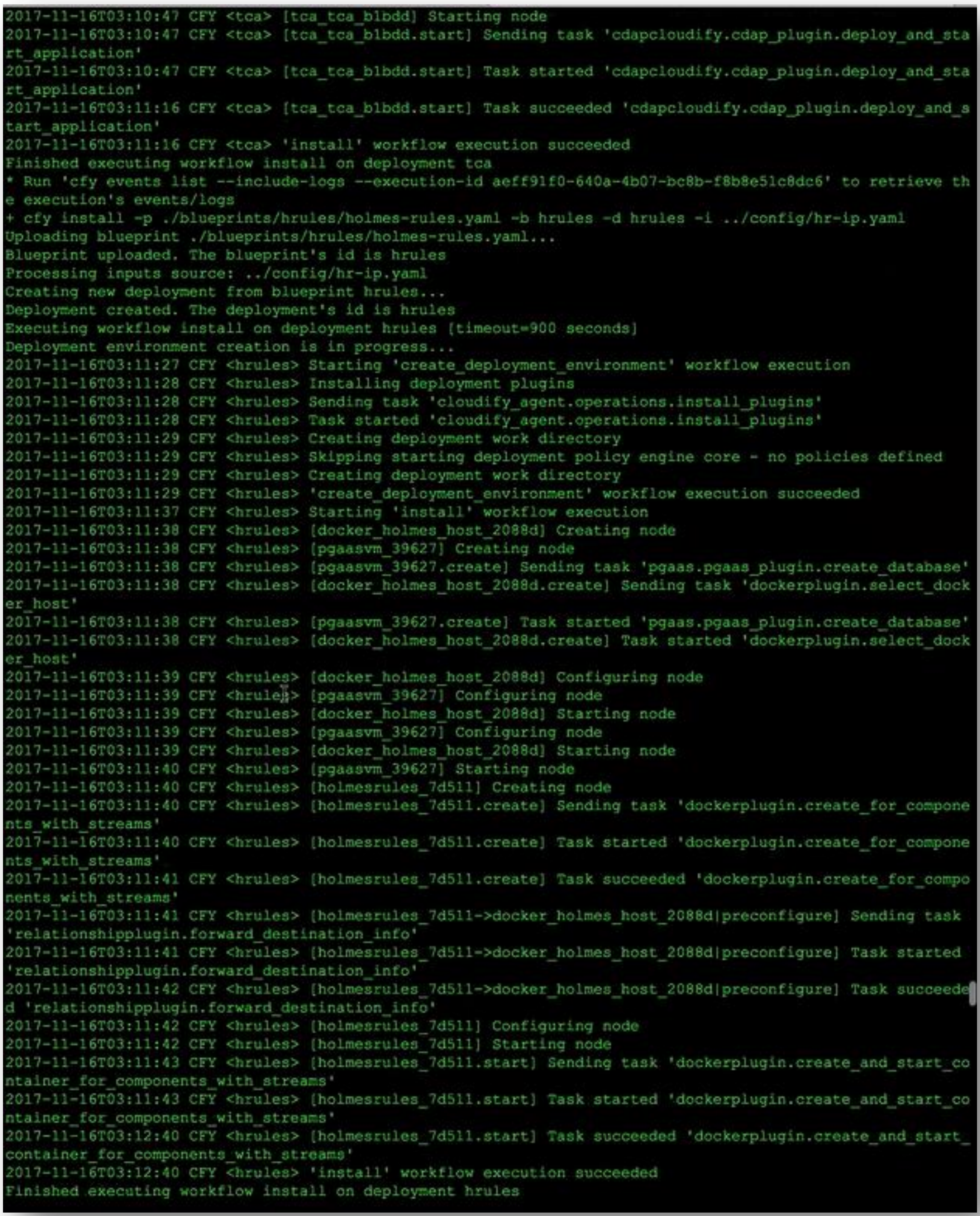
Question: is it configurable the number of VMs in cluster to have for example only 1 VM for a demo or test lab 7 cluster is quite big – it will be explained later on how to customize this configuration.

After the CDAP is installed there were installed additional components, you can see that there is a CDAP Broker,



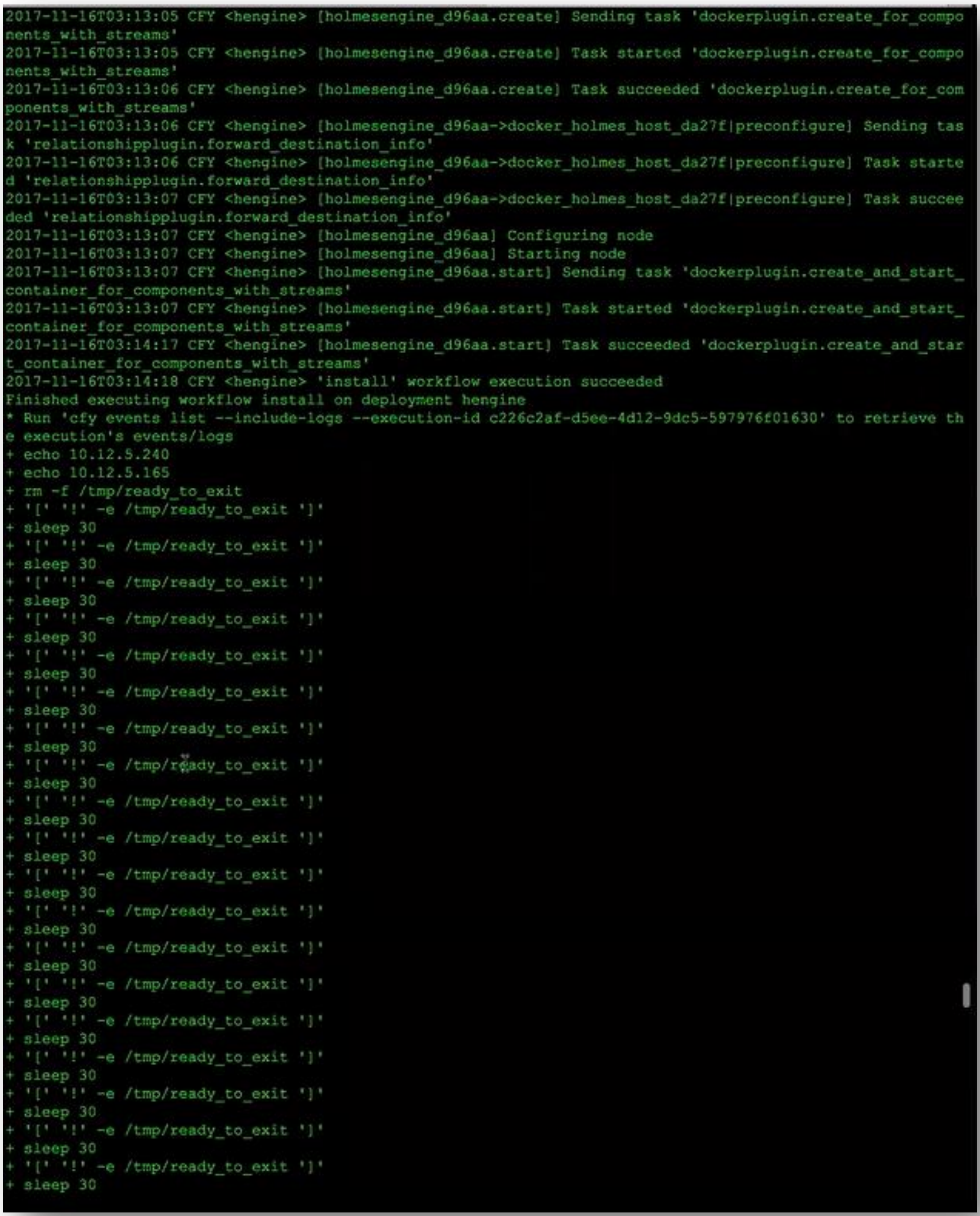
```
Deployment environment creation is in progress...
2017-11-16T03:05:03 CFY <cdapbroker> Starting 'create_deployment_environment' workflow execution
2017-11-16T03:05:03 CFY <cdapbroker> Installing deployment plugins
2017-11-16T03:05:03 CFY <cdapbroker> Sending task 'cloudify_agent.operations.install_plugins'
2017-11-16T03:05:03 CFY <cdapbroker> Task started 'cloudify_agent.operations.install_plugins'
2017-11-16T03:05:05 CFY <cdapbroker> Task succeeded 'cloudify_agent.operations.install_plugins'
2017-11-16T03:05:05 CFY <cdapbroker> Skipping starting deployment policy engine core - no policies define
d
2017-11-16T03:05:05 CFY <cdapbroker> Creating deployment work directory
2017-11-16T03:05:05 CFY <cdapbroker> 'create_deployment_environment' workflow execution succeeded
2017-11-16T03:05:10 CFY <cdapbroker> Starting 'install' workflow execution
2017-11-16T03:05:10 CFY <cdapbroker> [docker_host_b9d78] Creating node
2017-11-16T03:05:11 CFY <cdapbroker> [docker_host_b9d78.create] Sending task 'dockerplugin.select_docker_
host'
2017-11-16T03:05:11 CFY <cdapbroker> [docker_host_b9d78.create] Task started 'dockerplugin.select_docker_
host'
2017-11-16T03:05:11 CFY <cdapbroker> [docker_host_b9d78.create] Task succeeded 'dockerplugin.select_docke
r_host'
2017-11-16T03:05:11 CFY <cdapbroker> [docker_host_b9d78] Configuring node
2017-11-16T03:05:12 CFY <cdapbroker> [docker_host_b9d78] Starting node
2017-11-16T03:05:13 CFY <cdapbroker> [cdap_broker_73491] Creating node
2017-11-16T03:05:13 CFY <cdapbroker> [cdap_broker_73491.create] Sending task 'dockerplugin.create_for_pla
tforms'
2017-11-16T03:05:13 CFY <cdapbroker> [cdap_broker_73491.create] Task started 'dockerplugin.create_for_pla
tforms'
2017-11-16T03:05:13 CFY <cdapbroker> [cdap_broker_73491.create] Task succeeded 'dockerplugin.create_for_p
latforms'
2017-11-16T03:05:13 CFY <cdapbroker> [cdap_broker_73491->docker_host_b9d78|preconfigure] Sending task 're
lationshipplugin.forward_destination_info'
2017-11-16T03:05:13 CFY <cdapbroker> [cdap_broker_73491->docker_host_b9d78|preconfigure] Task started 're
lationshipplugin.forward_destination_info'
2017-11-16T03:05:14 CFY <cdapbroker> [cdap_broker_73491->docker_host_b9d78|preconfigure] Task succeeded '
relationshipplugin.forward_destination_info'
2017-11-16T03:05:14 CFY <cdapbroker> [cdap_broker_73491] Configuring node
2017-11-16T03:05:14 CFY <cdapbroker> [cdap_broker_73491] Starting node
2017-11-16T03:05:15 CFY <cdapbroker> [cdap_broker_73491.start] Sending task 'dockerplugin.create_and_star
t_container_for_platforms'
2017-11-16T03:05:15 CFY <cdapbroker> [cdap_broker_73491.start] Task started 'dockerplugin.create_and_star
t_container_for_platforms'
2017-11-16T03:08:48 CFY <cdapbroker> [cdap_broker_73491.start] Task succeeded 'dockerplugin.create_and_st
art_container_for_platforms'
2017-11-16T03:08:48 CFY <cdapbroker> [broker_deleter_47e40] Creating node
2017-11-16T03:08:49 CFY <cdapbroker> [broker_deleter_47e40] Configuring node
2017-11-16T03:08:49 CFY <cdapbroker> [broker_deleter_47e40] Starting node
2017-11-16T03:08:50 CFY <cdapbroker> 'install' workflow execution succeeded
Finished executing workflow install on deployment cdapbroker
```

Policy Handler. All those components then the VES Collector, then the TCA , here is a Holmes correlation



```
2017-11-16T03:10:47 CFY <tca> [tca_tca_b1bdd] Starting node
2017-11-16T03:10:47 CFY <tca> [tca_tca_b1bdd.start] Sending task 'cdapcloudify.cdap_plugin.deploy_and_sta
rt_application'
2017-11-16T03:10:47 CFY <tca> [tca_tca_b1bdd.start] Task started 'cdapcloudify.cdap_plugin.deploy_and_sta
rt_application'
2017-11-16T03:11:16 CFY <tca> [tca_tca_b1bdd.start] Task succeeded 'cdapcloudify.cdap_plugin.deploy_and_s
tart_application'
2017-11-16T03:11:16 CFY <tca> 'install' workflow execution succeeded
Finished executing workflow install on deployment tca
* Run 'cfy events list --include-logs --execution-id aeff91f0-640a-4b07-bc8b-f8b8e51c8dc6' to retrieve th
e execution's events/logs
+ cfy install -p ./blueprints/hrules/holmes-rules.yaml -b hrules -d hrules -i ../config/hr-ip.yaml
Uploading blueprint ./blueprints/hrules/holmes-rules.yaml...
Blueprint uploaded. The blueprint's id is hrules
Processing inputs source: ../config/hr-ip.yaml
Creating new deployment from blueprint hrules...
Deployment created. The deployment's id is hrules
Executing workflow install on deployment hrules [timeout=900 seconds]
Deployment environment creation is in progress...
2017-11-16T03:11:27 CFY <hrules> Starting 'create_deployment_environment' workflow execution
2017-11-16T03:11:28 CFY <hrules> Installing deployment plugins
2017-11-16T03:11:28 CFY <hrules> Sending task 'cloudify_agent.operations.install_plugins'
2017-11-16T03:11:28 CFY <hrules> Task started 'cloudify_agent.operations.install_plugins'
2017-11-16T03:11:29 CFY <hrules> Creating deployment work directory
2017-11-16T03:11:29 CFY <hrules> Skipping starting deployment policy engine core - no policies defined
2017-11-16T03:11:29 CFY <hrules> Creating deployment work directory
2017-11-16T03:11:29 CFY <hrules> 'create_deployment_environment' workflow execution succeeded
2017-11-16T03:11:37 CFY <hrules> Starting 'install' workflow execution
2017-11-16T03:11:38 CFY <hrules> [docker_holmes_host_2088d] Creating node
2017-11-16T03:11:38 CFY <hrules> [pgaasvm_39627] Creating node
2017-11-16T03:11:38 CFY <hrules> [pgaasvm_39627.create] Sending task 'pgaas.pgaas_plugin.create_database'
2017-11-16T03:11:38 CFY <hrules> [docker_holmes_host_2088d.create] Sending task 'dockerplugin.select_dock
er_host'
2017-11-16T03:11:38 CFY <hrules> [pgaasvm_39627.create] Task started 'pgaas.pgaas_plugin.create_database'
2017-11-16T03:11:38 CFY <hrules> [docker_holmes_host_2088d.create] Task started 'dockerplugin.select_dock
er_host'
2017-11-16T03:11:39 CFY <hrules> [docker_holmes_host_2088d] Configuring node
2017-11-16T03:11:39 CFY <hrules> [pgaasvm_39627] Configuring node
2017-11-16T03:11:39 CFY <hrules> [docker_holmes_host_2088d] Starting node
2017-11-16T03:11:39 CFY <hrules> [pgaasvm_39627] Configuring node
2017-11-16T03:11:39 CFY <hrules> [docker_holmes_host_2088d] Starting node
2017-11-16T03:11:40 CFY <hrules> [pgaasvm_39627] Starting node
2017-11-16T03:11:40 CFY <hrules> [holmesrules_7d511] Creating node
2017-11-16T03:11:40 CFY <hrules> [holmesrules_7d511.create] Sending task 'dockerplugin.create_for_compone
nts_with_streams'
2017-11-16T03:11:40 CFY <hrules> [holmesrules_7d511.create] Task started 'dockerplugin.create_for_compone
nts_with_streams'
2017-11-16T03:11:41 CFY <hrules> [holmesrules_7d511.create] Task succeeded 'dockerplugin.create_for_compo
nents_with_streams'
2017-11-16T03:11:41 CFY <hrules> [holmesrules_7d511->docker_holmes_host_2088d|preconfigure] Sending task
'relationshipplugin.forward_destination_info'
2017-11-16T03:11:41 CFY <hrules> [holmesrules_7d511->docker_holmes_host_2088d|preconfigure] Task started
'relationshipplugin.forward_destination_info'
2017-11-16T03:11:42 CFY <hrules> [holmesrules_7d511->docker_holmes_host_2088d|preconfigure] Task succeede
d 'relationshipplugin.forward_destination_info'
2017-11-16T03:11:42 CFY <hrules> [holmesrules_7d511] Configuring node
2017-11-16T03:11:42 CFY <hrules> [holmesrules_7d511] Starting node
2017-11-16T03:11:43 CFY <hrules> [holmesrules_7d511.start] Sending task 'dockerplugin.create_and_start_co
ntainer_for_components_with_streams'
2017-11-16T03:11:43 CFY <hrules> [holmesrules_7d511.start] Task started 'dockerplugin.create_and_start_co
ntainer_for_components_with_streams'
2017-11-16T03:12:40 CFY <hrules> [holmesrules_7d511.start] Task succeeded 'dockerplugin.create_and_start_
container_for_components_with_streams'
2017-11-16T03:12:40 CFY <hrules> 'install' workflow execution succeeded
Finished executing workflow install on deployment hrules
```

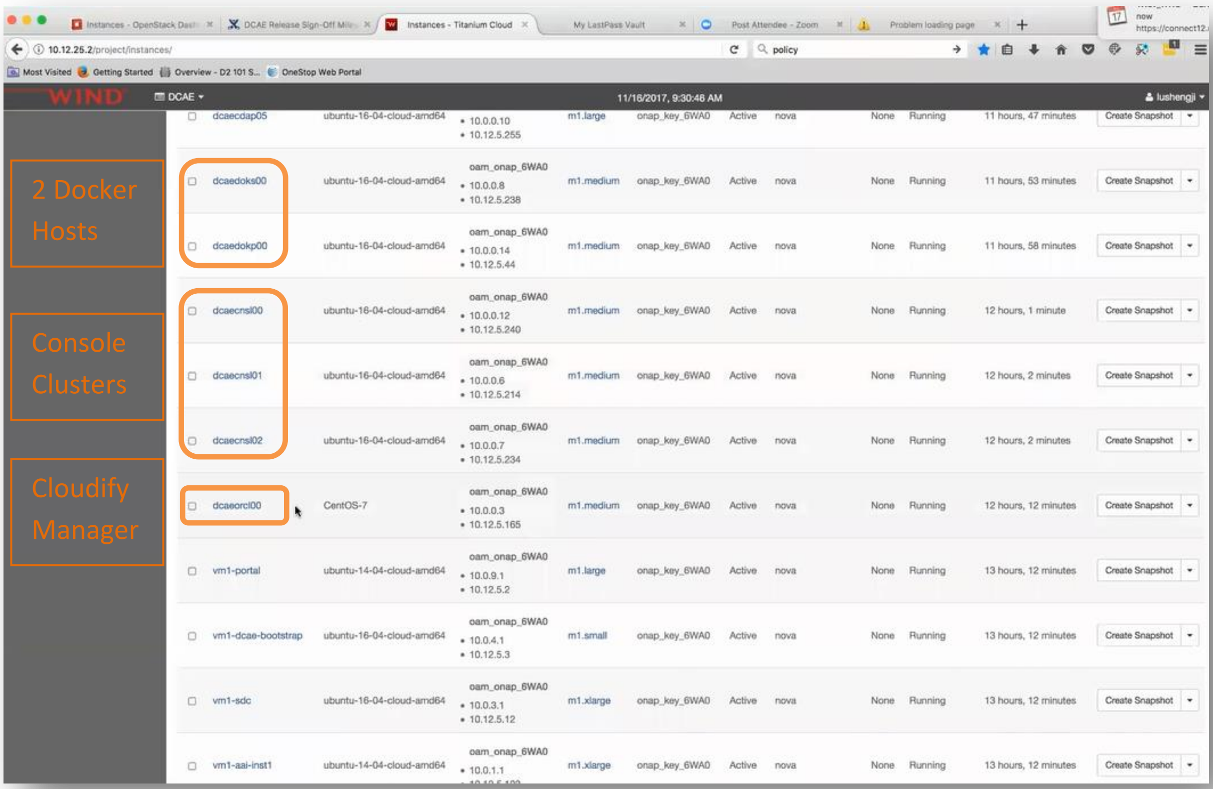
– all those they are installed and at the end of the bootstrap container just goes into an internal loop



```
2017-11-16T03:13:05 CFY <hengine> [holmesengine_d96aa.create] Sending task 'dockerplugin.create_for_compo
nents_with_streams'
2017-11-16T03:13:05 CFY <hengine> [holmesengine_d96aa.create] Task started 'dockerplugin.create_for_compo
nents_with_streams'
2017-11-16T03:13:06 CFY <hengine> [holmesengine_d96aa.create] Task succeeded 'dockerplugin.create_for_com
ponents_with_streams'
2017-11-16T03:13:06 CFY <hengine> [holmesengine_d96aa->docker_holmes_host_da27f|preconfigure] Sending tas
k 'relationshipplugin.forward_destination_info'
2017-11-16T03:13:06 CFY <hengine> [holmesengine_d96aa->docker_holmes_host_da27f|preconfigure] Task starte
d 'relationshipplugin.forward_destination_info'
2017-11-16T03:13:07 CFY <hengine> [holmesengine_d96aa->docker_holmes_host_da27f|preconfigure] Task succee
ded 'relationshipplugin.forward_destination_info'
2017-11-16T03:13:07 CFY <hengine> [holmesengine_d96aa] Configuring node
2017-11-16T03:13:07 CFY <hengine> [holmesengine_d96aa] Starting node
2017-11-16T03:13:07 CFY <hengine> [holmesengine_d96aa.start] Sending task 'dockerplugin.create_and_start_
container_for_components_with_streams'
2017-11-16T03:13:07 CFY <hengine> [holmesengine_d96aa.start] Task started 'dockerplugin.create_and_start_
container_for_components_with_streams'
2017-11-16T03:14:17 CFY <hengine> [holmesengine_d96aa.start] Task succeeded 'dockerplugin.create_and_star
t_container_for_components_with_streams'
2017-11-16T03:14:18 CFY <hengine> 'install' workflow execution succeeded
Finished executing workflow install on deployment hengine
* Run 'cfy events list --include-logs --execution-id c226c2af-d5ee-4d12-9dc5-597976f01630' to retrieve th
e execution's events/logs
+ echo 10.12.5.240
+ echo 10.12.5.165
+ rm -f /tmp/ready_to_exit
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
+ '[' '!' -e /tmp/ready_to_exit ']'
+ sleep 30
```

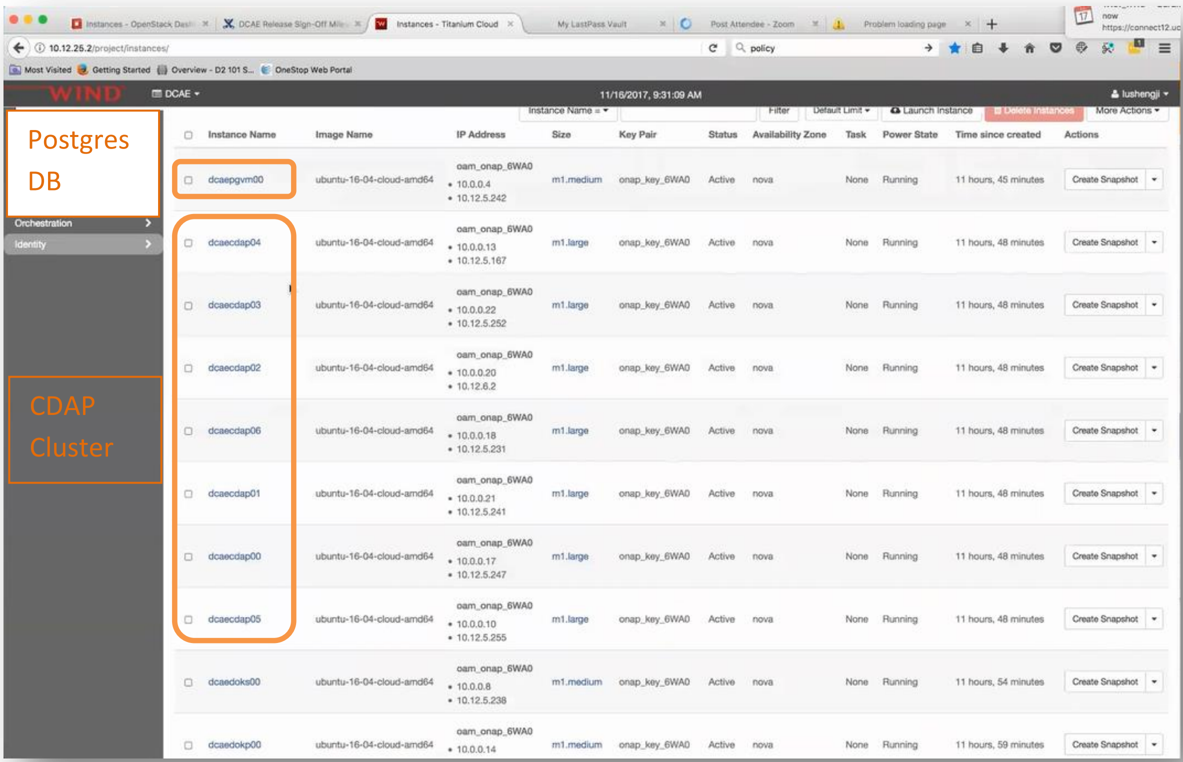
– the reason is that we want the container to be up – we can do things like that, we can log into that and do additional blueprints to deploy, you can go to the container to do that. So in the logs you gonna see a lot of this kind of statements.

Let's go back to a dashboard. You can see what we saw in the log. The first on is the Cloudify Manager (dcaeorcl00),



dcaecdap05	ubuntu-16-04-cloud-amd64	• 10.0.0.10 • 10.12.5.255	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 47 minutes	Create Snapshot
dcaedoks00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.8 • 10.12.5.238	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 53 minutes	Create Snapshot
dcaedokp00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.14 • 10.12.5.44	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 58 minutes	Create Snapshot
dcaecnsl00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.12 • 10.12.5.240	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 1 minute	Create Snapshot
dcaecnsl01	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.6 • 10.12.5.214	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 2 minutes	Create Snapshot
dcaecnsl02	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.7 • 10.12.5.234	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 2 minutes	Create Snapshot
dcaeorcl00	CentOS-7	oam_onap_6WA0 • 10.0.0.3 • 10.12.5.165	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 12 minutes	Create Snapshot
vm1-portal	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.9.1 • 10.12.5.2	m1.large	onap_key_6WA0	Active	nova	None	Running	13 hours, 12 minutes	Create Snapshot
vm1-dcae-bootstrap	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.4.1 • 10.12.5.3	m1.small	onap_key_6WA0	Active	nova	None	Running	13 hours, 12 minutes	Create Snapshot
vm1-sdc	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.3.1 • 10.12.5.12	m1.xlarge	onap_key_6WA0	Active	nova	None	Running	13 hours, 12 minutes	Create Snapshot
vm1-aai-inst1	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.1.1	m1.xlarge	onap_key_6WA0	Active	nova	None	Running	13 hours, 12 minutes	Create Snapshot

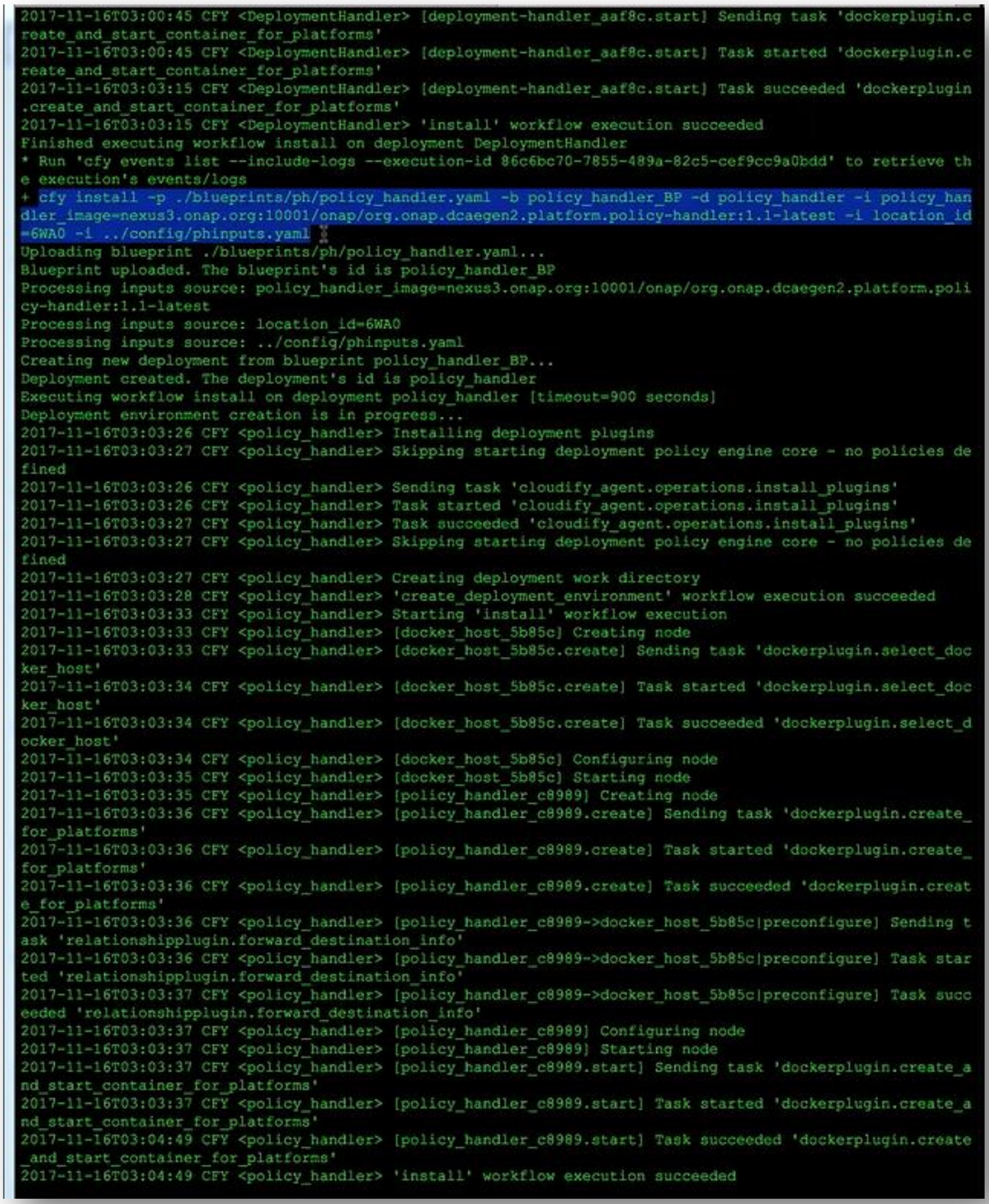
console cluster - 3 VMs cluster (dcaecnsl00, dcaecnsl01 and dcaecnsl02), you have 2 docker hosts (dcaedoks00 and dcaedokp00) then you have the CDAP cluster (dcaecdap00, dcaecdap01, dcaecdap02, dcaecdap03, dcaecdap04, dcaecdap05),



Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
dcaepgvm00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.4 • 10.12.5.242	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 45 minutes	Create Snapshot
dcaecdap04	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.13 • 10.12.5.167	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 48 minutes	Create Snapshot
dcaecdap03	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.22 • 10.12.5.252	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 48 minutes	Create Snapshot
dcaecdap02	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.20 • 10.12.6.2	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 48 minutes	Create Snapshot
dcaecdap06	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.18 • 10.12.5.231	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 48 minutes	Create Snapshot
dcaecdap01	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.21 • 10.12.5.241	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 48 minutes	Create Snapshot
dcaecdap00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.17 • 10.12.5.247	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 48 minutes	Create Snapshot
dcaecdap05	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.10 • 10.12.5.255	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 48 minutes	Create Snapshot
dcaedoks00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.8 • 10.12.5.238	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 54 minutes	Create Snapshot
dcaedokp00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.14	m1.medium	onap_key_6WA0	Active	nova	None	Running	11 hours, 59 minutes	Create Snapshot

there is 1 VM for running postgres database (dcaepgvm00).

Because we are using Cloudify Manager the deployment is actually very flexible – you just need to provide blueprint describing the system you want to deploy, then just call the Cloudify Manager to do that. From the logs you can see that all the CFY commands. For example in the following command:



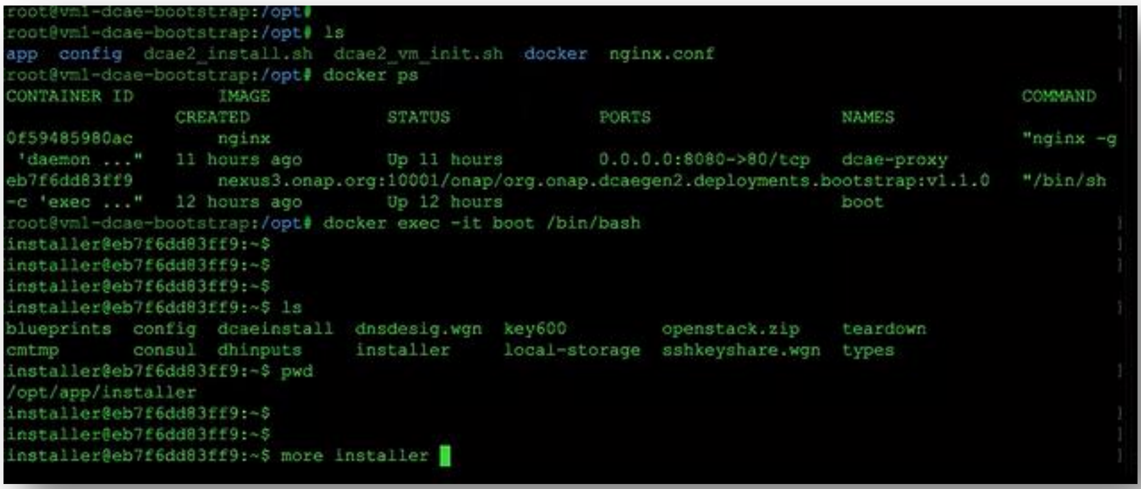
```
2017-11-16T03:00:45 CFY <DeploymentHandler> [deployment-handler_aaf8c.start] Sending task 'dockerplugin.c
reate_and_start_container_for_platforms'
2017-11-16T03:00:45 CFY <DeploymentHandler> [deployment-handler_aaf8c.start] Task started 'dockerplugin.c
reate_and_start_container_for_platforms'
2017-11-16T03:03:15 CFY <DeploymentHandler> [deployment-handler_aaf8c.start] Task succeeded 'dockerplugin
.create_and_start_container_for_platforms'
2017-11-16T03:03:15 CFY <DeploymentHandler> 'install' workflow execution succeeded
Finished executing workflow install on deployment DeploymentHandler
* Run 'cfy events list --include-logs --execution-id 86c6bc70-7855-489a-82c5-cef9cc9a0bdd' to retrieve th
e execution's events/logs
+ cfy install -p ./blueprints/ph/policy_handler.yaml -b policy_handler_BP -d policy_handler -i policy_han
dler_image=nexus3.onap.org:10001/onap/org.onap.dcaegen2.platform.policy-handler:1.1-latest -i location_id
=6WA0 -i ../config/phinputs.yaml
Uploading blueprint ./blueprints/ph/policy_handler.yaml...
Blueprint uploaded. The blueprint's id is policy_handler_BP
Processing inputs source: policy_handler_image=nexus3.onap.org:10001/onap/org.onap.dcaegen2.platform.poli
cy-handler:1.1-latest
Processing inputs source: location_id=6WA0
Processing inputs source: ../config/phinputs.yaml
Creating new deployment from blueprint policy_handler_BP...
Deployment created. The deployment's id is policy_handler
Executing workflow install on deployment policy_handler [timeout=900 seconds]
Deployment environment creation is in progress...
2017-11-16T03:03:26 CFY <policy_handler> Installing deployment plugins
2017-11-16T03:03:27 CFY <policy_handler> Skipping starting deployment policy engine core - no policies de
fined
2017-11-16T03:03:26 CFY <policy_handler> Sending task 'cloudify_agent.operations.install_plugins'
2017-11-16T03:03:26 CFY <policy_handler> Task started 'cloudify_agent.operations.install_plugins'
2017-11-16T03:03:27 CFY <policy_handler> Task succeeded 'cloudify_agent.operations.install_plugins'
2017-11-16T03:03:27 CFY <policy_handler> Skipping starting deployment policy engine core - no policies de
fined
2017-11-16T03:03:27 CFY <policy_handler> Creating deployment work directory
2017-11-16T03:03:28 CFY <policy_handler> 'create_deployment_environment' workflow execution succeeded
2017-11-16T03:03:33 CFY <policy_handler> Starting 'install' workflow execution
2017-11-16T03:03:33 CFY <policy_handler> [docker_host_5b85c] Creating node
2017-11-16T03:03:33 CFY <policy_handler> [docker_host_5b85c.create] Sending task 'dockerplugin.select_doc
ker_host'
2017-11-16T03:03:34 CFY <policy_handler> [docker_host_5b85c.create] Task started 'dockerplugin.select_doc
ker_host'
2017-11-16T03:03:34 CFY <policy_handler> [docker_host_5b85c.create] Task succeeded 'dockerplugin.select_d
ocker_host'
2017-11-16T03:03:34 CFY <policy_handler> [docker_host_5b85c] Configuring node
2017-11-16T03:03:35 CFY <policy_handler> [docker_host_5b85c] Starting node
2017-11-16T03:03:35 CFY <policy_handler> [policy_handler_c8989] Creating node
2017-11-16T03:03:36 CFY <policy_handler> [policy_handler_c8989.create] Sending task 'dockerplugin.create_
for_platforms'
2017-11-16T03:03:36 CFY <policy_handler> [policy_handler_c8989.create] Task started 'dockerplugin.create_
for_platforms'
2017-11-16T03:03:36 CFY <policy_handler> [policy_handler_c8989.create] Task succeeded 'dockerplugin.creat
e_for_platforms'
2017-11-16T03:03:36 CFY <policy_handler> [policy_handler_c8989->docker_host_5b85c|preconfigure] Sending t
ask 'relationshipplugin.forward_destination_info'
2017-11-16T03:03:36 CFY <policy_handler> [policy_handler_c8989->docker_host_5b85c|preconfigure] Task star
ted 'relationshipplugin.forward_destination_info'
2017-11-16T03:03:37 CFY <policy_handler> [policy_handler_c8989->docker_host_5b85c|preconfigure] Task succ
eeded 'relationshipplugin.forward_destination_info'
2017-11-16T03:03:37 CFY <policy_handler> [policy_handler_c8989] Configuring node
2017-11-16T03:03:37 CFY <policy_handler> [policy_handler_c8989] Starting node
2017-11-16T03:03:37 CFY <policy_handler> [policy_handler_c8989.start] Sending task 'dockerplugin.create_a
nd_start_container_for_platforms'
2017-11-16T03:03:37 CFY <policy_handler> [policy_handler_c8989.start] Task started 'dockerplugin.create_a
nd_start_container_for_platforms'
2017-11-16T03:04:49 CFY <policy_handler> [policy_handler_c8989.start] Task succeeded 'dockerplugin.create
_and_start_container_for_platforms'
2017-11-16T03:04:49 CFY <policy_handler> 'install' workflow execution succeeded
```

We are calling Cloudify Manager to deploy the policy handler. For example the current CDAP cluster blueprint that is provided on Nexus it is describing 7 VMs cluster – to deploy a much smaller CDAP cluster, for example 3 VMs, then we just need to provide that 3 VMs blueprint or maybe even just 1 VM – depends on what is needed and what is available. The tradeoff of course is when you have much smaller CDAP then you can provide a more friendlier dev environment however it becomes less production ready so we deal with this different kinds of blueprints used for different environments and long time ago we had the smaller blueprints but because the use experience and feedback from AT&T production side and from vendor recommendations we have hence with 2 o 7 VMs inside of

AT&T that is why we are putting this out as a contribution to ONAP. In the future, especially for earlier releases of ONAP, probably more larger interest is in just for pack and dev way so we work on those kind of things making maybe a dev version configuration of the DCAE. Now in the meanwhile also there are other things that we could configure.

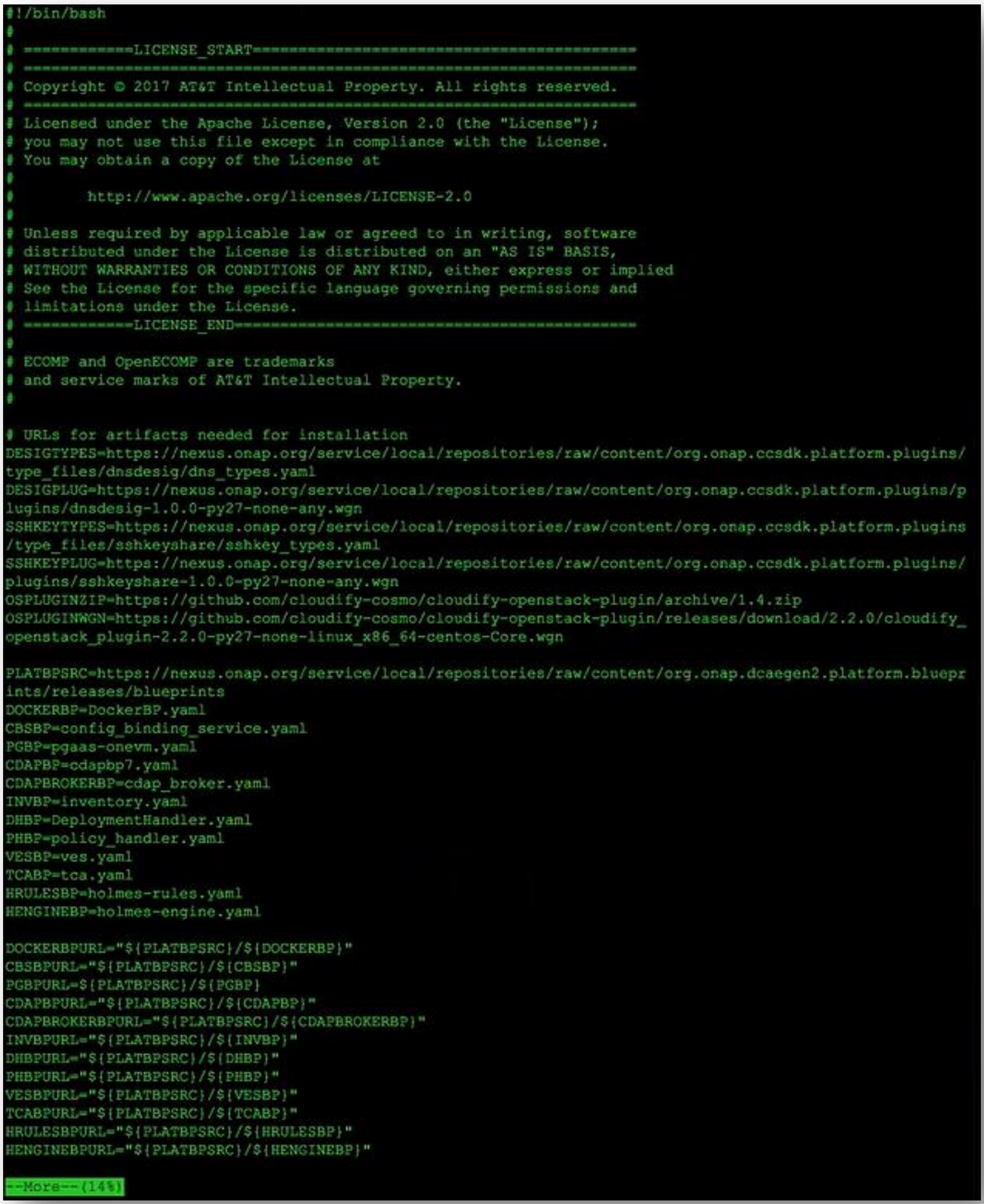
Question: can you please completely show how blueprints needs to be changed?

Yes, that is what I am doing right now. We probably willing to get into the docker because everything is formed there and inside of docker (docker exec –it boot /bin/bash) you can see that there is a file called: installer.

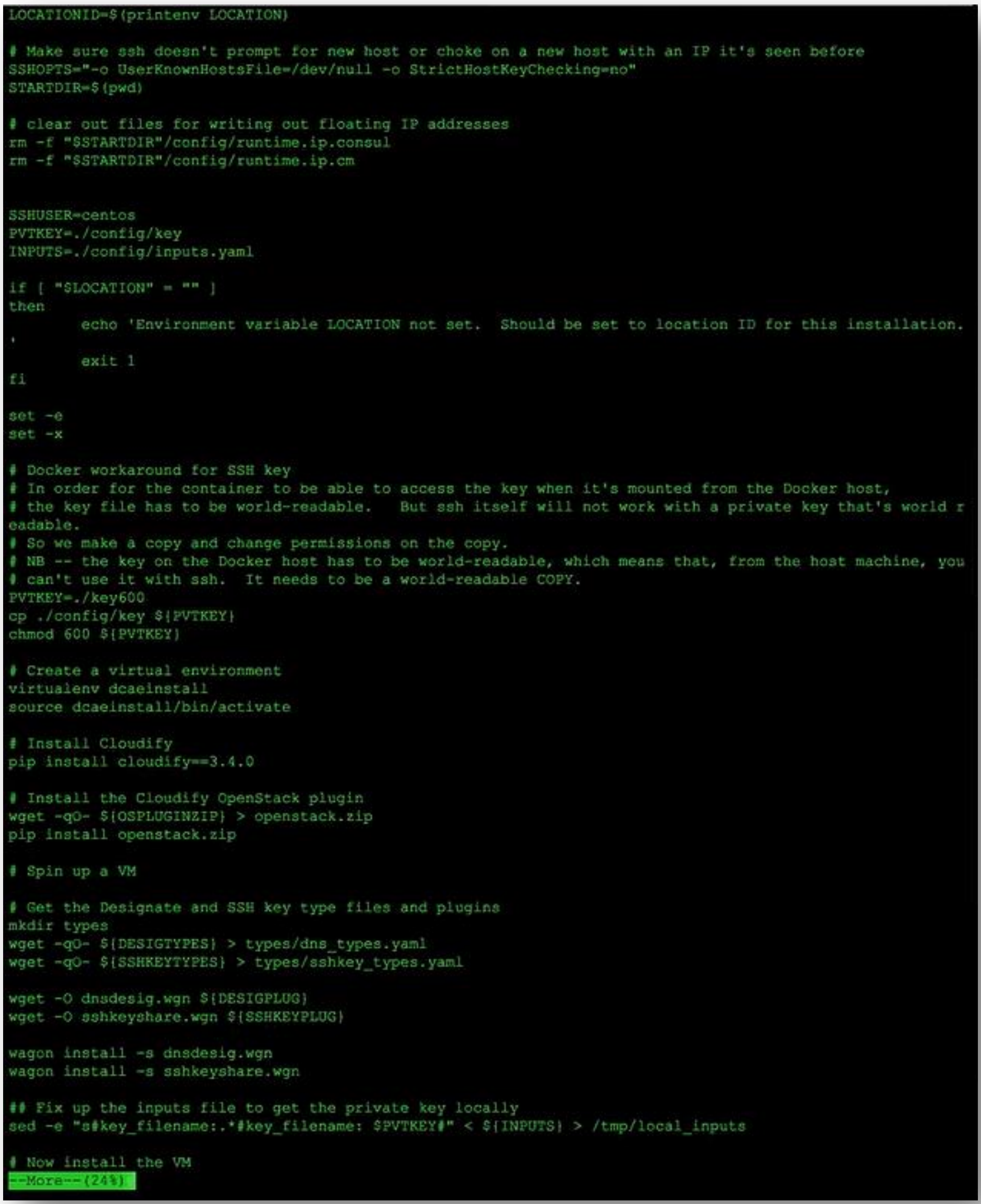


```
root@vm1-dcae-bootstrap:/opt#
root@vm1-dcae-bootstrap:/opt# ls
app  config  dcae2_install.sh  dcae2_vm_init.sh  docker  nginx.conf
root@vm1-dcae-bootstrap:/opt# docker ps
CONTAINER ID        IMAGE                                                                       COMMAND
                CREATED             STATUS              PORTS                  NAMES
0f59485980ac        nginx                                                                       "nginx -g
 'daemon ..."   11 hours ago        Up 11 hours         0.0.0.0:8080->80/tcp   dcae-proxy
eb7f6dd83ff9        nexus3.onap.org:10001/onap/org.onap.dcaegen2.deployments.bootstrap:v1.1.0   "/bin/sh
-c 'exec ..."   12 hours ago        Up 12 hours                                boot
root@vm1-dcae-bootstrap:/opt# docker exec -it boot /bin/bash
installer@eb7f6dd83ff9:~$
installer@eb7f6dd83ff9:~$
installer@eb7f6dd83ff9:~$
installer@eb7f6dd83ff9:~$ ls
blueprints  config  dcaeinstall  dnsdesig.wgn  key600         openstack.zip    teardown
cmtmp       consul  dhinputs     installer     local-storage  sshkeyshare.wgn  types
installer@eb7f6dd83ff9:~$ pwd
/opt/app/installer
installer@eb7f6dd83ff9:~$
installer@eb7f6dd83ff9:~$
installer@eb7f6dd83ff9:~$ more installer
```

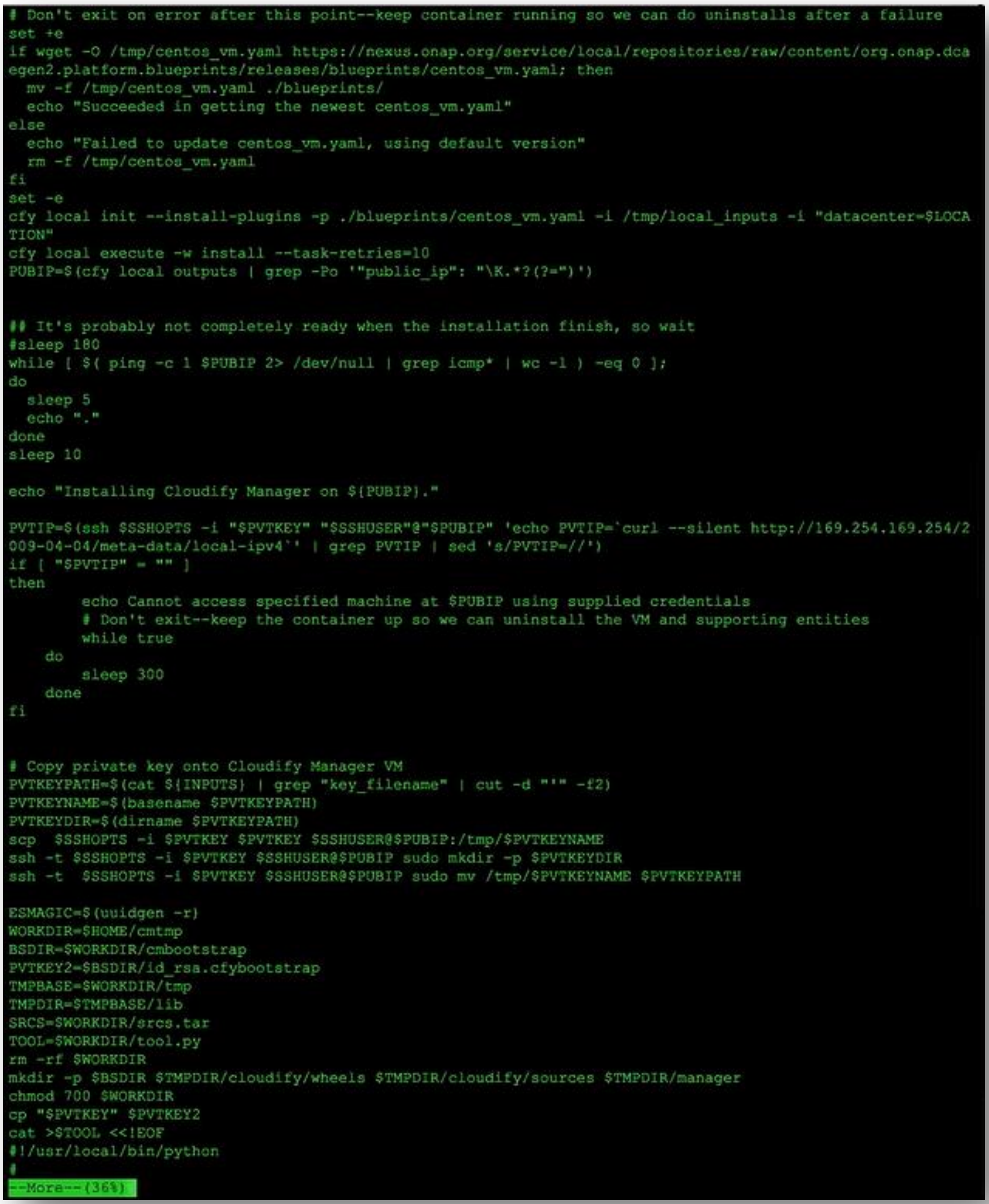
Inside of installer (more installer) is the master installation engine, so you see all those commands,



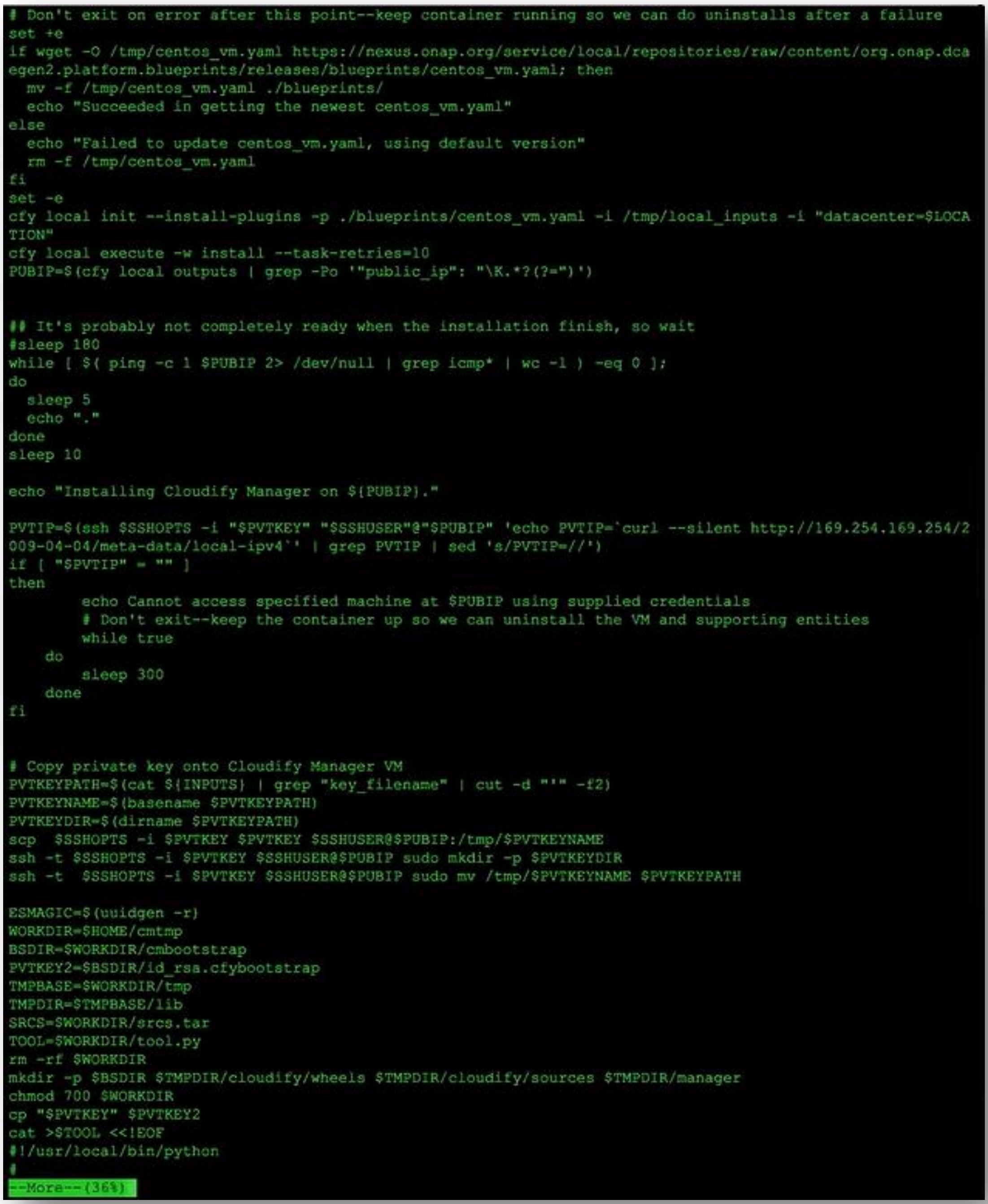
```
#!/bin/bash
#
# ============LICENSE_START=========================================
# ==================================================================
# Copyright © 2017 AT&T Intellectual Property. All rights reserved.
# ==================================================================
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#        http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied
# See the License for the specific language governing permissions and
# limitations under the License.
# ============LICENSE_END===========================================
#
# ECOMP and OpenECOMP are trademarks
# and service marks of AT&T Intellectual Property.
#

# URLs for artifacts needed for installation
DESIGTYPES=https://nexus.onap.org/service/local/repositories/raw/content/org.onap.ccsdk.platform.plugins/
type_files/dnsdesig/dns_types.yaml
DESIGPLUG=https://nexus.onap.org/service/local/repositories/raw/content/org.onap.ccsdk.platform.plugins/p
lugins/dnsdesig-1.0.0-py27-none-any.wgn
SSHKEYTYPES=https://nexus.onap.org/service/local/repositories/raw/content/org.onap.ccsdk.platform.plugins
/type_files/sshkeyshare/sshkey_types.yaml
SSHKEYPLUG=https://nexus.onap.org/service/local/repositories/raw/content/org.onap.ccsdk.platform.plugins/
plugins/sshkeyshare-1.0.0-py27-none-any.wgn
OSPLUGINZIP=https://github.com/cloudify-cosmo/cloudify-openstack-plugin/archive/1.4.zip
OSPLUGINWGN=https://github.com/cloudify-cosmo/cloudify-openstack-plugin/releases/download/2.2.0/cloudify_
openstack_plugin-2.2.0-py27-none-linux_x86_64-centos-Core.wgn

PLATBPSRC=https://nexus.onap.org/service/local/repositories/raw/content/org.onap.dcaegen2.platform.bluepr
ints/releases/blueprints
DOCKERBP=DockerBP.yaml
CBSBP=config_binding_service.yaml
PGBP=pgaas-onevm.yaml
CDAPBP=cdapbp7.yaml
CDAPBROKERBP=cdap_broker.yaml
INVBP=inventory.yaml
DHBP=DeploymentHandler.yaml
PHBP=policy_handler.yaml
VESBP=ves.yaml
TCABP=tca.yaml
HRULESBP=holmes-rules.yaml
HENGINEBP=holmes-engine.yaml

DOCKERBPURL="${PLATBPSRC}/${DOCKERBP}"
CBSBPURL="${PLATBPSRC}/${CBSBP}"
PGBPURL=${PLATBPSRC}/${PGBP}
CDAPBPURL="${PLATBPSRC}/${CDAPBP}"
CDAPBROKERBPURL="${PLATBPSRC}/${CDAPBROKERBP}"
INVBPURL="${PLATBPSRC}/${INVBP}"
DHBPURL="${PLATBPSRC}/${DHBP}"
PHBPURL="${PLATBPSRC}/${PHBP}"
VESBPURL="${PLATBPSRC}/${VESBP}"
TCABPURL="${PLATBPSRC}/${TCABP}"
HRULESBPURL="${PLATBPSRC}/${HRULESBP}"
HENGINEBPURL="${PLATBPSRC}/${HENGINEBP}"

--More--(14%)
```



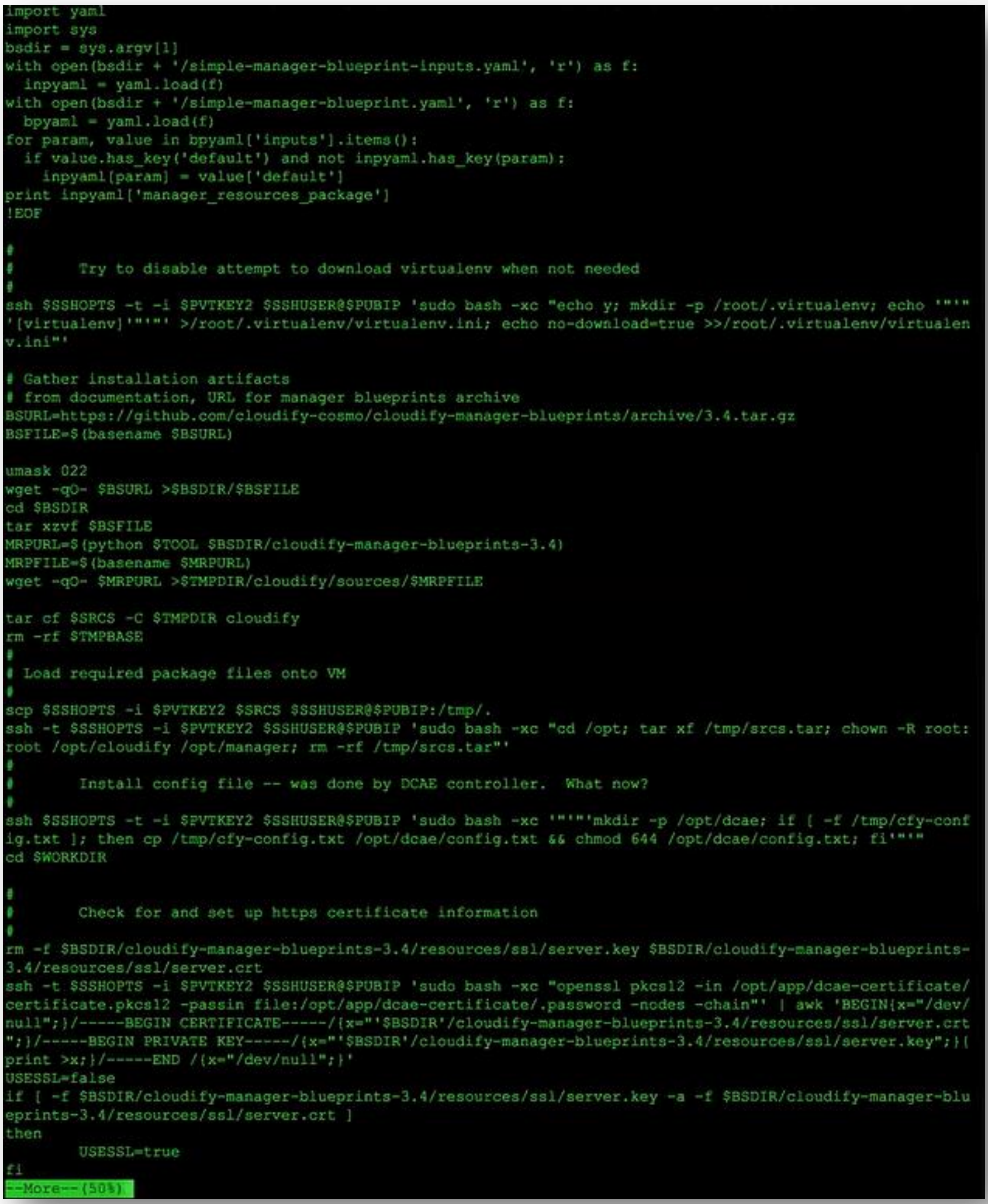
```
LOCATIONID=$(printenv LOCATION)

# Make sure ssh doesn't prompt for new host or choke on a new host with an IP it's seen before
SSHOPTS="-o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no"
STARTDIR=$(pwd)

# clear out files for writing out floating IP addresses
rm -f "$STARTDIR"/config/runtime.ip.consul
rm -f "$STARTDIR"/config/runtime.ip.cm


SSHUSER=centos
PVTKEY=./config/key
INPUTS=./config/inputs.yaml

if [ "$LOCATION" = "" ]
then
        echo 'Environment variable LOCATION not set.  Should be set to location ID for this installation.
'
        exit 1
fi

set -e
set -x

# Docker workaround for SSH key
# In order for the container to be able to access the key when it's mounted from the Docker host,
# the key file has to be world-readable.   But ssh itself will not work with a private key that's world r
eadable.
# So we make a copy and change permissions on the copy.
# NB -- the key on the Docker host has to be world-readable, which means that, from the host machine, you
# can't use it with ssh.  It needs to be a world-readable COPY.
PVTKEY=./key600
cp ./config/key ${PVTKEY}
chmod 600 ${PVTKEY}

# Create a virtual environment
virtualenv dcaeinstall
source dcaeinstall/bin/activate

# Install Cloudify
pip install cloudify==3.4.0

# Install the Cloudify OpenStack plugin
wget -qO- ${OSPLUGINZIP} > openstack.zip
pip install openstack.zip

# Spin up a VM

# Get the Designate and SSH key type files and plugins
mkdir types
wget -qO- ${DESIGTYPES} > types/dns_types.yaml
wget -qO- ${SSHKEYTYPES} > types/sshkey_types.yaml

wget -O dnsdesig.wgn ${DESIGPLUG}
wget -O sshkeyshare.wgn ${SSHKEYPLUG}

wagon install -s dnsdesig.wgn
wagon install -s sshkeyshare.wgn

## Fix up the inputs file to get the private key locally
sed -e "s#key_filename:.*#key_filename: $PVTKEY#" < ${INPUTS} > /tmp/local_inputs

# Now install the VM
--More--(24%)
```



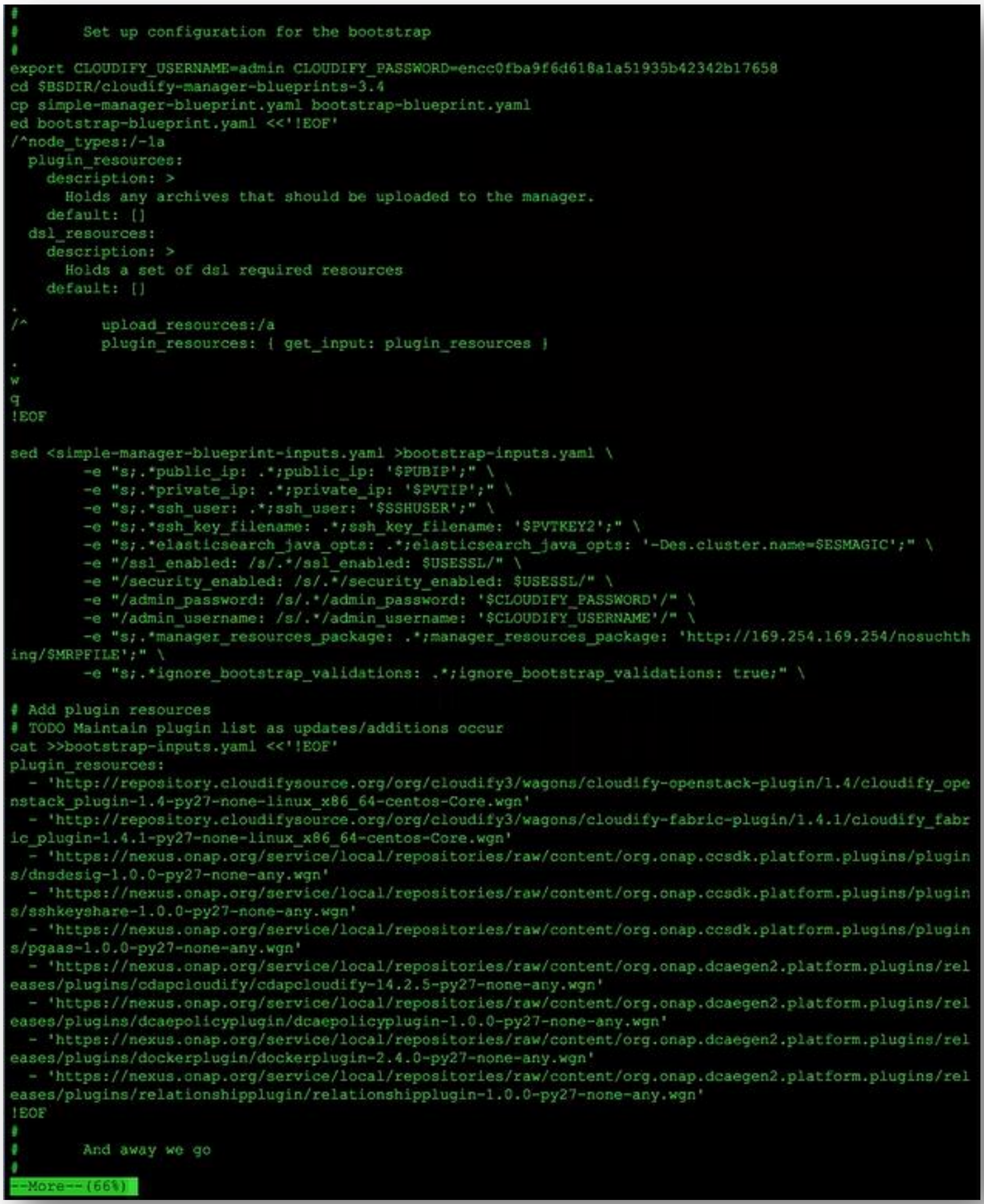
```
# Don't exit on error after this point--keep container running so we can do uninstalls after a failure
set +e
if wget -O /tmp/centos_vm.yaml https://nexus.onap.org/service/local/repositories/raw/content/org.onap.dca
egen2.platform.blueprints/releases/blueprints/centos_vm.yaml; then
  mv -f /tmp/centos_vm.yaml ./blueprints/
  echo "Succeeded in getting the newest centos_vm.yaml"
else
  echo "Failed to update centos_vm.yaml, using default version"
  rm -f /tmp/centos_vm.yaml
fi
set -e
cfy local init --install-plugins -p ./blueprints/centos_vm.yaml -i /tmp/local_inputs -i "datacenter=$LOCA
TION"
cfy local execute -w install --task-retries=10
PUBIP=$(cfy local outputs | grep -Po '"public_ip": "\K.*?(?=")')


## It's probably not completely ready when the installation finish, so wait
#sleep 180
while [ $( ping -c 1 $PUBIP 2> /dev/null | grep icmp* | wc -l ) -eq 0 ];
do
  sleep 5
  echo "."
done
sleep 10

echo "Installing Cloudify Manager on ${PUBIP}."

PVTIP=$(ssh $SSHOPTS -i "$PVTKEY" "$SSHUSER"@"$PUBIP" 'echo PVTIP=`curl --silent http://169.254.169.254/2
009-04-04/meta-data/local-ipv4`' | grep PVTIP | sed 's/PVTIP=//')
if [ "$PVTIP" = "" ]
then
        echo Cannot access specified machine at $PUBIP using supplied credentials
        # Don't exit--keep the container up so we can uninstall the VM and supporting entities
        while true
    do
        sleep 300
    done
fi


# Copy private key onto Cloudify Manager VM
PVTKEYPATH=$(cat ${INPUTS} | grep "key_filename" | cut -d "'" -f2)
PVTKEYNAME=$(basename $PVTKEYPATH)
PVTKEYDIR=$(dirname $PVTKEYPATH)
scp  $SSHOPTS -i $PVTKEY $PVTKEY $SSHUSER@$PUBIP:/tmp/$PVTKEYNAME
ssh -t $SSHOPTS -i $PVTKEY $SSHUSER@$PUBIP sudo mkdir -p $PVTKEYDIR
ssh -t  $SSHOPTS -i $PVTKEY $SSHUSER@$PUBIP sudo mv /tmp/$PVTKEYNAME $PVTKEYPATH

ESMAGIC=$(uuidgen -r)
WORKDIR=$HOME/cmtmp
BSDIR=$WORKDIR/cmbootstrap
PVTKEY2=$BSDIR/id_rsa.cfybootstrap
TMPBASE=$WORKDIR/tmp
TMPDIR=$TMPBASE/lib
SRCS=$WORKDIR/srcs.tar
TOOL=$WORKDIR/tool.py
rm -rf $WORKDIR
mkdir -p $BSDIR $TMPDIR/cloudify/wheels $TMPDIR/cloudify/sources $TMPDIR/manager
chmod 700 $WORKDIR
cp "$PVTKEY" $PVTKEY2
cat >$TOOL <<!EOF
#!/usr/local/bin/python
#
--More--(36%)
```



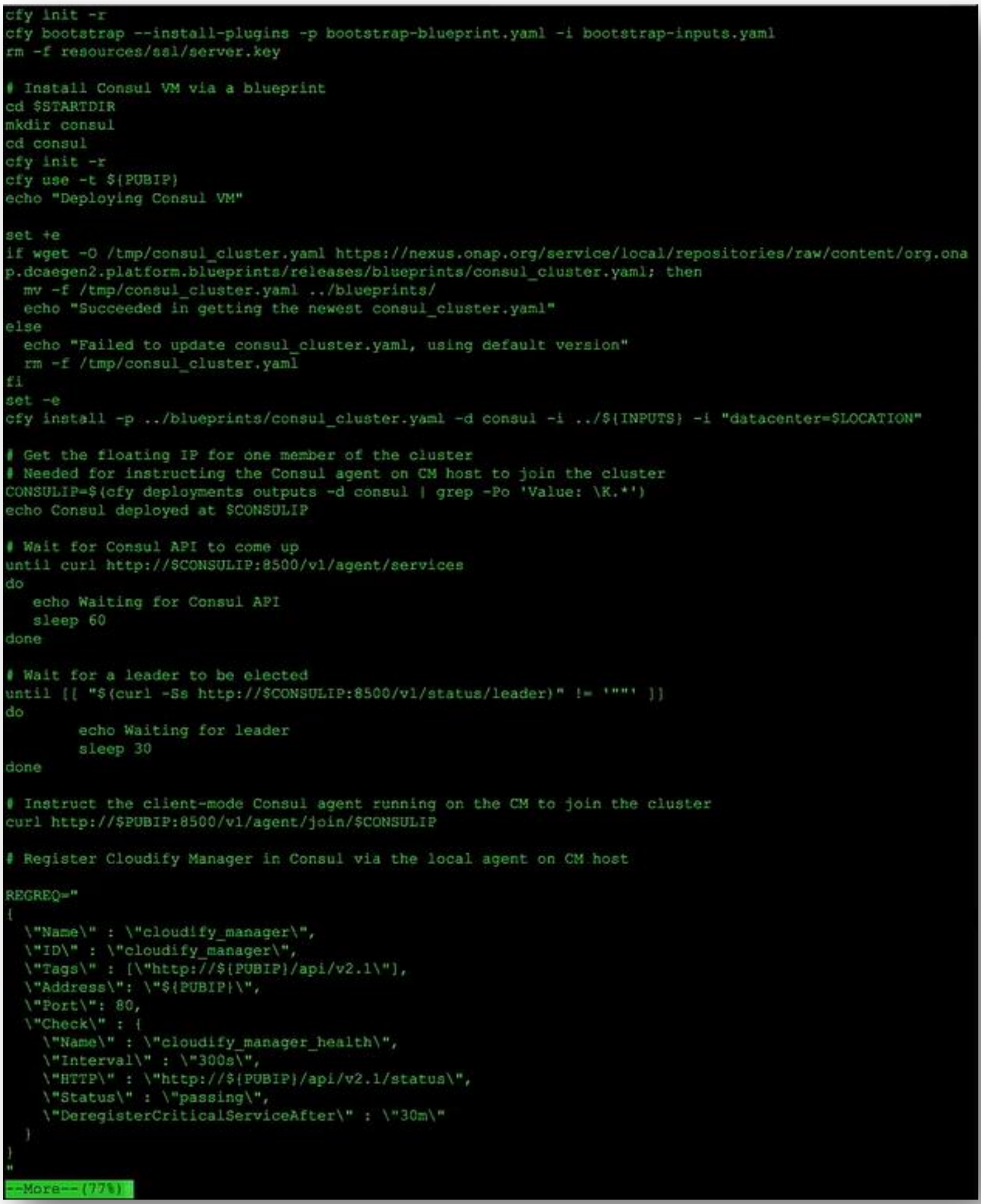
```
# Don't exit on error after this point--keep container running so we can do uninstalls after a failure
set +e
if wget -O /tmp/centos_vm.yaml https://nexus.onap.org/service/local/repositories/raw/content/org.onap.dca
egen2.platform.blueprints/releases/blueprints/centos_vm.yaml; then
  mv -f /tmp/centos_vm.yaml ./blueprints/
  echo "Succeeded in getting the newest centos_vm.yaml"
else
  echo "Failed to update centos_vm.yaml, using default version"
  rm -f /tmp/centos_vm.yaml
fi
set -e
cfy local init --install-plugins -p ./blueprints/centos_vm.yaml -i /tmp/local_inputs -i "datacenter=$LOCA
TION"
cfy local execute -w install --task-retries=10
PUBIP=$(cfy local outputs | grep -Po '"public_ip": "\K.*?(?=")')


## It's probably not completely ready when the installation finish, so wait
#sleep 180
while [ $( ping -c 1 $PUBIP 2> /dev/null | grep icmp* | wc -l ) -eq 0 ];
do
  sleep 5
  echo "."
done
sleep 10

echo "Installing Cloudify Manager on ${PUBIP}."

PVTIP=$(ssh $SSHOPTS -i "$PVTKEY" "$SSHUSER"@"$PUBIP" 'echo PVTIP=`curl --silent http://169.254.169.254/2
009-04-04/meta-data/local-ipv4`' | grep PVTIP | sed 's/PVTIP=//')
if [ "$PVTIP" = "" ]
then
        echo Cannot access specified machine at $PUBIP using supplied credentials
        # Don't exit--keep the container up so we can uninstall the VM and supporting entities
        while true
    do
        sleep 300
    done
fi


# Copy private key onto Cloudify Manager VM
PVTKEYPATH=$(cat ${INPUTS} | grep "key_filename" | cut -d "'" -f2)
PVTKEYNAME=$(basename $PVTKEYPATH)
PVTKEYDIR=$(dirname $PVTKEYPATH)
scp  $SSHOPTS -i $PVTKEY $PVTKEY $SSHUSER@$PUBIP:/tmp/$PVTKEYNAME
ssh -t $SSHOPTS -i $PVTKEY $SSHUSER@$PUBIP sudo mkdir -p $PVTKEYDIR
ssh -t  $SSHOPTS -i $PVTKEY $SSHUSER@$PUBIP sudo mv /tmp/$PVTKEYNAME $PVTKEYPATH

ESMAGIC=$(uuidgen -r)
WORKDIR=$HOME/cmtmp
BSDIR=$WORKDIR/cmbootstrap
PVTKEY2=$BSDIR/id_rsa.cfybootstrap
TMPBASE=$WORKDIR/tmp
TMPDIR=$TMPBASE/lib
SRCS=$WORKDIR/srcs.tar
TOOL=$WORKDIR/tool.py
rm -rf $WORKDIR
mkdir -p $BSDIR $TMPDIR/cloudify/wheels $TMPDIR/cloudify/sources $TMPDIR/manager
chmod 700 $WORKDIR
cp "$PVTKEY" $PVTKEY2
cat >$TOOL <<!EOF
#!/usr/local/bin/python
#
--More--(36%)
```



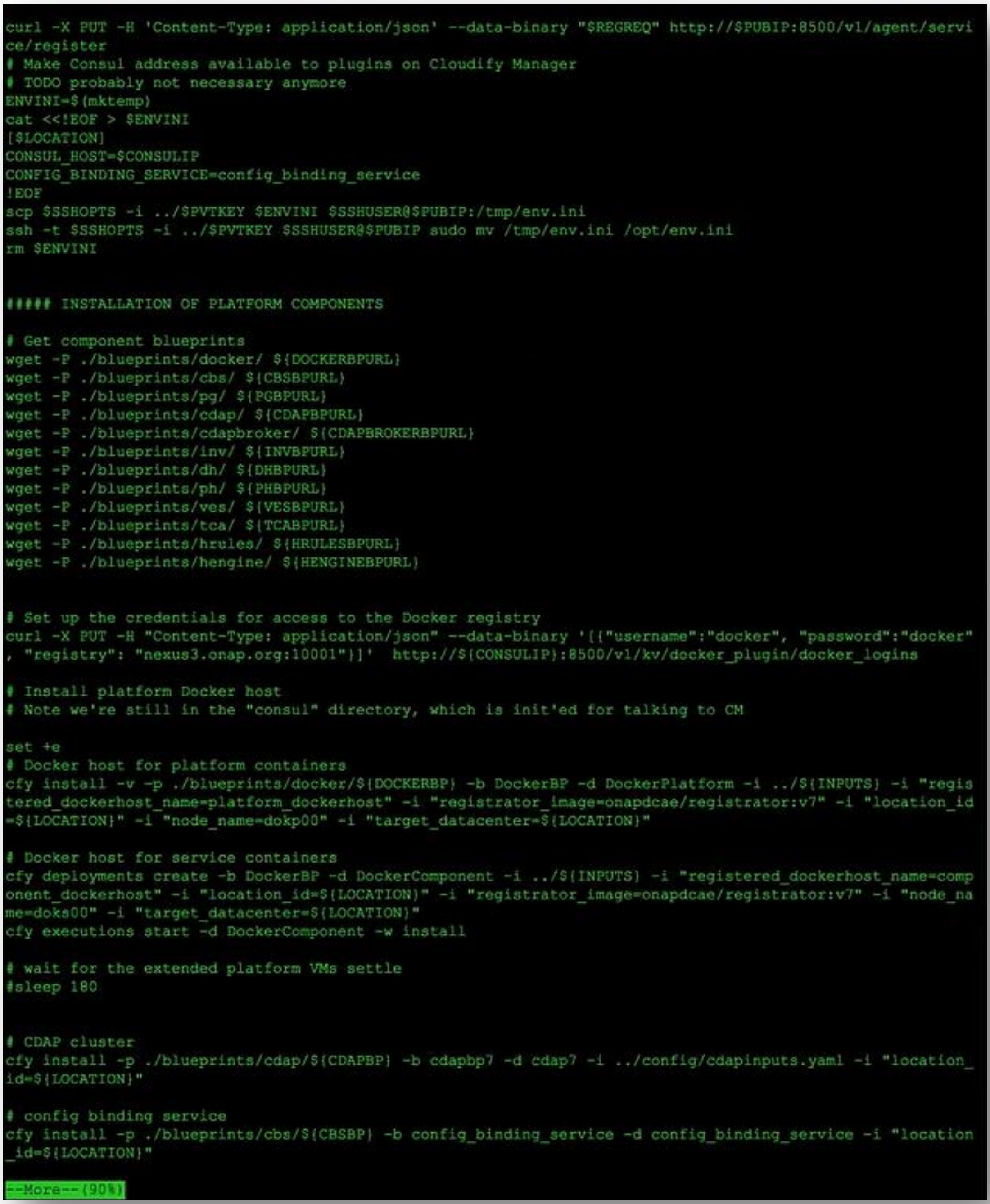
```
import yaml
import sys
bsdir = sys.argv[1]
with open(bsdir + '/simple-manager-blueprint-inputs.yaml', 'r') as f:
  inpyaml = yaml.load(f)
with open(bsdir + '/simple-manager-blueprint.yaml', 'r') as f:
  bpyaml = yaml.load(f)
for param, value in bpyaml['inputs'].items():
  if value.has_key('default') and not inpyaml.has_key(param):
    inpyaml[param] = value['default']
print inpyaml['manager_resources_package']
!EOF

#
#       Try to disable attempt to download virtualenv when not needed
#
ssh $SSHOPTS -t -i $PVTKEY2 $SSHUSER@$PUBIP 'sudo bash -xc "echo y; mkdir -p /root/.virtualenv; echo '"'"'
'[virtualenv]'"'"' >/root/.virtualenv/virtualenv.ini; echo no-download=true >>/root/.virtualenv/virtualen
v.ini"'

# Gather installation artifacts
# from documentation, URL for manager blueprints archive
BSURL=https://github.com/cloudify-cosmo/cloudify-manager-blueprints/archive/3.4.tar.gz
BSFILE=$(basename $BSURL)

umask 022
wget -qO- $BSURL >$BSDIR/$BSFILE
cd $BSDIR
tar xzvf $BSFILE
MRPURL=$(python $TOOL $BSDIR/cloudify-manager-blueprints-3.4)
MRPFILE=$(basename $MRPURL)
wget -qO- $MRPURL >$TMPDIR/cloudify/sources/$MRPFILE

tar cf $SRCS -C $TMPDIR cloudify
rm -rf $TMPBASE
#
# Load required package files onto VM
#
scp $SSHOPTS -i $PVTKEY2 $SRCS $SSHUSER@$PUBIP:/tmp/.
ssh -t $SSHOPTS -i $PVTKEY2 $SSHUSER@$PUBIP 'sudo bash -xc "cd /opt; tar xf /tmp/srcs.tar; chown -R root:
root /opt/cloudify /opt/manager; rm -rf /tmp/srcs.tar"'
#
#       Install config file -- was done by DCAE controller.  What now?
#
ssh $SSHOPTS -t -i $PVTKEY2 $SSHUSER@$PUBIP 'sudo bash -xc '"'"'mkdir -p /opt/dcae; if [ -f /tmp/cfy-conf
ig.txt ]; then cp /tmp/cfy-config.txt /opt/dcae/config.txt && chmod 644 /opt/dcae/config.txt; fi'"'"
cd $WORKDIR

#
#       Check for and set up https certificate information
#
rm -f $BSDIR/cloudify-manager-blueprints-3.4/resources/ssl/server.key $BSDIR/cloudify-manager-blueprints-
3.4/resources/ssl/server.crt
ssh -t $SSHOPTS -i $PVTKEY2 $SSHUSER@$PUBIP 'sudo bash -xc "openssl pkcs12 -in /opt/app/dcae-certificate/
certificate.pkcs12 -passin file:/opt/app/dcae-certificate/.password -nodes -chain"' | awk 'BEGIN{x="/dev/
null";}/-----BEGIN CERTIFICATE-----/{x="'$BSDIR'/cloudify-manager-blueprints-3.4/resources/ssl/server.crt
";}/-----BEGIN PRIVATE KEY-----/{x="'$BSDIR'/cloudify-manager-blueprints-3.4/resources/ssl/server.key";}{
print >x;}/-----END /{x="/dev/null";}'
USESSL=false
if [ -f $BSDIR/cloudify-manager-blueprints-3.4/resources/ssl/server.key -a -f $BSDIR/cloudify-manager-blu
eprints-3.4/resources/ssl/server.crt ]
then
        USESSL=true
fi
--More--(50%)
```



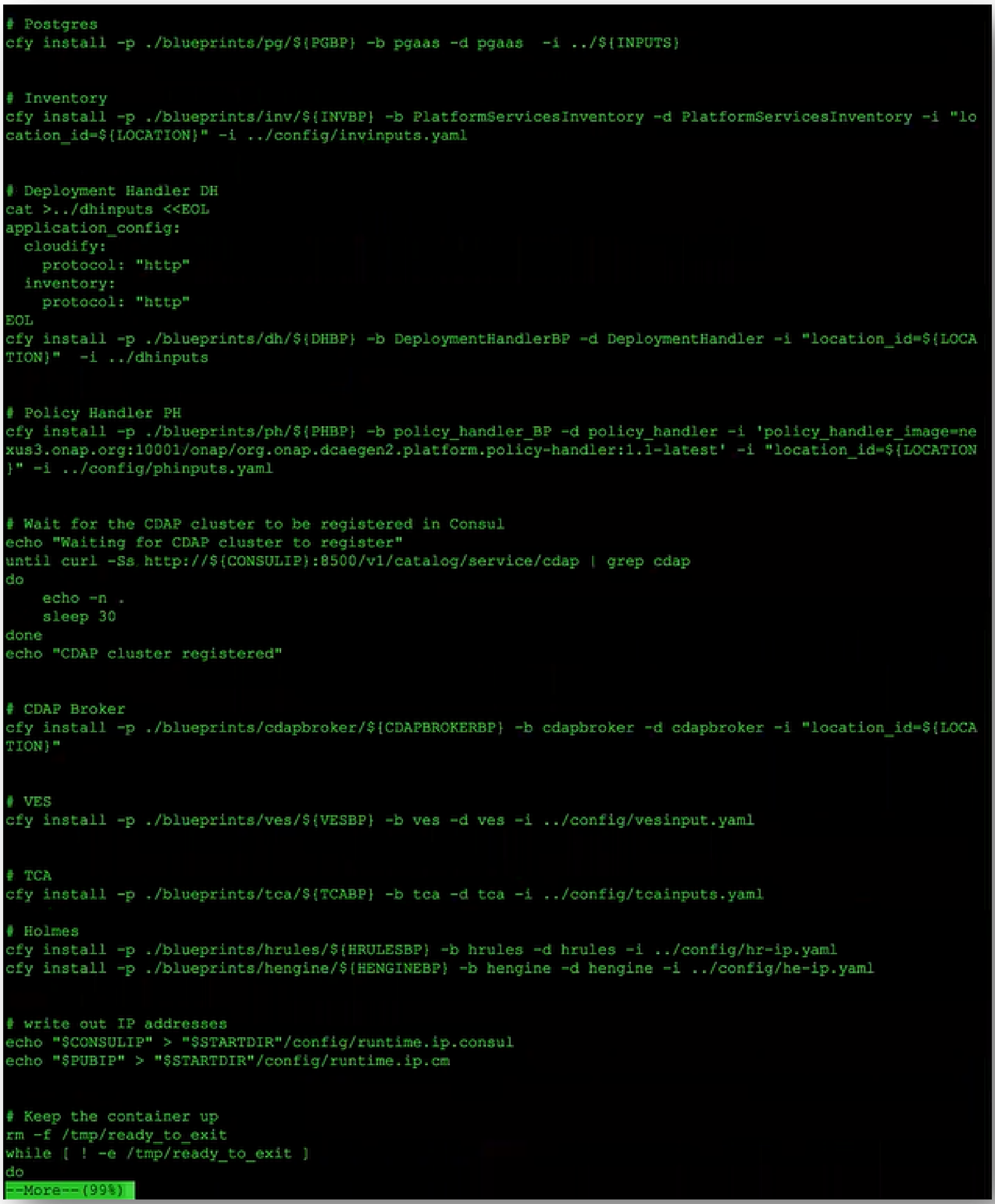
```
#
#       Set up configuration for the bootstrap
#
export CLOUDIFY_USERNAME=admin CLOUDIFY_PASSWORD=encc0fba9f6d618a1a51935b42342b17658
cd $BSDIR/cloudify-manager-blueprints-3.4
cp simple-manager-blueprint.yaml bootstrap-blueprint.yaml
ed bootstrap-blueprint.yaml <<'!EOF'
/^node_types:/-1a
  plugin_resources:
    description: >
      Holds any archives that should be uploaded to the manager.
    default: []
  dsl_resources:
    description: >
      Holds a set of dsl required resources
    default: []
.
/^        upload_resources:/a
          plugin_resources: { get_input: plugin_resources }
.
w
q
!EOF

sed <simple-manager-blueprint-inputs.yaml >bootstrap-inputs.yaml \
        -e "s;.*public_ip: .*;public_ip: '$PUBIP';" \
        -e "s;.*private_ip: .*;private_ip: '$PVTIP';" \
        -e "s;.*ssh_user: .*;ssh_user: '$SSHUSER';" \
        -e "s;.*ssh_key_filename: .*;ssh_key_filename: '$PVTKEY2';" \
        -e "s;.*elasticsearch_java_opts: .*;elasticsearch_java_opts: '-Des.cluster.name=$ESMAGIC';" \
        -e "/ssl_enabled: /s/.*/ssl_enabled: $USESSL/" \
        -e "/security_enabled: /s/.*/security_enabled: $USESSL/" \
        -e "/admin_password: /s/.*/admin_password: '$CLOUDIFY_PASSWORD'/" \
        -e "/admin_username: /s/.*/admin_username: '$CLOUDIFY_USERNAME'/" \
        -e "s;.*manager_resources_package: .*;manager_resources_package: 'http://169.254.169.254/nosuchth
ing/$MRPFILE';" \
        -e "s;.*ignore_bootstrap_validations: .*;ignore_bootstrap_validations: true;" \

# Add plugin resources
# TODO Maintain plugin list as updates/additions occur
cat >>bootstrap-inputs.yaml <<'!EOF'
plugin_resources:
  - 'http://repository.cloudifysource.org/org/cloudify3/wagons/cloudify-openstack-plugin/1.4/cloudify_ope
nstack_plugin-1.4-py27-none-linux_x86_64-centos-Core.wgn'
  - 'http://repository.cloudifysource.org/org/cloudify3/wagons/cloudify-fabric-plugin/1.4.1/cloudify_fabr
ic_plugin-1.4.1-py27-none-linux_x86_64-centos-Core.wgn'
  - 'https://nexus.onap.org/service/local/repositories/raw/content/org.onap.ccsdk.platform.plugins/plugin
s/dnsdesig-1.0.0-py27-none-any.wgn'
  - 'https://nexus.onap.org/service/local/repositories/raw/content/org.onap.ccsdk.platform.plugins/plugin
s/sshkeyshare-1.0.0-py27-none-any.wgn'
  - 'https://nexus.onap.org/service/local/repositories/raw/content/org.onap.ccsdk.platform.plugins/plugin
s/pgaas-1.0.0-py27-none-any.wgn'
  - 'https://nexus.onap.org/service/local/repositories/raw/content/org.onap.dcaegen2.platform.plugins/rel
eases/plugins/cdapcloudify/cdapcloudify-14.2.5-py27-none-any.wgn'
  - 'https://nexus.onap.org/service/local/repositories/raw/content/org.onap.dcaegen2.platform.plugins/rel
eases/plugins/dcaepolicyplugin/dcaepolicyplugin-1.0.0-py27-none-any.wgn'
  - 'https://nexus.onap.org/service/local/repositories/raw/content/org.onap.dcaegen2.platform.plugins/rel
eases/plugins/dockerplugin/dockerplugin-2.4.0-py27-none-any.wgn'
  - 'https://nexus.onap.org/service/local/repositories/raw/content/org.onap.dcaegen2.platform.plugins/rel
eases/plugins/relationshipplugin/relationshipplugin-1.0.0-py27-none-any.wgn'
!EOF
#
#       And away we go
#
--More--(66%)
```



```
cfy init -r
cfy bootstrap --install-plugins -p bootstrap-blueprint.yaml -i bootstrap-inputs.yaml
rm -f resources/ssl/server.key

# Install Consul VM via a blueprint
cd $STARTDIR
mkdir consul
cd consul
cfy init -r
cfy use -t ${PUBIP}
echo "Deploying Consul VM"

set +e
if wget -O /tmp/consul_cluster.yaml https://nexus.onap.org/service/local/repositories/raw/content/org.ona
p.dcaegen2.platform.blueprints/releases/blueprints/consul_cluster.yaml; then
  mv -f /tmp/consul_cluster.yaml ../blueprints/
  echo "Succeeded in getting the newest consul_cluster.yaml"
else
  echo "Failed to update consul_cluster.yaml, using default version"
  rm -f /tmp/consul_cluster.yaml
fi
set -e
cfy install -p ../blueprints/consul_cluster.yaml -d consul -i ../${INPUTS} -i "datacenter=$LOCATION"

# Get the floating IP for one member of the cluster
# Needed for instructing the Consul agent on CM host to join the cluster
CONSULIP=$(cfy deployments outputs -d consul | grep -Po 'Value: \K.*')
echo Consul deployed at $CONSULIP

# Wait for Consul API to come up
until curl http://$CONSULIP:8500/v1/agent/services
do
   echo Waiting for Consul API
   sleep 60
done

# Wait for a leader to be elected
until [[ "$(curl -Ss http://$CONSULIP:8500/v1/status/leader)" != '""' ]]
do
        echo Waiting for leader
        sleep 30
done

# Instruct the client-mode Consul agent running on the CM to join the cluster
curl http://$PUBIP:8500/v1/agent/join/$CONSULIP

# Register Cloudify Manager in Consul via the local agent on CM host

REGREQ="
{
  \"Name\" : \"cloudify_manager\",
  \"ID\" : \"cloudify_manager\",
  \"Tags\" : [\"http://${PUBIP}/api/v2.1\"],
  \"Address\": \"${PUBIP}\",
  \"Port\": 80,
  \"Check\" : {
    \"Name\" : \"cloudify_manager_health\",
    \"Interval\" : \"300s\",
    \"HTTP\" : \"http://${PUBIP}/api/v2.1/status\",
    \"Status\" : \"passing\",
    \"DeregisterCriticalServiceAfter\" : \"30m\"
  }
}
"
--More--(77%)
```

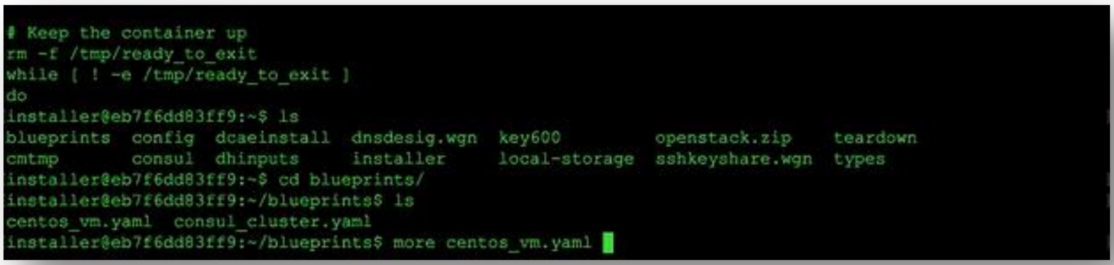


```
curl -X PUT -H 'Content-Type: application/json' --data-binary "$REGREQ" http://$PUBIP:8500/v1/agent/servi
ce/register
# Make Consul address available to plugins on Cloudify Manager
# TODO probably not necessary anymore
ENVINI=$(mktemp)
cat <<!EOF > $ENVINI
[$LOCATION]
CONSUL_HOST=$CONSULIP
CONFIG_BINDING_SERVICE=config_binding_service
!EOF
scp $SSHOPTS -i ../$PVTKEY $ENVINI $SSHUSER@$PUBIP:/tmp/env.ini
ssh -t $SSHOPTS -i ../$PVTKEY $SSHUSER@$PUBIP sudo mv /tmp/env.ini /opt/env.ini
rm $ENVINI


##### INSTALLATION OF PLATFORM COMPONENTS

# Get component blueprints
wget -P ./blueprints/docker/ ${DOCKERBPURL}
wget -P ./blueprints/cbs/ ${CBSBPURL}
wget -P ./blueprints/pg/ ${PGBPURL}
wget -P ./blueprints/cdap/ ${CDAPBPURL}
wget -P ./blueprints/cdapbroker/ ${CDAPBROKERBPURL}
wget -P ./blueprints/inv/ ${INVBPURL}
wget -P ./blueprints/dh/ ${DHBPURL}
wget -P ./blueprints/ph/ ${PHBPURL}
wget -P ./blueprints/ves/ ${VESBPURL}
wget -P ./blueprints/tca/ ${TCABPURL}
wget -P ./blueprints/hrules/ ${HRULESBPURL}
wget -P ./blueprints/hengine/ ${HENGINEBPURL}


# Set up the credentials for access to the Docker registry
curl -X PUT -H "Content-Type: application/json" --data-binary '[{"username":"docker", "password":"docker"
, "registry": "nexus3.onap.org:10001"}]'  http://${CONSULIP}:8500/v1/kv/docker_plugin/docker_logins

# Install platform Docker host
# Note we're still in the "consul" directory, which is init'ed for talking to CM

set +e
# Docker host for platform containers
cfy install -v -p ./blueprints/docker/${DOCKERBP} -b DockerBP -d DockerPlatform -i ../${INPUTS} -i "regis
tered_dockerhost_name=platform_dockerhost" -i "registrator_image=onapdcae/registrator:v7" -i "location_id
=${LOCATION}" -i "node_name=dokp00" -i "target_datacenter=${LOCATION}"

# Docker host for service containers
cfy deployments create -b DockerBP -d DockerComponent -i ../${INPUTS} -i "registered_dockerhost_name=comp
onent_dockerhost" -i "location_id=${LOCATION}" -i "registrator_image=onapdcae/registrator:v7" -i "node_na
me=doks00" -i "target_datacenter=${LOCATION}"
cfy executions start -d DockerComponent -w install

# wait for the extended platform VMs settle
#sleep 180


# CDAP cluster
cfy install -p ./blueprints/cdap/${CDAPBP} -b cdapbp7 -d cdap7 -i ../config/cdapinputs.yaml -i "location_
id=${LOCATION}"

# config binding service
cfy install -p ./blueprints/cbs/${CBSBP} -b config_binding_service -d config_binding_service -i "location
_id=${LOCATION}"

--More--(90%)
```

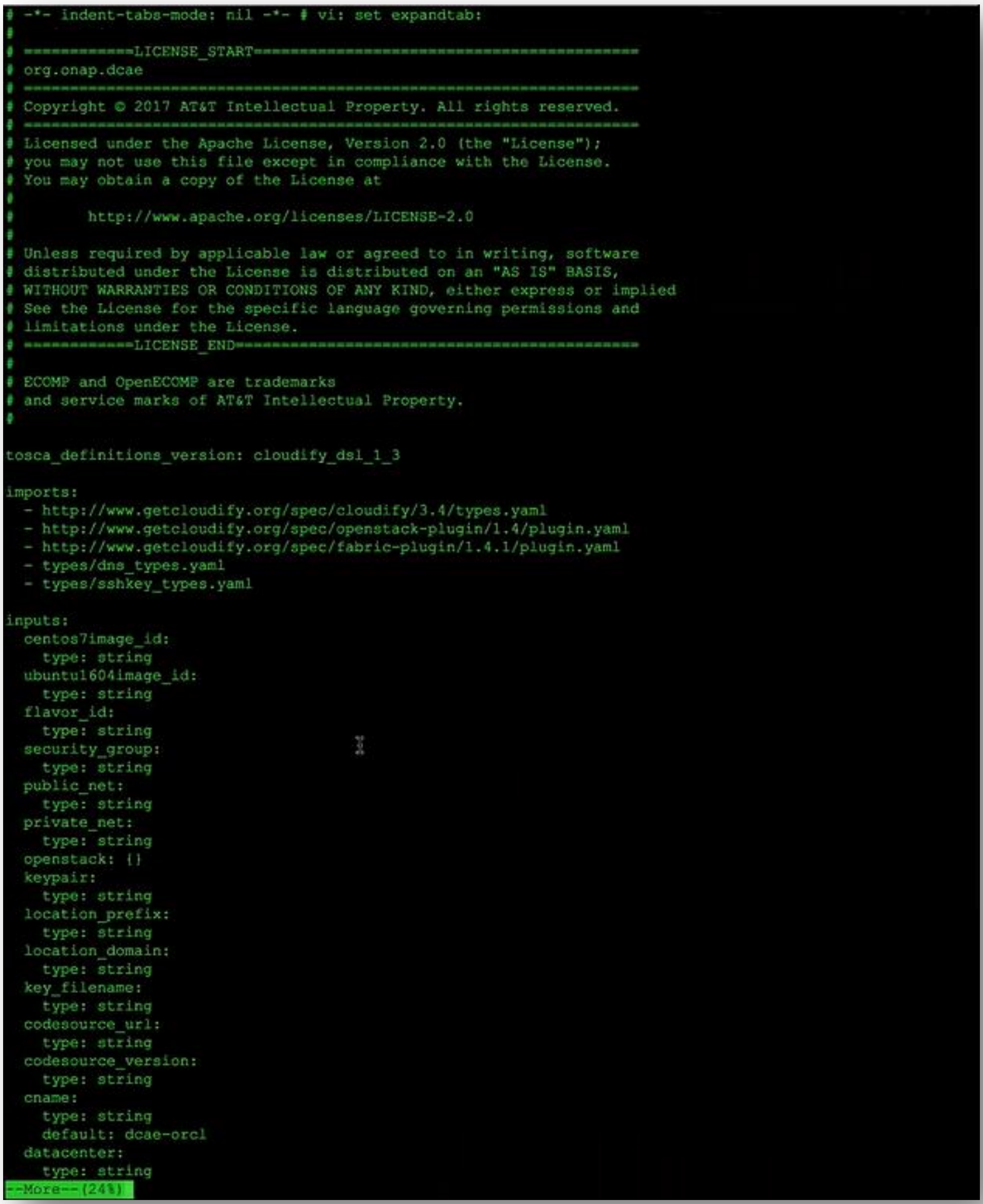


```
# Postgres
cfy install -p ./blueprints/pg/${PGBP} -b pgaas -d pgaas  -i ../${INPUTS}


# Inventory
cfy install -p ./blueprints/inv/${INVBP} -b PlatformServicesInventory -d PlatformServicesInventory -i "lo
cation_id=${LOCATION}" -i ../config/invinputs.yaml


# Deployment Handler DH
cat >../dhinputs <<EOL
application_config:
  cloudify:
    protocol: "http"
  inventory:
    protocol: "http"
EOL
cfy install -p ./blueprints/dh/${DHBP} -b DeploymentHandlerBP -d DeploymentHandler -i "location_id=${LOCA
TION}"  -i ../dhinputs


# Policy Handler PH
cfy install -p ./blueprints/ph/${PHBP} -b policy_handler_BP -d policy_handler -i 'policy_handler_image=ne
xus3.onap.org:10001/onap/org.onap.dcaegen2.platform.policy-handler:1.1-latest' -i "location_id=${LOCATION
}" -i ../config/phinputs.yaml


# Wait for the CDAP cluster to be registered in Consul
echo "Waiting for CDAP cluster to register"
until curl -Ss http://${CONSULIP}:8500/v1/catalog/service/cdap | grep cdap
do
    echo -n .
    sleep 30
done
echo "CDAP cluster registered"


# CDAP Broker
cfy install -p ./blueprints/cdapbroker/${CDAPBROKERBP} -b cdapbroker -d cdapbroker -i "location_id=${LOCA
TION}"


# VES
cfy install -p ./blueprints/ves/${VESBP} -b ves -d ves -i ../config/vesinput.yaml


# TCA
cfy install -p ./blueprints/tca/${TCABP} -b tca -d tca -i ../config/tcainputs.yaml

# Holmes
cfy install -p ./blueprints/hrules/${HRULESBP} -b hrules -d hrules -i ../config/hr-ip.yaml
cfy install -p ./blueprints/hengine/${HENGINEBP} -b hengine -d hengine -i ../config/he-ip.yaml


# write out IP addresses
echo "$CONSULIP" > "$STARTDIR"/config/runtime.ip.consul
echo "$PUBIP" > "$STARTDIR"/config/runtime.ip.cm


# Keep the container up
rm -f /tmp/ready_to_exit
while [ ! -e /tmp/ready_to_exit ]
do
--More--(99%)
```

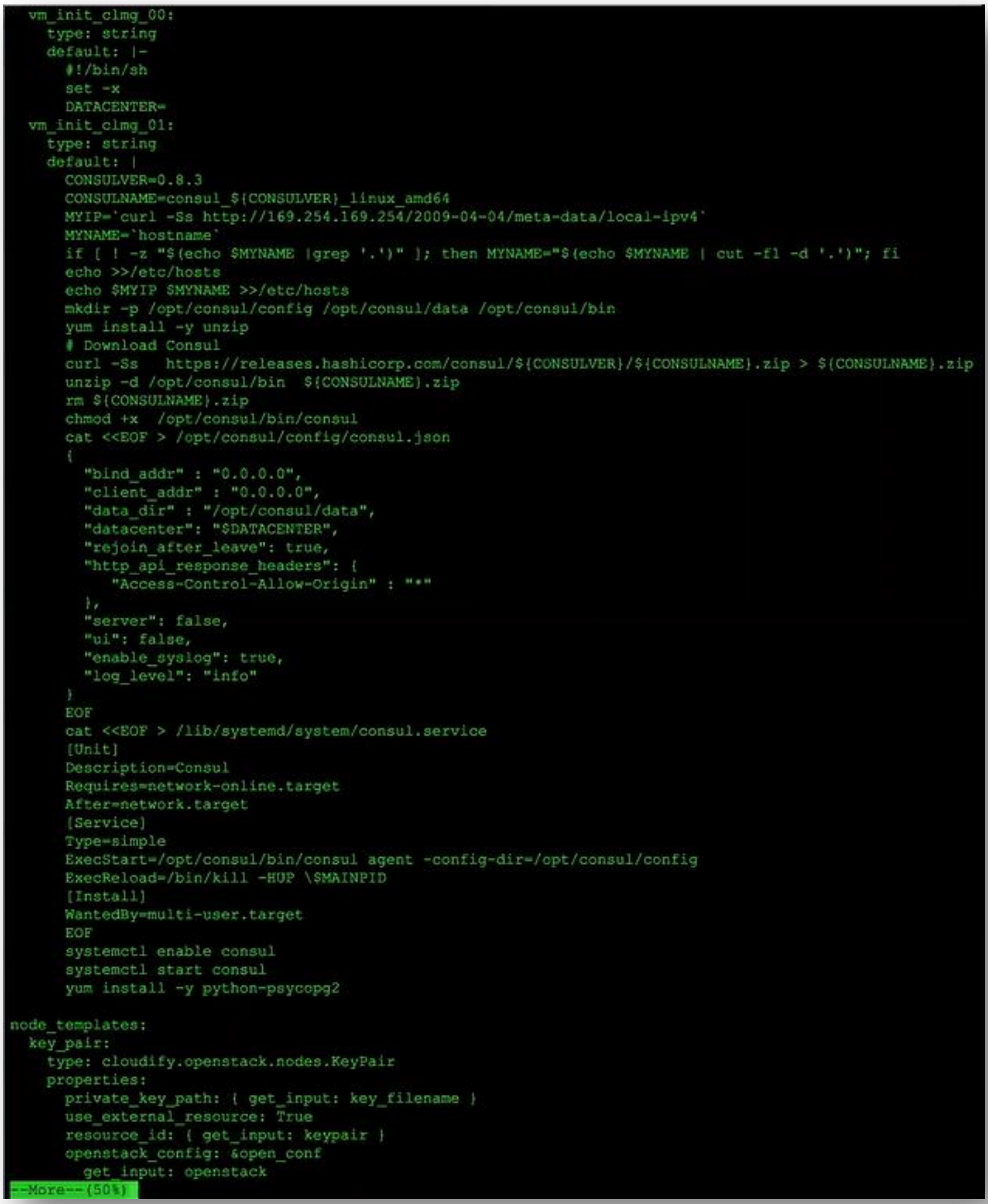
we saw the logs earlier, really all is triggered right here so if you want to get look for the next detail level of information for how DCAE deploys that it the script you want to start with. And for the blueprints there is a folder blueprints right here- there are 2 blueprints we use – that is - they really form the core of DCAE platform: you have centos_vm.yaml which is the blueprint for the Cloudify Manager



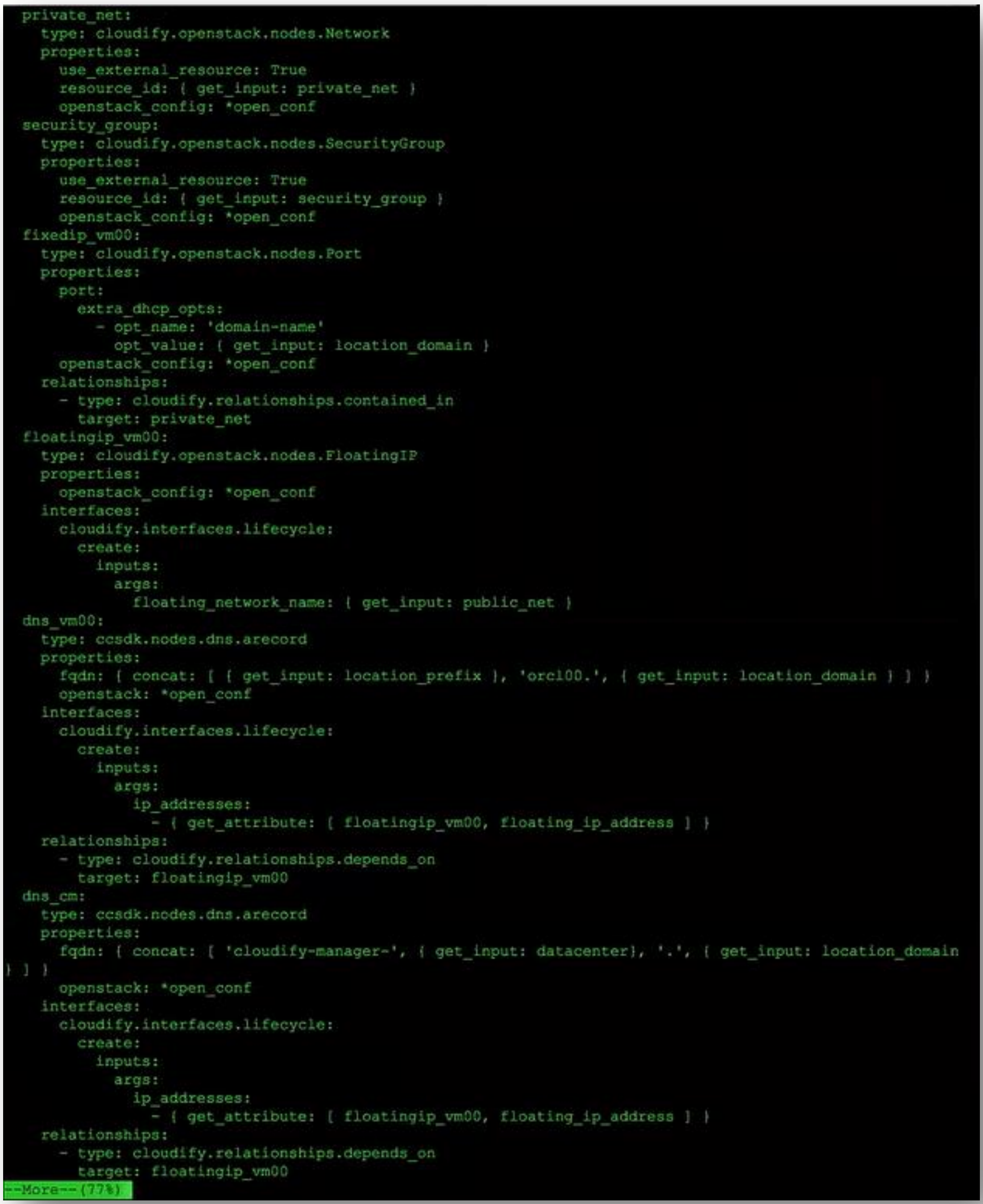
```
# Keep the container up
rm -f /tmp/ready_to_exit
while [ ! -e /tmp/ready_to_exit ]
do
installer@eb7f6dd83ff9:~$ ls
blueprints  config  dcaeinstall  dnsdesig.wgn  key600         openstack.zip    teardown
cmtmp       consul  dhinputs     installer     local-storage  sshkeyshare.wgn  types
installer@eb7f6dd83ff9:~$ cd blueprints/
installer@eb7f6dd83ff9:~/blueprints$ ls
centos_vm.yaml  consul_cluster.yaml
installer@eb7f6dd83ff9:~/blueprints$ more centos_vm.yaml
```



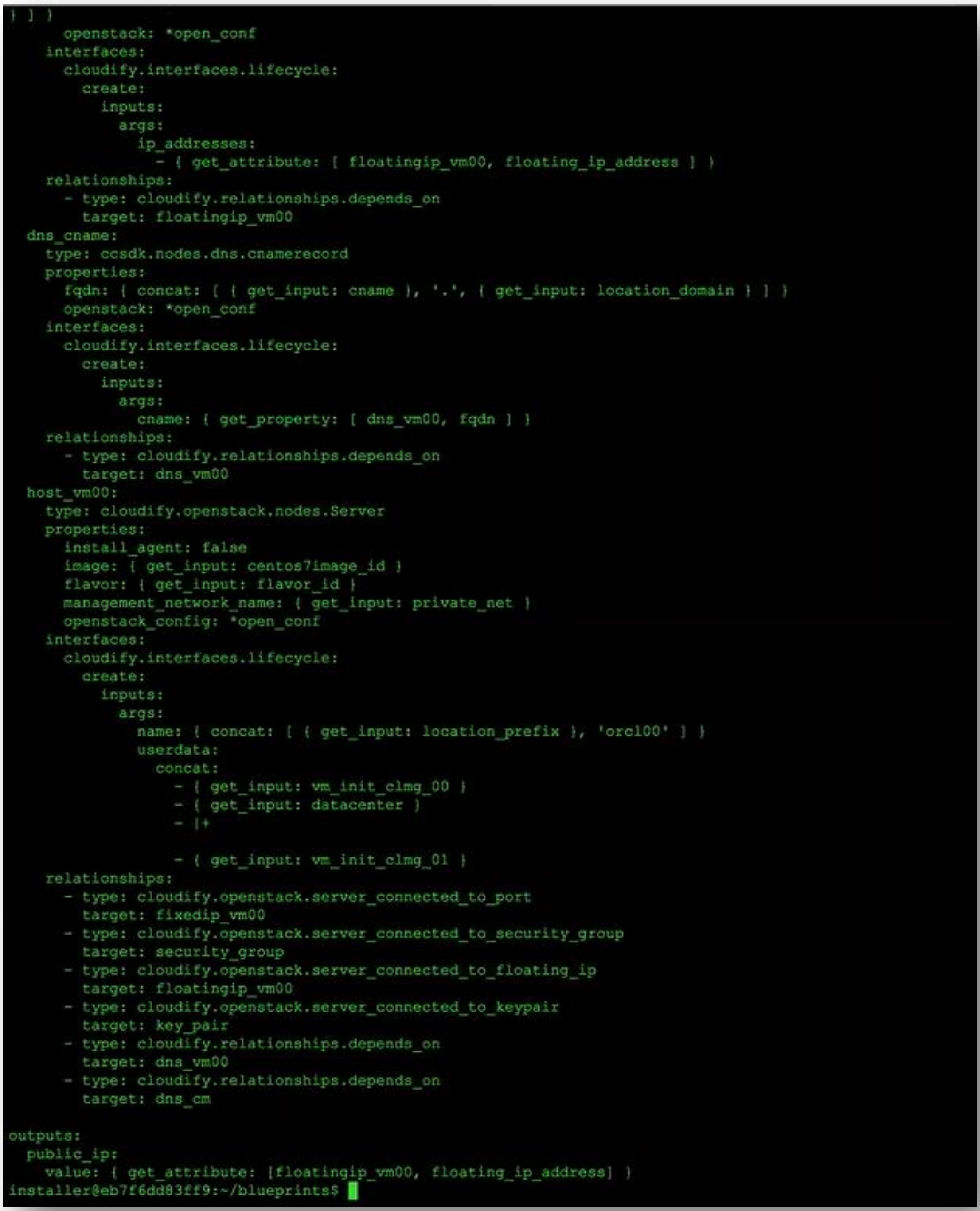
```
# -*- indent-tabs-mode: nil -*- # vi: set expandtab:
#
# ============LICENSE_START==========================================
# org.onap.dcae
# ===================================================================
# Copyright © 2017 AT&T Intellectual Property. All rights reserved.
# ===================================================================
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#       http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied
# See the License for the specific language governing permissions and
# limitations under the License.
# ============LICENSE_END============================================
#
# ECOMP and OpenECOMP are trademarks
# and service marks of AT&T Intellectual Property.
#

tosca_definitions_version: cloudify_dsl_1_3

imports:
  - http://www.getcloudify.org/spec/cloudify/3.4/types.yaml
  - http://www.getcloudify.org/spec/openstack-plugin/1.4/plugin.yaml
  - http://www.getcloudify.org/spec/fabric-plugin/1.4.1/plugin.yaml
  - types/dns_types.yaml
  - types/sshkey_types.yaml

inputs:
  centos7image_id:
    type: string
  ubuntu1604image_id:
    type: string
  flavor_id:
    type: string
  security_group:
    type: string
  public_net:
    type: string
  private_net:
    type: string
  openstack: {}
  keypair:
    type: string
  location_prefix:
    type: string
  location_domain:
    type: string
  key_filename:
    type: string
  codesource_url:
    type: string
  codesource_version:
    type: string
  cname:
    type: string
    default: dcae-orcl
  datacenter:
    type: string
--More--(24%)
```



```
  vm_init_clmg_00:
    type: string
    default: |-
      #!/bin/sh
      set -x
      DATACENTER=
  vm_init_clmg_01:
    type: string
    default: |
      CONSULVER=0.8.3
      CONSULNAME=consul_${CONSULVER}_linux_amd64
      MYIP=`curl -Ss http://169.254.169.254/2009-04-04/meta-data/local-ipv4`
      MYNAME=`hostname`
      if [ ! -z "$(echo $MYNAME |grep '.')" ]; then MYNAME="$(echo $MYNAME | cut -f1 -d '.')"; fi
      echo >>/etc/hosts
      echo $MYIP $MYNAME >>/etc/hosts
      mkdir -p /opt/consul/config /opt/consul/data /opt/consul/bin
      yum install -y unzip
      # Download Consul
      curl -Ss   https://releases.hashicorp.com/consul/${CONSULVER}/${CONSULNAME}.zip > ${CONSULNAME}.zip
      unzip -d /opt/consul/bin  ${CONSULNAME}.zip
      rm ${CONSULNAME}.zip
      chmod +x  /opt/consul/bin/consul
      cat <<EOF > /opt/consul/config/consul.json
      {
        "bind_addr" : "0.0.0.0",
        "client_addr" : "0.0.0.0",
        "data_dir" : "/opt/consul/data",
        "datacenter": "$DATACENTER",
        "rejoin_after_leave": true,
        "http_api_response_headers": {
           "Access-Control-Allow-Origin" : "*"
        },
        "server": false,
        "ui": false,
        "enable_syslog": true,
        "log_level": "info"
      }
      EOF
      cat <<EOF > /lib/systemd/system/consul.service
      [Unit]
      Description=Consul
      Requires=network-online.target
      After=network.target
      [Service]
      Type=simple
      ExecStart=/opt/consul/bin/consul agent -config-dir=/opt/consul/config
      ExecReload=/bin/kill -HUP \$MAINPID
      [Install]
      WantedBy=multi-user.target
      EOF
      systemctl enable consul
      systemctl start consul
      yum install -y python-psycopg2

node_templates:
  key_pair:
    type: cloudify.openstack.nodes.KeyPair
    properties:
      private_key_path: { get_input: key_filename }
      use_external_resource: True
      resource_id: { get_input: keypair }
      openstack_config: &open_conf
        get_input: openstack
--More--(50%)
```

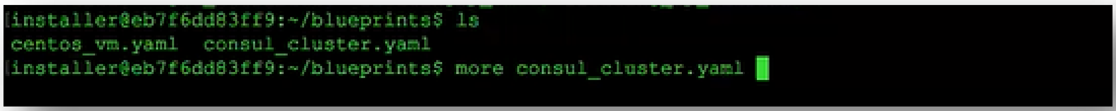


```
  private_net:
    type: cloudify.openstack.nodes.Network
    properties:
      use_external_resource: True
      resource_id: { get_input: private_net }
      openstack_config: *open_conf
  security_group:
    type: cloudify.openstack.nodes.SecurityGroup
    properties:
      use_external_resource: True
      resource_id: { get_input: security_group }
      openstack_config: *open_conf
  fixedip_vm00:
    type: cloudify.openstack.nodes.Port
    properties:
      port:
        extra_dhcp_opts:
          - opt_name: 'domain-name'
            opt_value: { get_input: location_domain }
      openstack_config: *open_conf
    relationships:
      - type: cloudify.relationships.contained_in
        target: private_net
  floatingip_vm00:
    type: cloudify.openstack.nodes.FloatingIP
    properties:
      openstack_config: *open_conf
    interfaces:
      cloudify.interfaces.lifecycle:
        create:
          inputs:
            args:
              floating_network_name: { get_input: public_net }
  dns_vm00:
    type: ccsdk.nodes.dns.arecord
    properties:
      fqdn: { concat: [ { get_input: location_prefix }, 'orcl00.', { get_input: location_domain } ] }
      openstack: *open_conf
    interfaces:
      cloudify.interfaces.lifecycle:
        create:
          inputs:
            args:
              ip_addresses:
                - { get_attribute: [ floatingip_vm00, floating_ip_address ] }
    relationships:
      - type: cloudify.relationships.depends_on
        target: floatingip_vm00
  dns_cm:
    type: ccsdk.nodes.dns.arecord
    properties:
      fqdn: { concat: [ 'cloudify-manager-', { get_input: datacenter}, '.', { get_input: location_domain
} ] }
      openstack: *open_conf
    interfaces:
      cloudify.interfaces.lifecycle:
        create:
          inputs:
            args:
              ip_addresses:
                - { get_attribute: [ floatingip_vm00, floating_ip_address ] }
    relationships:
      - type: cloudify.relationships.depends_on
        target: floatingip_vm00
--More--(77%)
```



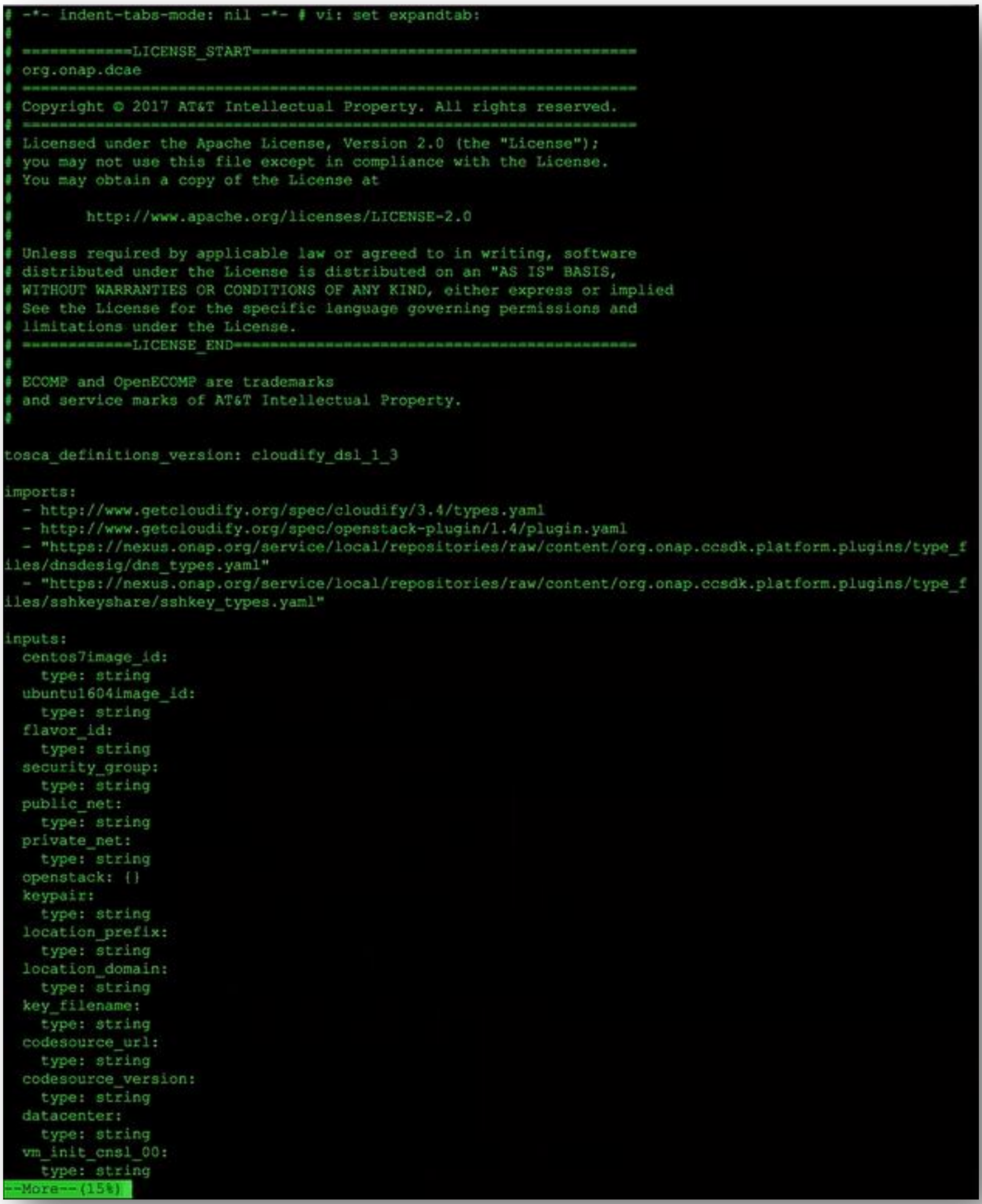
```
} ] }
      openstack: *open_conf
    interfaces:
      cloudify.interfaces.lifecycle:
        create:
          inputs:
            args:
              ip_addresses:
                - { get_attribute: [ floatingip_vm00, floating_ip_address ] }
    relationships:
      - type: cloudify.relationships.depends_on
        target: floatingip_vm00
  dns_cname:
    type: ccsdk.nodes.dns.cnamerecord
    properties:
      fqdn: { concat: [ { get_input: cname }, '.', { get_input: location_domain } ] }
      openstack: *open_conf
    interfaces:
      cloudify.interfaces.lifecycle:
        create:
          inputs:
            args:
              cname: { get_property: [ dns_vm00, fqdn ] }
    relationships:
      - type: cloudify.relationships.depends_on
        target: dns_vm00
  host_vm00:
    type: cloudify.openstack.nodes.Server
    properties:
      install_agent: false
      image: { get_input: centos7image_id }
      flavor: { get_input: flavor_id }
      management_network_name: { get_input: private_net }
      openstack_config: *open_conf
    interfaces:
      cloudify.interfaces.lifecycle:
        create:
          inputs:
            args:
              name: { concat: [ { get_input: location_prefix }, 'orcl00' ] }
              userdata:
                concat:
                  - { get_input: vm_init_clmg_00 }
                  - { get_input: datacenter }
                  - |+

                  - { get_input: vm_init_clmg_01 }
    relationships:
      - type: cloudify.openstack.server_connected_to_port
        target: fixedip_vm00
      - type: cloudify.openstack.server_connected_to_security_group
        target: security_group
      - type: cloudify.openstack.server_connected_to_floating_ip
        target: floatingip_vm00
      - type: cloudify.openstack.server_connected_to_keypair
        target: key_pair
      - type: cloudify.relationships.depends_on
        target: dns_vm00
      - type: cloudify.relationships.depends_on
        target: dns_cm

outputs:
  public_ip:
    value: { get_attribute: [floatingip_vm00, floating_ip_address] }
installer@eb7f6dd83ff9:~/blueprints$
```

and then you have the consul_cluster.yaml.

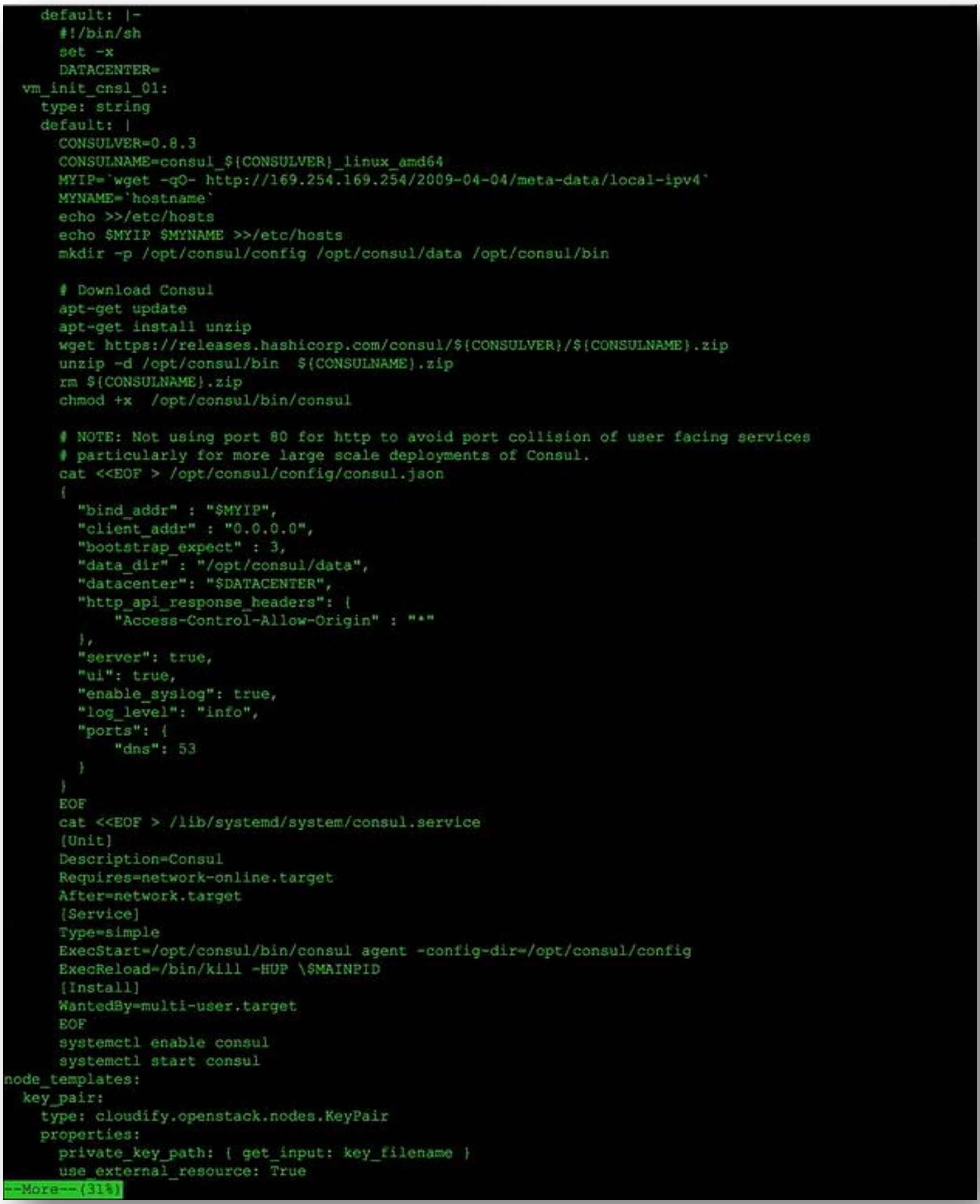


```
installer@eb7f6dd83ff9:~/blueprints$ ls
centos_vm.yaml  consul_cluster.yaml
installer@eb7f6dd83ff9:~/blueprints$ more consul_cluster.yaml
```

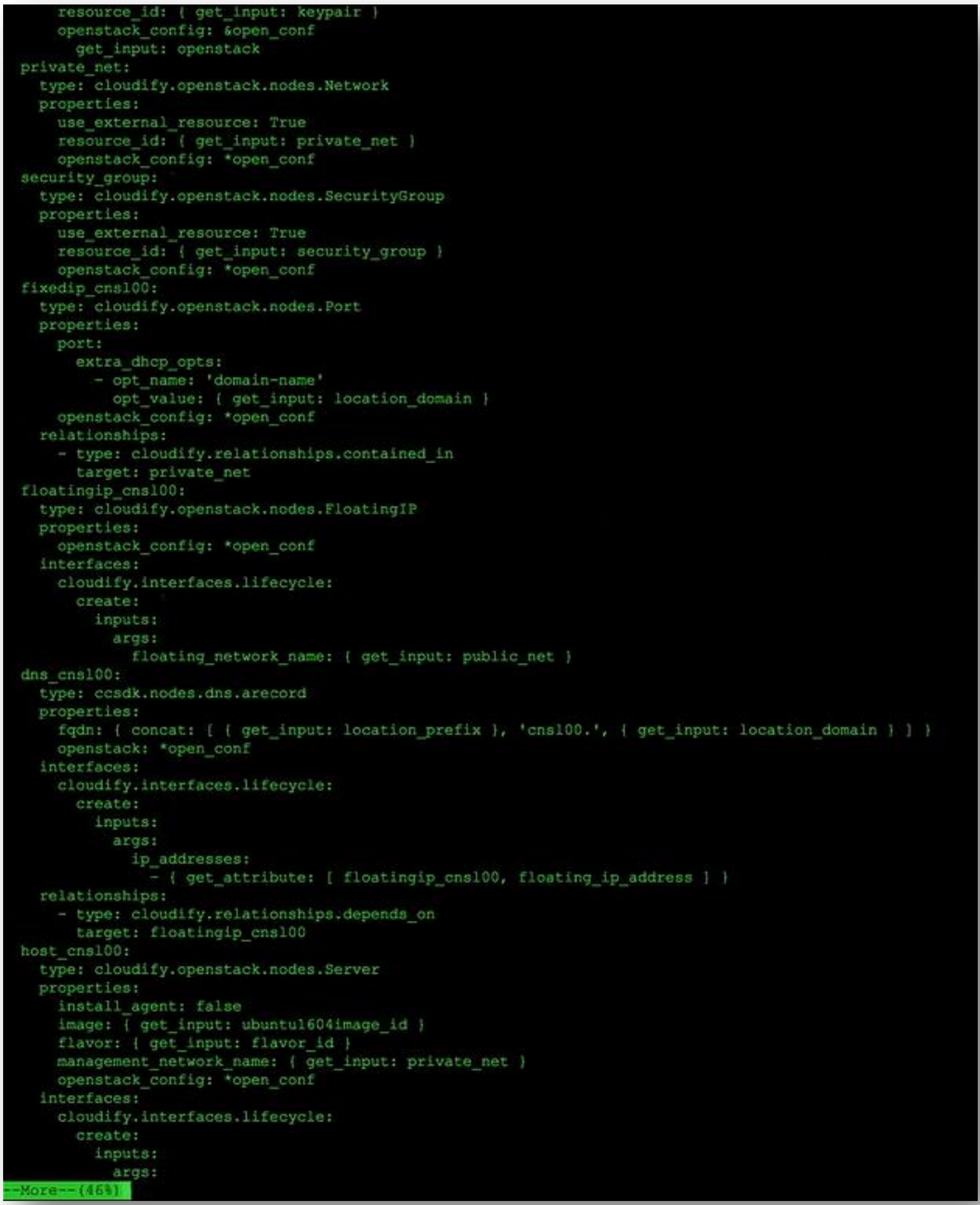
Some people may say in dev environment I really don’t want to have 3 VM cluster then you want to locate this blueprint and come up with probably just single VM blueprint alternative.



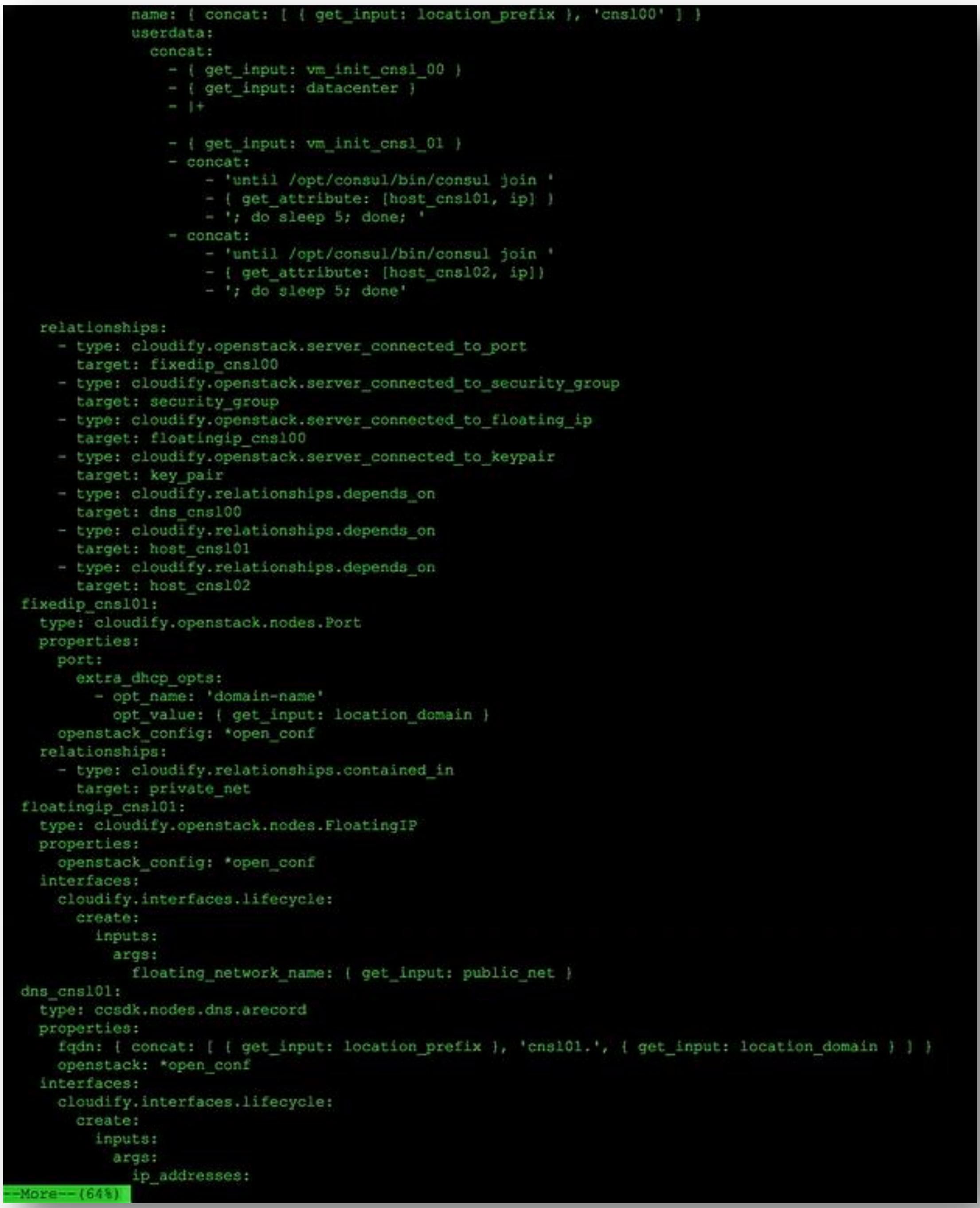
```
# -*- indent-tabs-mode: nil -*- # vi: set expandtab:
#
# ============LICENSE_START==========================================
# org.onap.dcae
# ===================================================================
# Copyright © 2017 AT&T Intellectual Property. All rights reserved.
# ===================================================================
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#       http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied
# See the License for the specific language governing permissions and
# limitations under the License.
# ============LICENSE_END============================================
#
# ECOMP and OpenECOMP are trademarks
# and service marks of AT&T Intellectual Property.
#

tosca_definitions_version: cloudify_dsl_1_3

imports:
  - http://www.getcloudify.org/spec/cloudify/3.4/types.yaml
  - http://www.getcloudify.org/spec/openstack-plugin/1.4/plugin.yaml
  - "https://nexus.onap.org/service/local/repositories/raw/content/org.onap.ccsdk.platform.plugins/type_f
iles/dnsdesig/dns_types.yaml"
  - "https://nexus.onap.org/service/local/repositories/raw/content/org.onap.ccsdk.platform.plugins/type_f
iles/sshkeyshare/sshkey_types.yaml"

inputs:
  centos7image_id:
    type: string
  ubuntu1604image_id:
    type: string
  flavor_id:
    type: string
  security_group:
    type: string
  public_net:
    type: string
  private_net:
    type: string
  openstack: {}
  keypair:
    type: string
  location_prefix:
    type: string
  location_domain:
    type: string
  key_filename:
    type: string
  codesource_url:
    type: string
  codesource_version:
    type: string
  datacenter:
    type: string
  vm_init_cnsl_00:
    type: string
--More--(15%)
```



```
    default: |-
      #!/bin/sh
      set -x
      DATACENTER=
  vm_init_cnsl_01:
    type: string
    default: |
      CONSULVER=0.8.3
      CONSULNAME=consul_${CONSULVER}_linux_amd64
      MYIP=`wget -qO- http://169.254.169.254/2009-04-04/meta-data/local-ipv4`
      MYNAME=`hostname`
      echo >>/etc/hosts
      echo $MYIP $MYNAME >>/etc/hosts
      mkdir -p /opt/consul/config /opt/consul/data /opt/consul/bin

      # Download Consul
      apt-get update
      apt-get install unzip
      wget https://releases.hashicorp.com/consul/${CONSULVER}/${CONSULNAME}.zip
      unzip -d /opt/consul/bin  ${CONSULNAME}.zip
      rm ${CONSULNAME}.zip
      chmod +x  /opt/consul/bin/consul

      # NOTE: Not using port 80 for http to avoid port collision of user facing services
      # particularly for more large scale deployments of Consul.
      cat <<EOF > /opt/consul/config/consul.json
      {
        "bind_addr" : "$MYIP",
        "client_addr" : "0.0.0.0",
        "bootstrap_expect" : 3,
        "data_dir" : "/opt/consul/data",
        "datacenter": "$DATACENTER",
        "http_api_response_headers": {
            "Access-Control-Allow-Origin" : "*"
        },
        "server": true,
        "ui": true,
        "enable_syslog": true,
        "log_level": "info",
        "ports": {
            "dns": 53
        }
      }
      EOF
      cat <<EOF > /lib/systemd/system/consul.service
      [Unit]
      Description=Consul
      Requires=network-online.target
      After=network.target
      [Service]
      Type=simple
      ExecStart=/opt/consul/bin/consul agent -config-dir=/opt/consul/config
      ExecReload=/bin/kill -HUP \$MAINPID
      [Install]
      WantedBy=multi-user.target
      EOF
      systemctl enable consul
      systemctl start consul
node_templates:
  key_pair:
    type: cloudify.openstack.nodes.KeyPair
    properties:
      private_key_path: { get_input: key_filename }
      use_external_resource: True
--More--(31%)
```

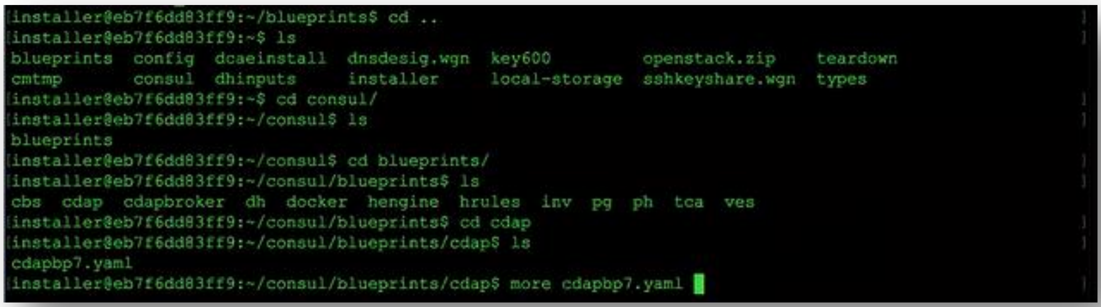


```
      resource_id: { get_input: keypair }
      openstack_config: &open_conf
        get_input: openstack
  private_net:
    type: cloudify.openstack.nodes.Network
    properties:
      use_external_resource: True
      resource_id: { get_input: private_net }
      openstack_config: *open_conf
  security_group:
    type: cloudify.openstack.nodes.SecurityGroup
    properties:
      use_external_resource: True
      resource_id: { get_input: security_group }
      openstack_config: *open_conf
  fixedip_cnsl00:
    type: cloudify.openstack.nodes.Port
    properties:
      port:
        extra_dhcp_opts:
          - opt_name: 'domain-name'
            opt_value: { get_input: location_domain }
      openstack_config: *open_conf
    relationships:
      - type: cloudify.relationships.contained_in
        target: private_net
  floatingip_cnsl00:
    type: cloudify.openstack.nodes.FloatingIP
    properties:
      openstack_config: *open_conf
    interfaces:
      cloudify.interfaces.lifecycle:
        create:
          inputs:
            args:
              floating_network_name: { get_input: public_net }
  dns_cnsl00:
    type: ccsdk.nodes.dns.arecord
    properties:
      fqdn: { concat: [ { get_input: location_prefix }, 'cnsl00.', { get_input: location_domain } ] }
      openstack: *open_conf
    interfaces:
      cloudify.interfaces.lifecycle:
        create:
          inputs:
            args:
              ip_addresses:
                - { get_attribute: [ floatingip_cnsl00, floating_ip_address ] }
    relationships:
      - type: cloudify.relationships.depends_on
        target: floatingip_cnsl00
  host_cnsl00:
    type: cloudify.openstack.nodes.Server
    properties:
      install_agent: false
      image: { get_input: ubuntu1604image_id }
      flavor: { get_input: flavor_id }
      management_network_name: { get_input: private_net }
      openstack_config: *open_conf
    interfaces:
      cloudify.interfaces.lifecycle:
        create:
          inputs:
            args:
--More--(46%)
```

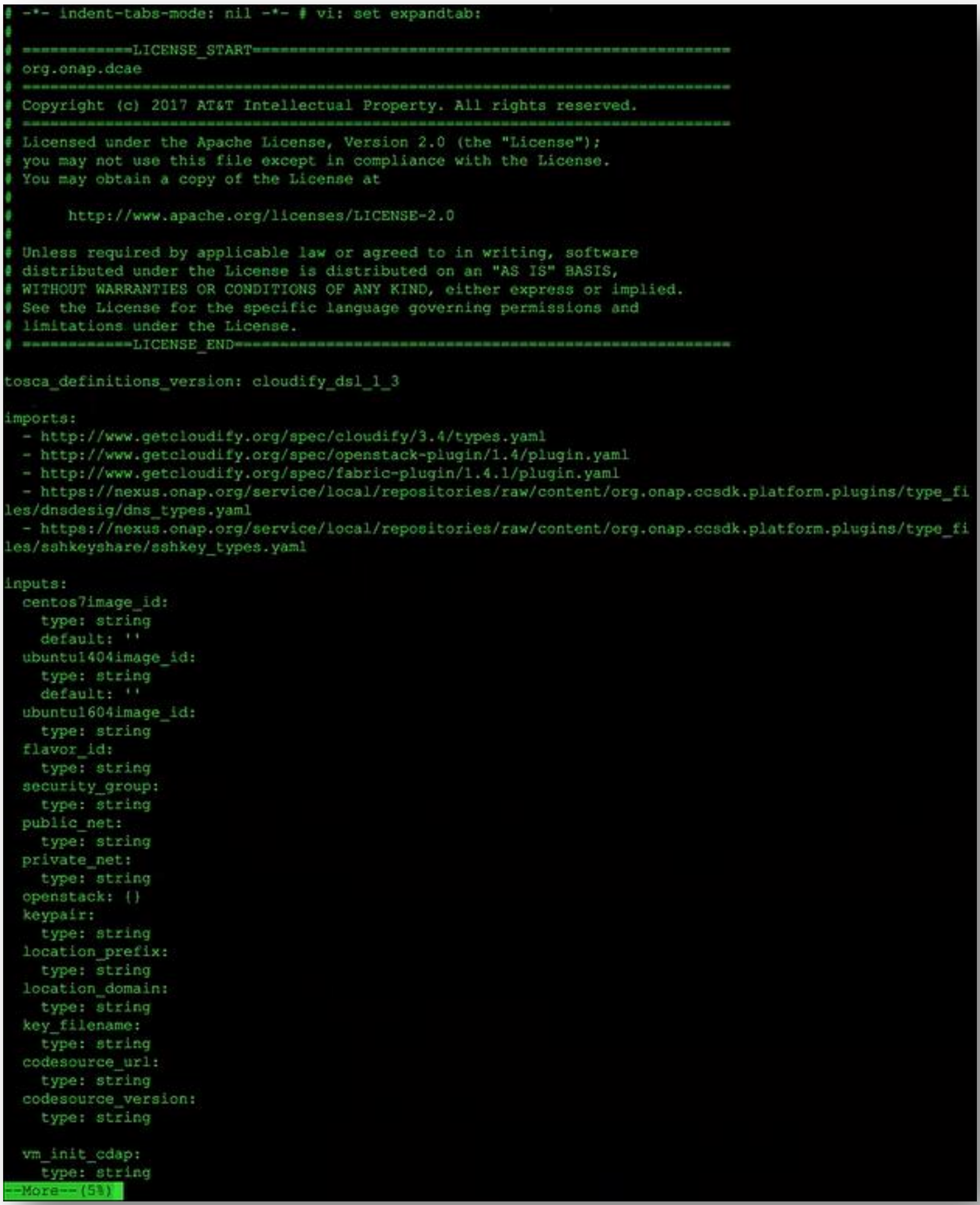


```
              name: { concat: [ { get_input: location_prefix }, 'cnsl00' ] }
              userdata:
                concat:
                  - { get_input: vm_init_cnsl_00 }
                  - { get_input: datacenter }
                  - |+

                  - { get_input: vm_init_cnsl_01 }
                  - concat:
                      - 'until /opt/consul/bin/consul join '
                      - { get_attribute: [host_cnsl01, ip] }
                      - '; do sleep 5; done; '
                  - concat:
                      - 'until /opt/consul/bin/consul join '
                      - { get_attribute: [host_cnsl02, ip]}
                      - '; do sleep 5; done'

    relationships:
      - type: cloudify.openstack.server_connected_to_port
        target: fixedip_cnsl00
      - type: cloudify.openstack.server_connected_to_security_group
        target: security_group
      - type: cloudify.openstack.server_connected_to_floating_ip
        target: floatingip_cnsl00
      - type: cloudify.openstack.server_connected_to_keypair
        target: key_pair
      - type: cloudify.relationships.depends_on
        target: dns_cnsl00
      - type: cloudify.relationships.depends_on
        target: host_cnsl01
      - type: cloudify.relationships.depends_on
        target: host_cnsl02
  fixedip_cnsl01:
    type: cloudify.openstack.nodes.Port
    properties:
      port:
        extra_dhcp_opts:
          - opt_name: 'domain-name'
            opt_value: { get_input: location_domain }
      openstack_config: *open_conf
    relationships:
      - type: cloudify.relationships.contained_in
        target: private_net
  floatingip_cnsl01:
    type: cloudify.openstack.nodes.FloatingIP
    properties:
      openstack_config: *open_conf
    interfaces:
      cloudify.interfaces.lifecycle:
        create:
          inputs:
            args:
              floating_network_name: { get_input: public_net }
  dns_cnsl01:
    type: ccsdk.nodes.dns.arecord
    properties:
      fqdn: { concat: [ { get_input: location_prefix }, 'cnsl01.', { get_input: location_domain } ] }
      openstack: *open_conf
    interfaces:
      cloudify.interfaces.lifecycle:
        create:
          inputs:
            args:
              ip_addresses:
--More--(64%)
```

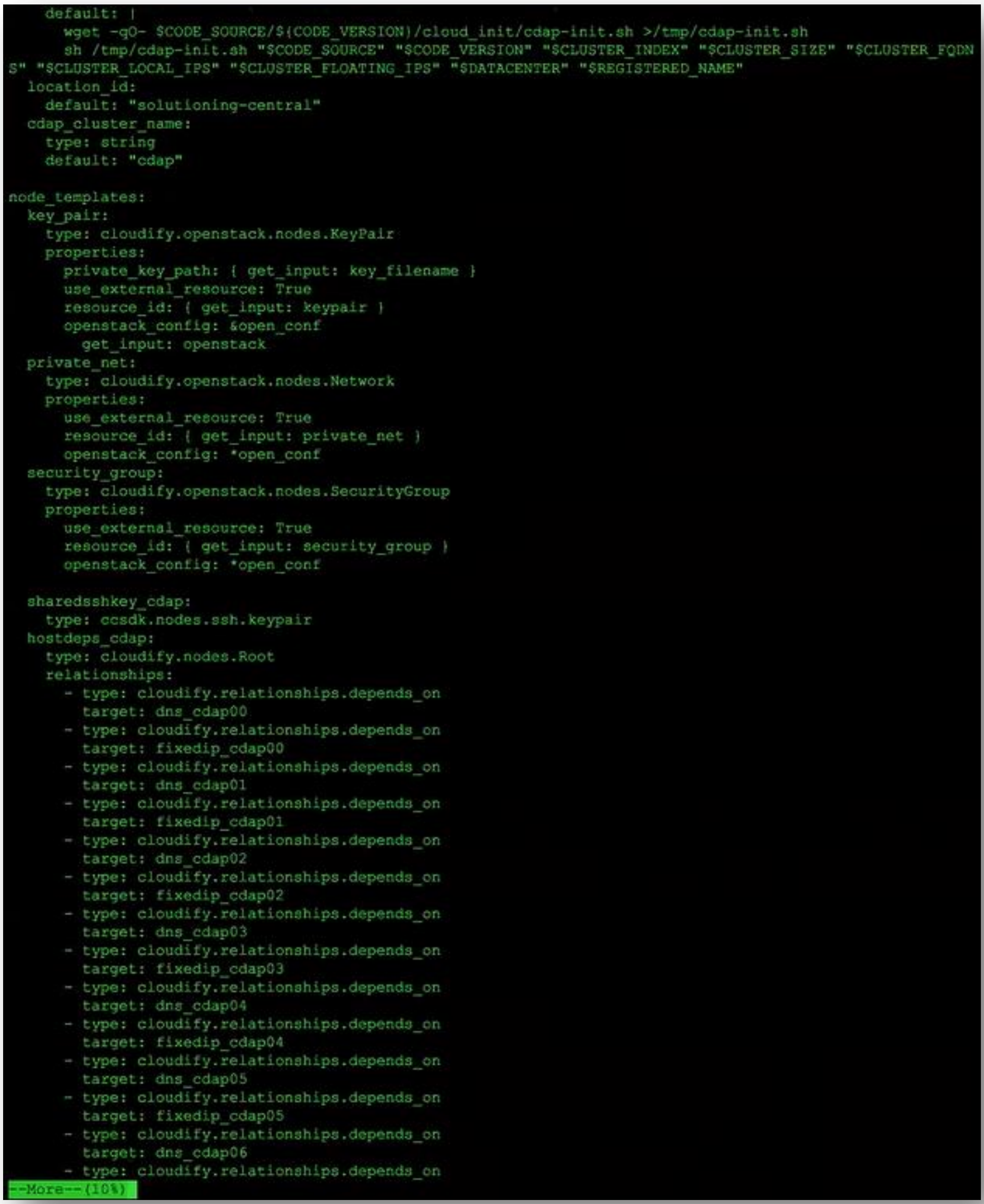
After the core components of DCAE platform are deployed then the rest of the blueprints are available and if you go to consul there is also the blueprints catalogue and here are the blueprints for all the other components so for example CDAP (with cdapbp7.yaml) that is where the 7 VMs CDAP blueprint is.



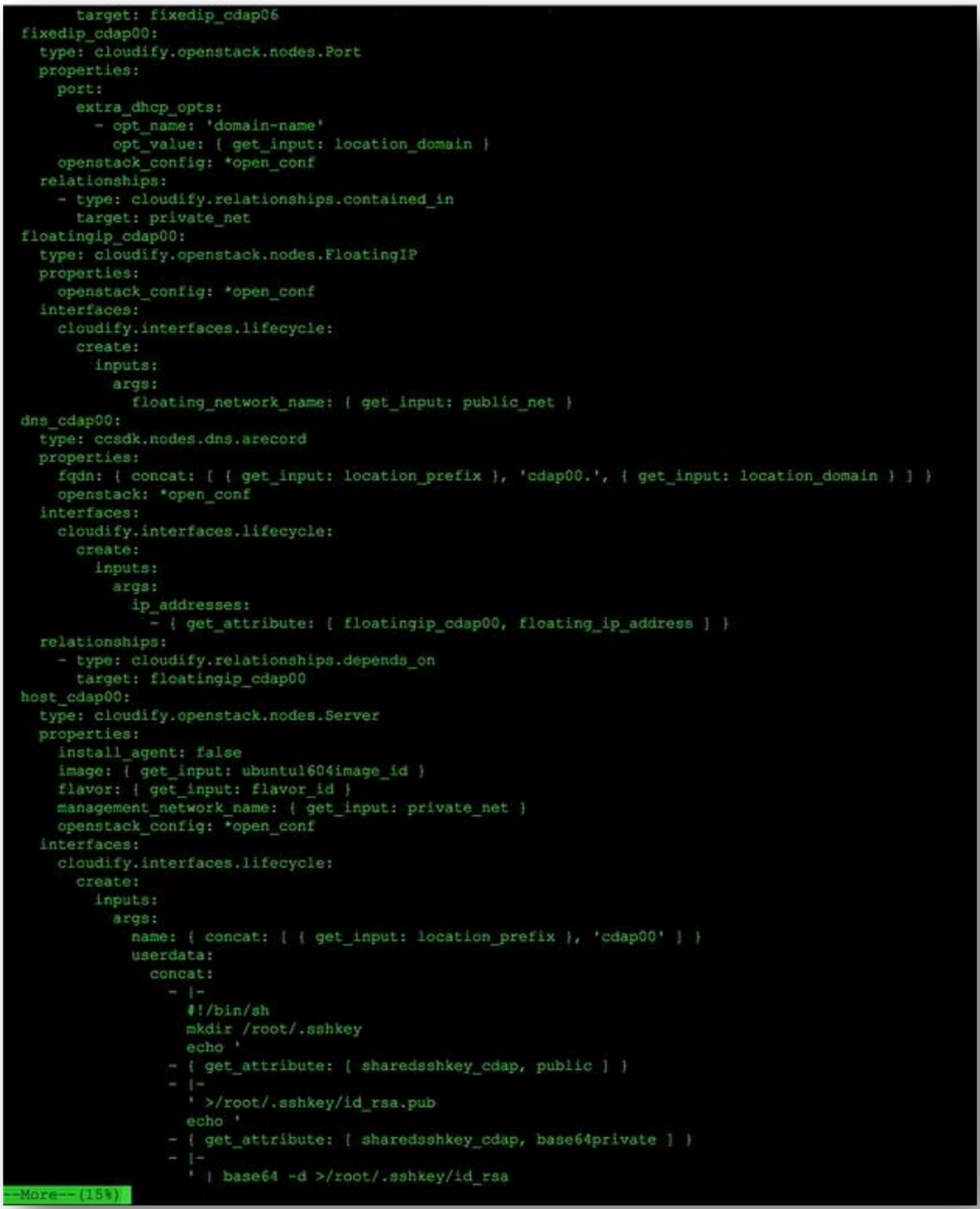
```
installer@eb7f6dd83ff9:~/blueprints$ cd ..
installer@eb7f6dd83ff9:~$ ls
blueprints  config  dcaeinstall  dnsdesig.wgn  key600         openstack.zip    teardown
cmtmp       consul  dhinputs     installer     local-storage  sshkeyshare.wgn  types
installer@eb7f6dd83ff9:~$ cd consul/
installer@eb7f6dd83ff9:~/consul$ ls
blueprints
installer@eb7f6dd83ff9:~/consul$ cd blueprints/
installer@eb7f6dd83ff9:~/consul/blueprints$ ls
cbs  cdap  cdapbroker  dh  docker  hengine  hrules  inv  pg  ph  tca  ves
installer@eb7f6dd83ff9:~/consul/blueprints$ cd cdap
installer@eb7f6dd83ff9:~/consul/blueprints/cdap$ ls
cdapbp7.yaml
installer@eb7f6dd83ff9:~/consul/blueprints/cdap$ more cdapbp7.yaml
```



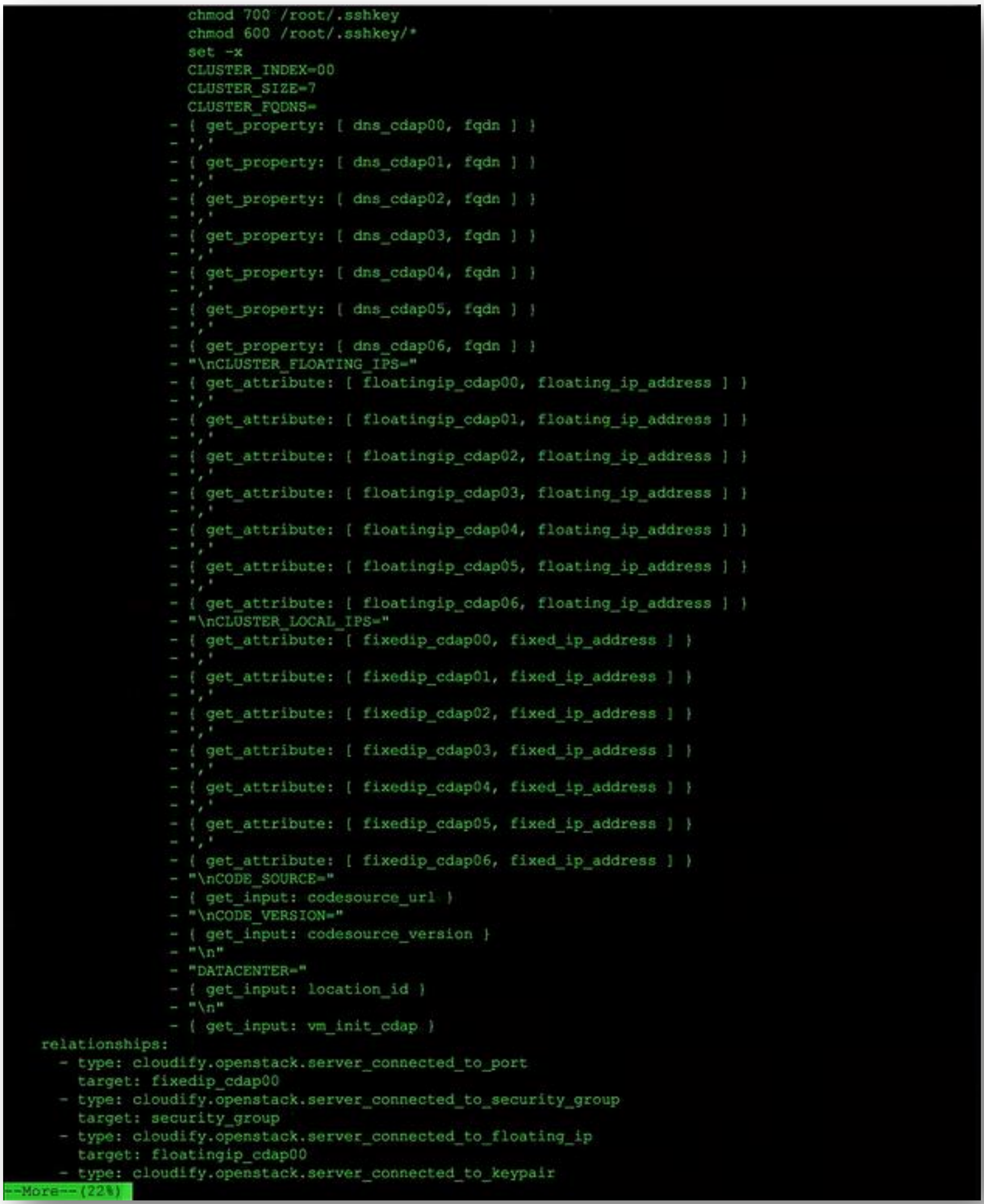
```
# -*- indent-tabs-mode: nil -*- # vi: set expandtab:
#
# ============LICENSE_START==================================================
# org.onap.dcae
# ===========================================================================
# Copyright (c) 2017 AT&T Intellectual Property. All rights reserved.
# ===========================================================================
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#      http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
# ============LICENSE_END====================================================

tosca_definitions_version: cloudify_dsl_1_3

imports:
  - http://www.getcloudify.org/spec/cloudify/3.4/types.yaml
  - http://www.getcloudify.org/spec/openstack-plugin/1.4/plugin.yaml
  - http://www.getcloudify.org/spec/fabric-plugin/1.4.1/plugin.yaml
  - https://nexus.onap.org/service/local/repositories/raw/content/org.onap.ccsdk.platform.plugins/type_fi
les/dnsdesig/dns_types.yaml
  - https://nexus.onap.org/service/local/repositories/raw/content/org.onap.ccsdk.platform.plugins/type_fi
les/sshkeyshare/sshkey_types.yaml

inputs:
  centos7image_id:
    type: string
    default: ''
  ubuntu1404image_id:
    type: string
    default: ''
  ubuntu1604image_id:
    type: string
  flavor_id:
    type: string
  security_group:
    type: string
  public_net:
    type: string
  private_net:
    type: string
  openstack: {}
  keypair:
    type: string
  location_prefix:
    type: string
  location_domain:
    type: string
  key_filename:
    type: string
  codesource_url:
    type: string
  codesource_version:
    type: string

  vm_init_cdap:
    type: string
--More--(5%)
```



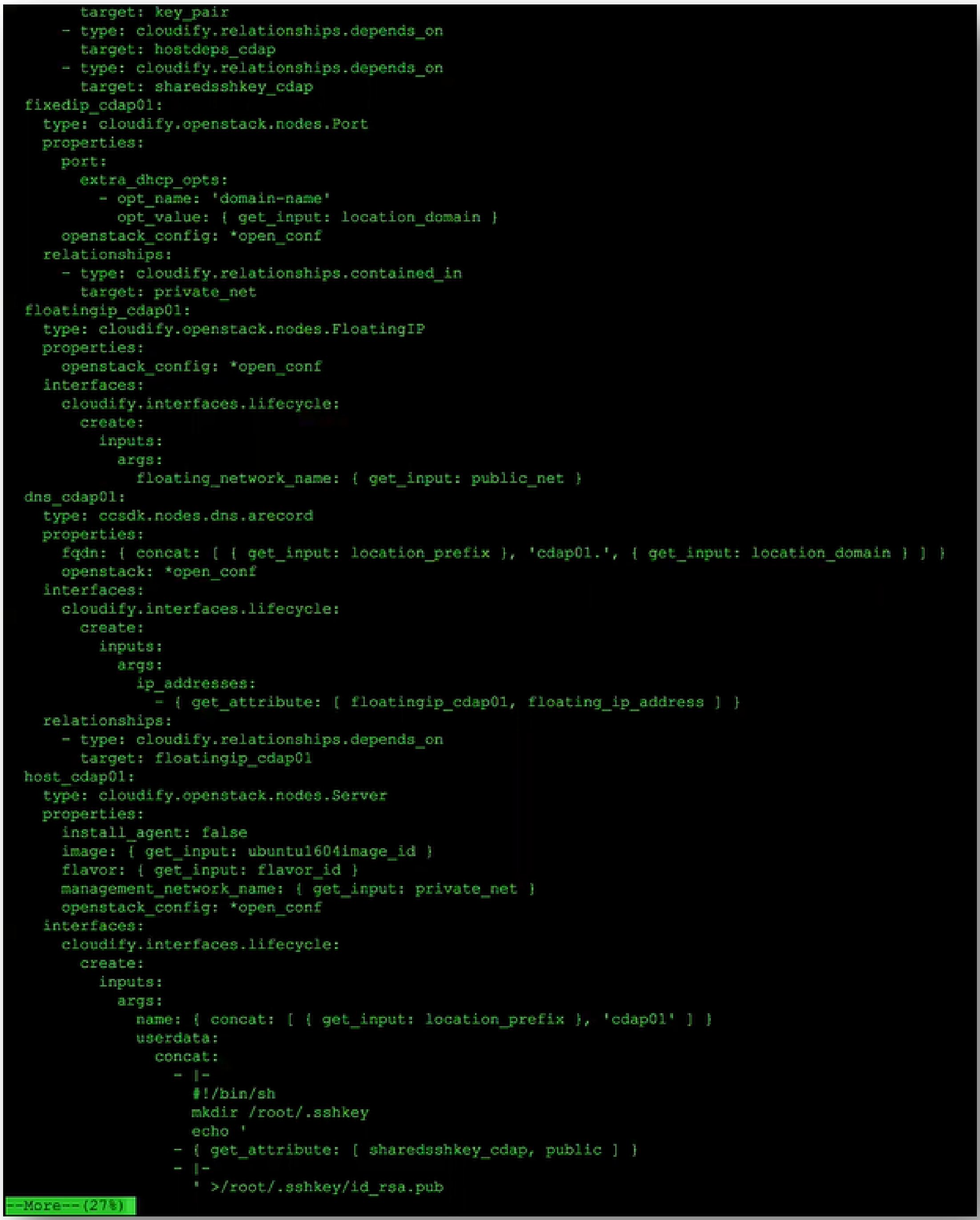
```
    default: |
      wget -qO- $CODE_SOURCE/${CODE_VERSION}/cloud_init/cdap-init.sh >/tmp/cdap-init.sh
      sh /tmp/cdap-init.sh "$CODE_SOURCE" "$CODE_VERSION" "$CLUSTER_INDEX" "$CLUSTER_SIZE" "$CLUSTER_FQDN
S" "$CLUSTER_LOCAL_IPS" "$CLUSTER_FLOATING_IPS" "$DATACENTER" "$REGISTERED_NAME"
  location_id:
    default: "solutioning-central"
  cdap_cluster_name:
    type: string
    default: "cdap"

node_templates:
  key_pair:
    type: cloudify.openstack.nodes.KeyPair
    properties:
      private_key_path: { get_input: key_filename }
      use_external_resource: True
      resource_id: { get_input: keypair }
      openstack_config: &open_conf
        get_input: openstack
  private_net:
    type: cloudify.openstack.nodes.Network
    properties:
      use_external_resource: True
      resource_id: { get_input: private_net }
      openstack_config: *open_conf
  security_group:
    type: cloudify.openstack.nodes.SecurityGroup
    properties:
      use_external_resource: True
      resource_id: { get_input: security_group }
      openstack_config: *open_conf

  sharedsshkey_cdap:
    type: ccsdk.nodes.ssh.keypair
  hostdeps_cdap:
    type: cloudify.nodes.Root
    relationships:
      - type: cloudify.relationships.depends_on
        target: dns_cdap00
      - type: cloudify.relationships.depends_on
        target: fixedip_cdap00
      - type: cloudify.relationships.depends_on
        target: dns_cdap01
      - type: cloudify.relationships.depends_on
        target: fixedip_cdap01
      - type: cloudify.relationships.depends_on
        target: dns_cdap02
      - type: cloudify.relationships.depends_on
        target: fixedip_cdap02
      - type: cloudify.relationships.depends_on
        target: dns_cdap03
      - type: cloudify.relationships.depends_on
        target: fixedip_cdap03
      - type: cloudify.relationships.depends_on
        target: dns_cdap04
      - type: cloudify.relationships.depends_on
        target: fixedip_cdap04
      - type: cloudify.relationships.depends_on
        target: dns_cdap05
      - type: cloudify.relationships.depends_on
        target: fixedip_cdap05
      - type: cloudify.relationships.depends_on
        target: dns_cdap06
      - type: cloudify.relationships.depends_on
--More--(10%)
```



```
      target: fixedip_cdap06
fixedip_cdap00:
  type: cloudify.openstack.nodes.Port
  properties:
    port:
      extra_dhcp_opts:
        - opt_name: 'domain-name'
          opt_value: { get_input: location_domain }
    openstack_config: *open_conf
  relationships:
    - type: cloudify.relationships.contained_in
      target: private_net
floatingip_cdap00:
  type: cloudify.openstack.nodes.FloatingIP
  properties:
    openstack_config: *open_conf
  interfaces:
    cloudify.interfaces.lifecycle:
      create:
        inputs:
          args:
            floating_network_name: { get_input: public_net }
dns_cdap00:
  type: ccsdk.nodes.dns.arecord
  properties:
    fqdn: { concat: [ { get_input: location_prefix }, 'cdap00.', { get_input: location_domain } ] }
    openstack: *open_conf
  interfaces:
    cloudify.interfaces.lifecycle:
      create:
        inputs:
          args:
            ip_addresses:
              - { get_attribute: [ floatingip_cdap00, floating_ip_address ] }
  relationships:
    - type: cloudify.relationships.depends_on
      target: floatingip_cdap00
host_cdap00:
  type: cloudify.openstack.nodes.Server
  properties:
    install_agent: false
    image: { get_input: ubuntu1604image_id }
    flavor: { get_input: flavor_id }
    management_network_name: { get_input: private_net }
    openstack_config: *open_conf
  interfaces:
    cloudify.interfaces.lifecycle:
      create:
        inputs:
          args:
            name: { concat: [ { get_input: location_prefix }, 'cdap00' ] }
            userdata:
              concat:
                - |-
                  #!/bin/sh
                  mkdir /root/.sshkey
                  echo '
                - { get_attribute: [ sharedsshkey_cdap, public ] }
                - |-
                  ' >/root/.sshkey/id_rsa.pub
                  echo '
                - { get_attribute: [ sharedsshkey_cdap, base64private ] }
                - |-
                  ' | base64 -d >/root/.sshkey/id_rsa
--More--(15%)
```



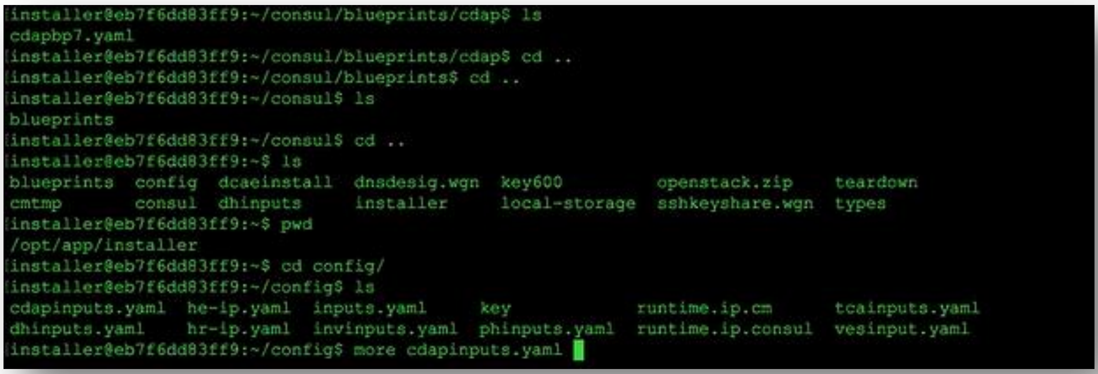
```
                    chmod 700 /root/.sshkey
                    chmod 600 /root/.sshkey/*
                    set -x
                    CLUSTER_INDEX=00
                    CLUSTER_SIZE=7
                    CLUSTER_FQDNS=
                  - { get_property: [ dns_cdap00, fqdn ] }
                  - ','
                  - { get_property: [ dns_cdap01, fqdn ] }
                  - ','
                  - { get_property: [ dns_cdap02, fqdn ] }
                  - ','
                  - { get_property: [ dns_cdap03, fqdn ] }
                  - ','
                  - { get_property: [ dns_cdap04, fqdn ] }
                  - ','
                  - { get_property: [ dns_cdap05, fqdn ] }
                  - ','
                  - { get_property: [ dns_cdap06, fqdn ] }
                  - "\nCLUSTER_FLOATING_IPS="
                  - { get_attribute: [ floatingip_cdap00, floating_ip_address ] }
                  - ','
                  - { get_attribute: [ floatingip_cdap01, floating_ip_address ] }
                  - ','
                  - { get_attribute: [ floatingip_cdap02, floating_ip_address ] }
                  - ','
                  - { get_attribute: [ floatingip_cdap03, floating_ip_address ] }
                  - ','
                  - { get_attribute: [ floatingip_cdap04, floating_ip_address ] }
                  - ','
                  - { get_attribute: [ floatingip_cdap05, floating_ip_address ] }
                  - ','
                  - { get_attribute: [ floatingip_cdap06, floating_ip_address ] }
                  - "\nCLUSTER_LOCAL_IPS="
                  - { get_attribute: [ fixedip_cdap00, fixed_ip_address ] }
                  - ','
                  - { get_attribute: [ fixedip_cdap01, fixed_ip_address ] }
                  - ','
                  - { get_attribute: [ fixedip_cdap02, fixed_ip_address ] }
                  - ','
                  - { get_attribute: [ fixedip_cdap03, fixed_ip_address ] }
                  - ','
                  - { get_attribute: [ fixedip_cdap04, fixed_ip_address ] }
                  - ','
                  - { get_attribute: [ fixedip_cdap05, fixed_ip_address ] }
                  - ','
                  - { get_attribute: [ fixedip_cdap06, fixed_ip_address ] }
                  - "\nCODE_SOURCE="
                  - { get_input: codesource_url }
                  - "\nCODE_VERSION="
                  - { get_input: codesource_version }
                  - "\n"
                  - "DATACENTER="
                  - { get_input: location_id }
                  - "\n"
                  - { get_input: vm_init_cdap }
    relationships:
      - type: cloudify.openstack.server_connected_to_port
        target: fixedip_cdap00
      - type: cloudify.openstack.server_connected_to_security_group
        target: security_group
      - type: cloudify.openstack.server_connected_to_floating_ip
        target: floatingip_cdap00
      - type: cloudify.openstack.server_connected_to_keypair
--More--(22%)
```



```
      target: key_pair
    - type: cloudify.relationships.depends_on
      target: hostdeps_cdap
    - type: cloudify.relationships.depends_on
      target: sharedsshkey_cdap
fixedip_cdap01:
  type: cloudify.openstack.nodes.Port
  properties:
    port:
      extra_dhcp_opts:
        - opt_name: 'domain-name'
          opt_value: { get_input: location_domain }
    openstack_config: *open_conf
  relationships:
    - type: cloudify.relationships.contained_in
      target: private_net
floatingip_cdap01:
  type: cloudify.openstack.nodes.FloatingIP
  properties:
    openstack_config: *open_conf
  interfaces:
    cloudify.interfaces.lifecycle:
      create:
        inputs:
          args:
            floating_network_name: { get_input: public_net }
dns_cdap01:
  type: ccsdk.nodes.dns.arecord
  properties:
    fqdn: { concat: [ { get_input: location_prefix }, 'cdap01.', { get_input: location_domain } ] }
    openstack: *open_conf
  interfaces:
    cloudify.interfaces.lifecycle:
      create:
        inputs:
          args:
            ip_addresses:
              - { get_attribute: [ floatingip_cdap01, floating_ip_address ] }
  relationships:
    - type: cloudify.relationships.depends_on
      target: floatingip_cdap01
host_cdap01:
  type: cloudify.openstack.nodes.Server
  properties:
    install_agent: false
    image: { get_input: ubuntu1604image_id }
    flavor: { get_input: flavor_id }
    management_network_name: { get_input: private_net }
    openstack_config: *open_conf
  interfaces:
    cloudify.interfaces.lifecycle:
      create:
        inputs:
          args:
            name: { concat: [ { get_input: location_prefix }, 'cdap01' ] }
            userdata:
              concat:
                - |-
                  #!/bin/sh
                  mkdir /root/.sshkey
                  echo '
                - { get_attribute: [ sharedsshkey_cdap, public ] }
                - |-
                  ' >/root/.sshkey/id_rsa.pub
--More--(27%)
```

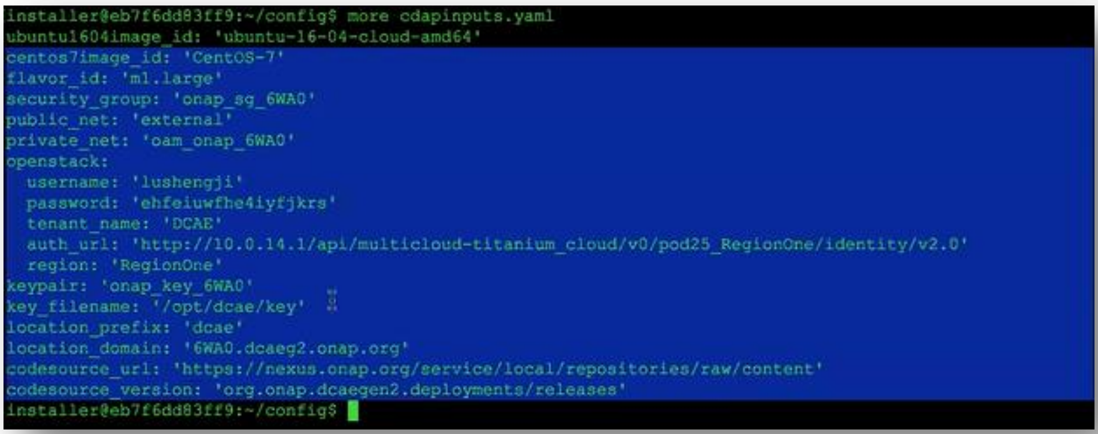
You can see how it is configured that is the beauty of open source – you can really take a look what is under knees.

Of course to modify and get a blueprint for cloudify fully working it may not be an easy task so that is probably later on into the learning process people would be able to “cup the lay dune?” but for now there is another place where you can play with configurations that is we have the input files in the config catalogue (inputs.yaml)



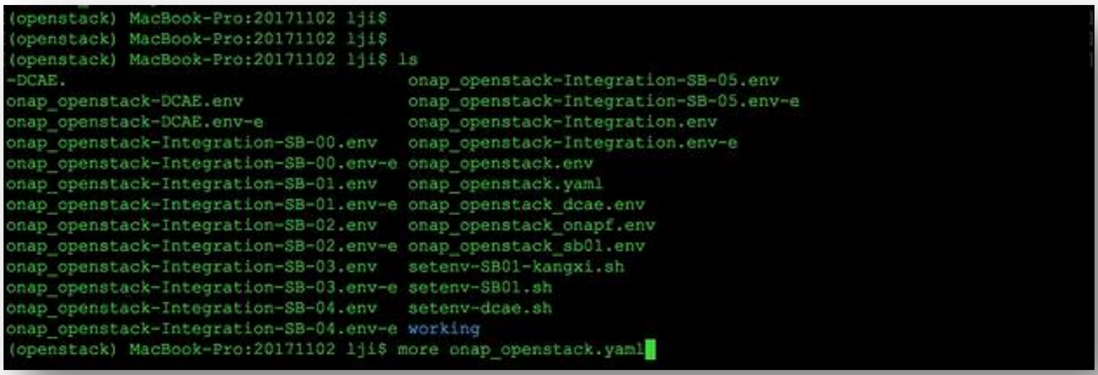
```
installer@eb7f6dd83ff9:~/consul/blueprints/cdap$ ls
cdapbp7.yaml
installer@eb7f6dd83ff9:~/consul/blueprints/cdap$ cd ..
installer@eb7f6dd83ff9:~/consul/blueprints$ cd ..
installer@eb7f6dd83ff9:~/consul$ ls
blueprints
installer@eb7f6dd83ff9:~/consul$ cd ..
installer@eb7f6dd83ff9:~$ ls
blueprints  config  dcaeinstall  dnsdesig.wgn  key600         openstack.zip    teardown
cmtmp       consul  dhinputs     installer     local-storage  sshkeyshare.wgn  types
installer@eb7f6dd83ff9:~$ pwd
/opt/app/installer
installer@eb7f6dd83ff9:~$ cd config/
installer@eb7f6dd83ff9:~/config$ ls
cdapinputs.yaml  he-ip.yaml  inputs.yaml     key             runtime.ip.cm      tcainputs.yaml
dhinputs.yaml    hr-ip.yaml  invinputs.yaml  phinputs.yaml   runtime.ip.consul  vesinput.yaml
installer@eb7f6dd83ff9:~/config$ more cdapinputs.yaml
```

you can play with the some of the input files, so like CDAP (cdapinputs.yaml) that is the input file that is red by the CDAP blueprint



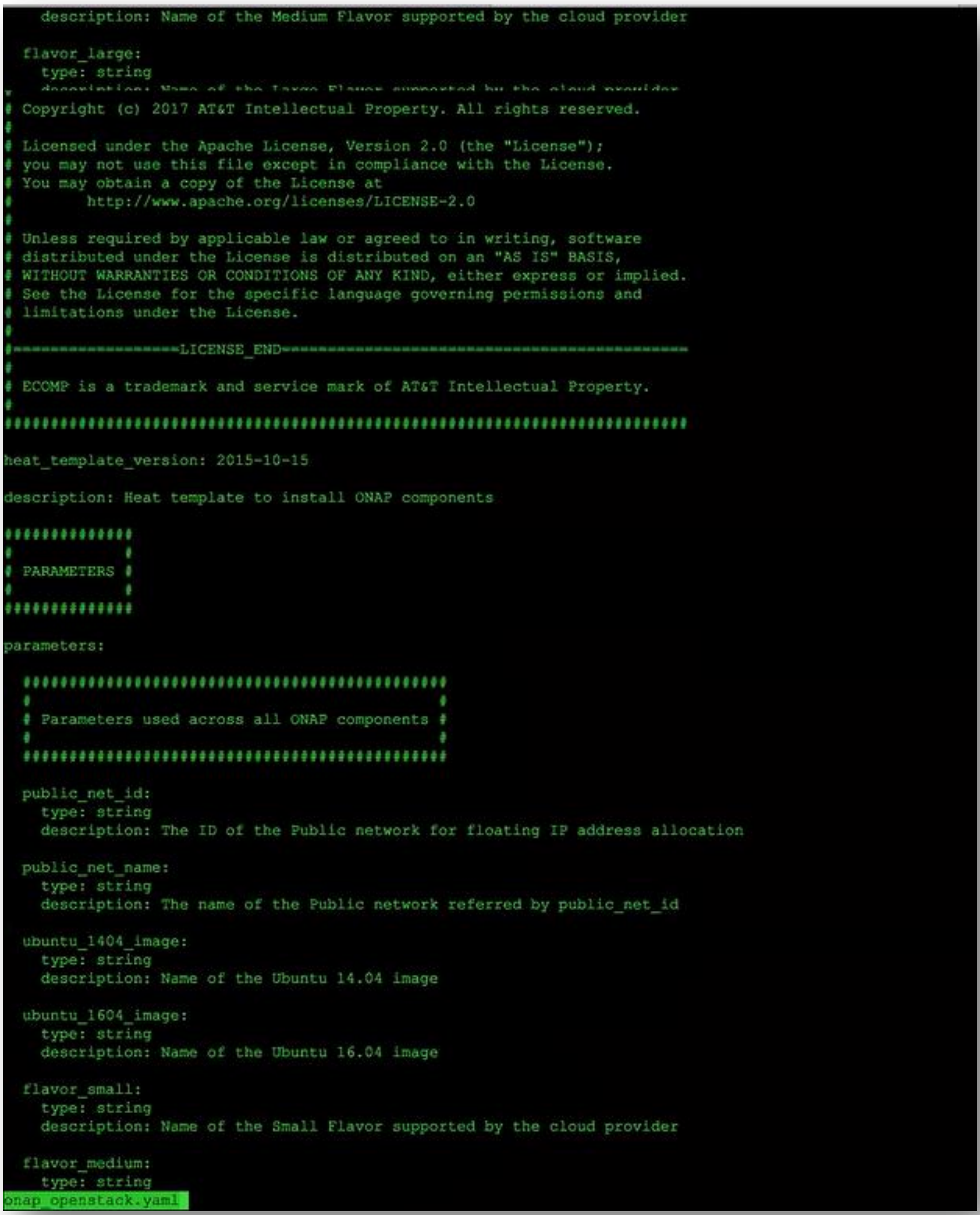
```
installer@eb7f6dd83ff9:~/config$ more cdapinputs.yaml
ubuntu1604image_id: 'ubuntu-16-04-cloud-amd64'
centos7image_id: 'CentOS-7'
flavor_id: 'm1.large'
security_group: 'onap_sg_6WA0'
public_net: 'external'
private_net: 'oam_onap_6WA0'
openstack:
  username: 'lushengji'
  password: 'ehfeiuwfhe4iyfjkrs'
  tenant_name: 'DCAE'
  auth_url: 'http://10.0.14.1/api/multicloud-titanium_cloud/v0/pod25_RegionOne/identity/v2.0'
  region: 'RegionOne'
keypair: 'onap_key_6WA0'
key_filename: '/opt/dcae/key'
location_prefix: 'dcae'
location_domain: '6WA0.dcaeg2.onap.org'
codesource_url: 'https://nexus.onap.org/service/local/repositories/raw/content'
codesource_version: 'org.onap.dcaegen2.deployments/releases'
installer@eb7f6dd83ff9:~/config$
```

– you can play with the flavor_id (‘m1.large’). Again the hold on work say: recommended large actually I think it would be a m1.xxlarge type of VM, only that it will have enough memory and storage to sustain large data analytics or creations but just for running it and keeper running our experience with Pod25 has been m1.large – it does not have to be xlarge, large seems to be to sustain TCA but if you do the medium then you will have the VMs up, it will work for a while but you can not really do anything with it. So that is one dimension that you can configure. Of course as everything is automated from the heat template, to deploy DCAE really you only need to issue the command of heat to create a stack (stack create) and the rest is automatic so to change those thing in the middle you have to either encounter an error and you going to fix it or you have to somehow interrupt the process. For all those input parameters the root of the information is in the heat template environment file, so I can show you how that is linked. I am back on my laptop



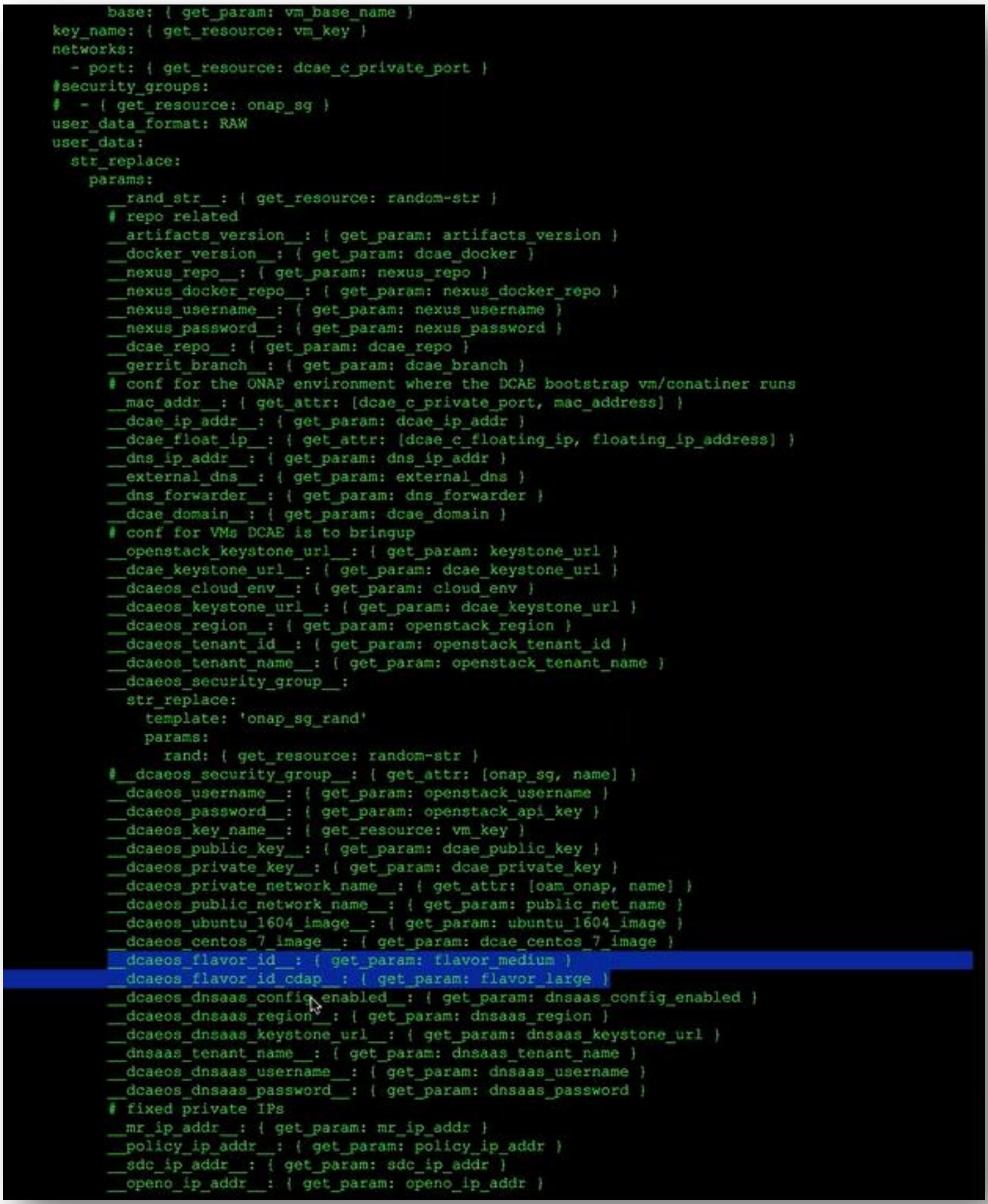
```
(openstack) MacBook-Pro:20171102 lji$
(openstack) MacBook-Pro:20171102 lji$
(openstack) MacBook-Pro:20171102 lji$ ls
-DCAE.                                 onap_openstack-Integration-SB-05.env
onap_openstack-DCAE.env                onap_openstack-Integration-SB-05.env-e
onap_openstack-DCAE.env-e              onap_openstack-Integration.env
onap_openstack-Integration-SB-00.env   onap_openstack-Integration.env-e
onap_openstack-Integration-SB-00.env-e onap_openstack.env
onap_openstack-Integration-SB-01.env   onap_openstack.yaml
onap_openstack-Integration-SB-01.env-e onap_openstack_dcae.env
onap_openstack-Integration-SB-02.env   onap_openstack_onapf.env
onap_openstack-Integration-SB-02.env-e onap_openstack_sb01.env
onap_openstack-Integration-SB-03.env   setenv-SB01-kangxi.sh
onap_openstack-Integration-SB-03.env-e setenv-SB01.sh
onap_openstack-Integration-SB-04.env   setenv-dcae.sh
onap_openstack-Integration-SB-04.env-e working
(openstack) MacBook-Pro:20171102 lji$ more onap_openstack.yaml
```

flavor.id is configured inside of the onap_openstack.yaml file.



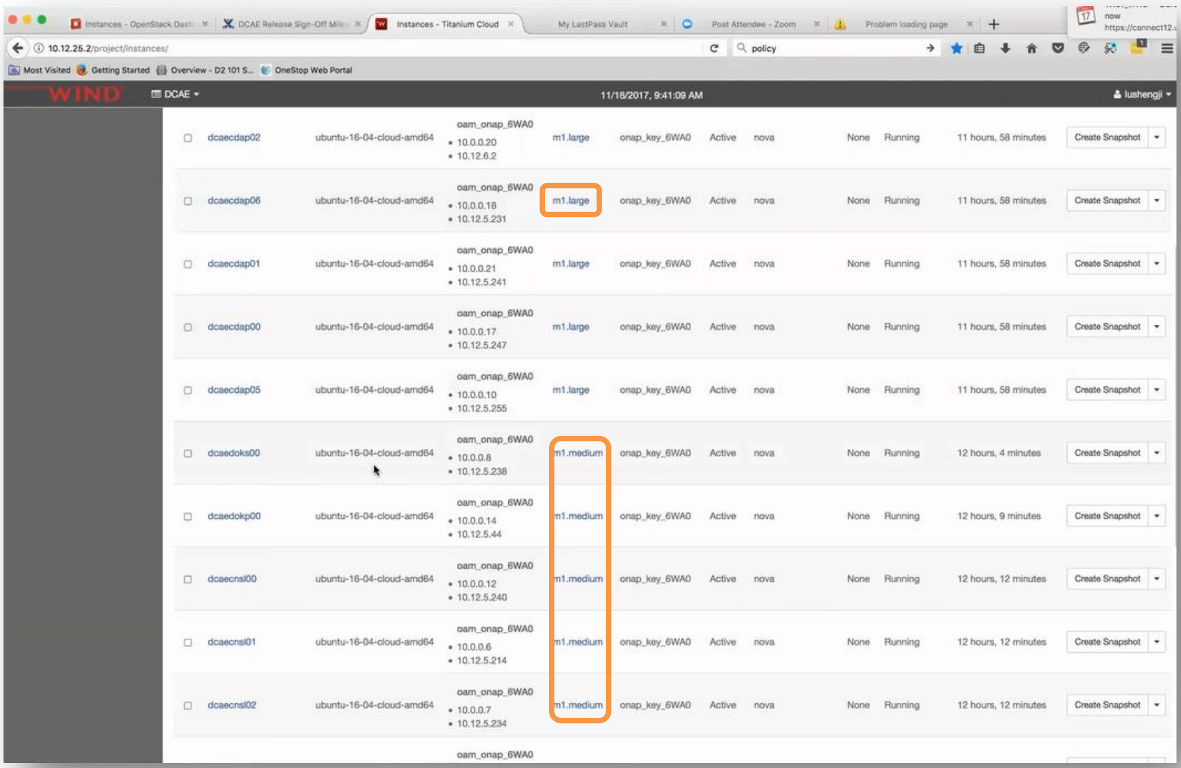
```
    description: Name of the Medium Flavor supported by the cloud provider

  flavor_large:
    type: string
    description: Name of the Large Flavor supported by the cloud provider
# Copyright (c) 2017 AT&T Intellectual Property. All rights reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#        http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
#=================LICENSE_END=========================================
#
# ECOMP is a trademark and service mark of AT&T Intellectual Property.
#
######################################################################

heat_template_version: 2015-10-15

description: Heat template to install ONAP components

##############
#            #
# PARAMETERS #
#            #
##############

parameters:

  ##############################################
  #                                            #
  # Parameters used across all ONAP components #
  #                                            #
  ##############################################

  public_net_id:
    type: string
    description: The ID of the Public network for floating IP address allocation

  public_net_name:
    type: string
    description: The name of the Public network referred by public_net_id

  ubuntu_1404_image:
    type: string
    description: Name of the Ubuntu 14.04 image

  ubuntu_1604_image:
    type: string
    description: Name of the Ubuntu 16.04 image

  flavor_small:
    type: string
    description: Name of the Small Flavor supported by the cloud provider

  flavor_medium:
    type: string
onap_openstack.yaml
```

Let’s go down into the DCAE section – you will see that there are 2 variables flavor_id:



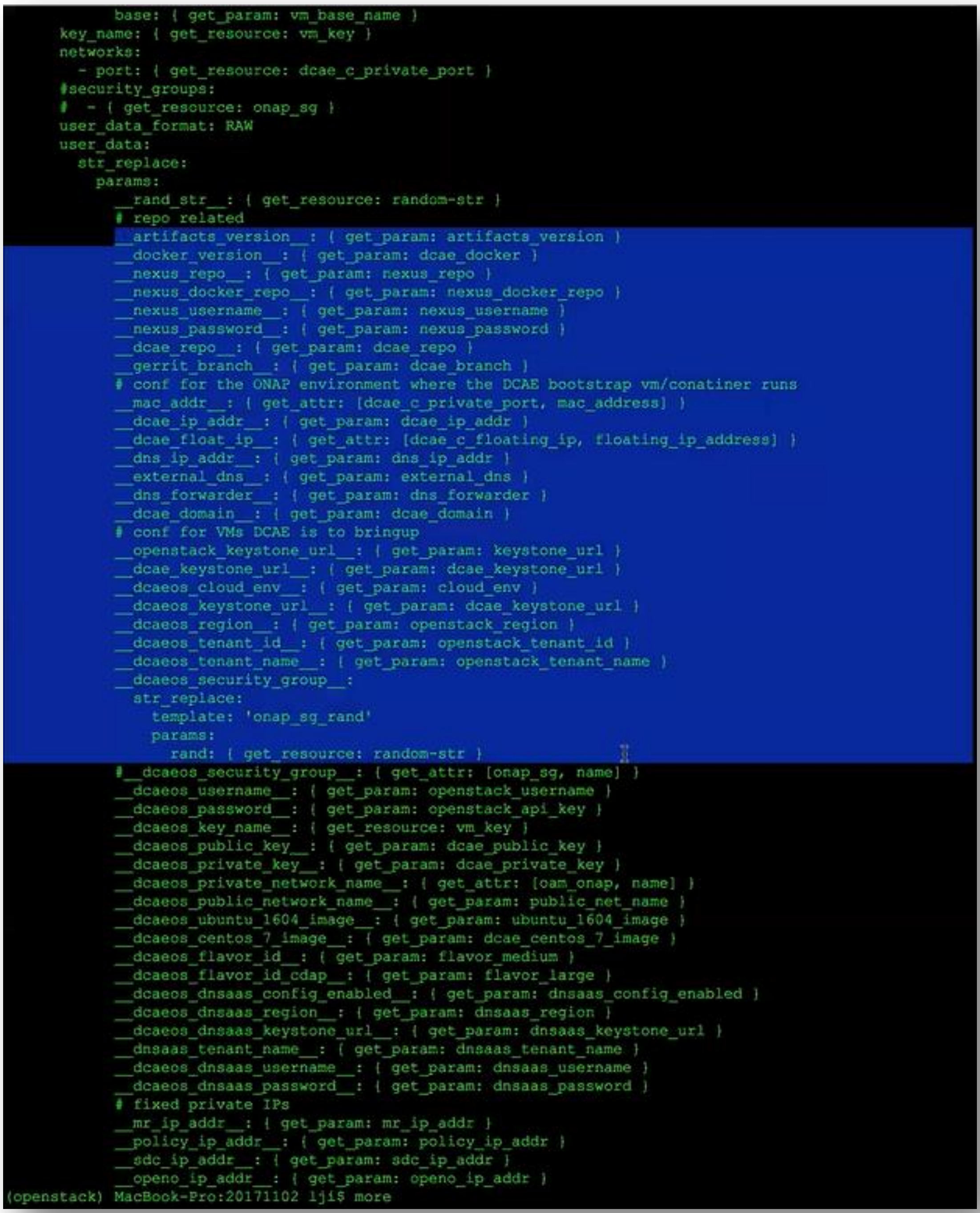
```
      base: { get_param: vm_base_name }
key_name: { get_resource: vm_key }
networks:
  - port: { get_resource: dcae_c_private_port }
#security_groups:
#  - { get_resource: onap_sg }
user_data_format: RAW
user_data:
  str_replace:
    params:
      __rand_str__: { get_resource: random-str }
      # repo related
      __artifacts_version__: { get_param: artifacts_version }
      __docker_version__: { get_param: dcae_docker }
      __nexus_repo__: { get_param: nexus_repo }
      __nexus_docker_repo__: { get_param: nexus_docker_repo }
      __nexus_username__: { get_param: nexus_username }
      __nexus_password__: { get_param: nexus_password }
      __dcae_repo__: { get_param: dcae_repo }
      __gerrit_branch__: { get_param: dcae_branch }
      # conf for the ONAP environment where the DCAE bootstrap vm/conatiner runs
      __mac_addr__: { get_attr: [dcae_c_private_port, mac_address] }
      __dcae_ip_addr__: { get_param: dcae_ip_addr }
      __dcae_float_ip__: { get_attr: [dcae_c_floating_ip, floating_ip_address] }
      __dns_ip_addr__: { get_param: dns_ip_addr }
      __external_dns__: { get_param: external_dns }
      __dns_forwarder__: { get_param: dns_forwarder }
      __dcae_domain__: { get_param: dcae_domain }
      # conf for VMs DCAE is to bringup
      __openstack_keystone_url__: { get_param: keystone_url }
      __dcae_keystone_url__: { get_param: dcae_keystone_url }
      __dcaeos_cloud_env__: { get_param: cloud_env }
      __dcaeos_keystone_url__: { get_param: dcae_keystone_url }
      __dcaeos_region__: { get_param: openstack_region }
      __dcaeos_tenant_id__: { get_param: openstack_tenant_id }
      __dcaeos_tenant_name__: { get_param: openstack_tenant_name }
      __dcaeos_security_group__:
        str_replace:
          template: 'onap_sg_rand'
          params:
            rand: { get_resource: random-str }
      #__dcaeos_security_group__: { get_attr: [onap_sg, name] }
      __dcaeos_username__: { get_param: openstack_username }
      __dcaeos_password__: { get_param: openstack_api_key }
      __dcaeos_key_name__: { get_resource: vm_key }
      __dcaeos_public_key__: { get_param: dcae_public_key }
      __dcaeos_private_key__: { get_param: dcae_private_key }
      __dcaeos_private_network_name__: { get_attr: [oam_onap, name] }
      __dcaeos_public_network_name__: { get_param: public_net_name }
      __dcaeos_ubuntu_1604_image__: { get_param: ubuntu_1604_image }
      __dcaeos_centos_7_image__: { get_param: dcae_centos_7_image }
      __dcaeos_flavor_id__: { get_param: flavor_medium }
      __dcaeos_flavor_id_cdap__: { get_param: flavor_large }
      __dcaeos_dnsaas_config_enabled__: { get_param: dnsaas_config_enabled }
      __dcaeos_dnsaas_region__: { get_param: dnsaas_region }
      __dcaeos_dnsaas_keystone_url__: { get_param: dnsaas_keystone_url }
      __dnsaas_tenant_name__: { get_param: dnsaas_tenant_name }
      __dcaeos_dnsaas_username__: { get_param: dnsaas_username }
      __dcaeos_dnsaas_password__: { get_param: dnsaas_password }
      # fixed private IPs
      __mr_ip_addr__: { get_param: mr_ip_addr }
      __policy_ip_addr__: { get_param: policy_ip_addr }
      __sdc_ip_addr__: { get_param: sdc_ip_addr }
      __openo_ip_addr__: { get_param: openo_ip_addr }
```

one is for the CDAP VM cluster what kind of flavor VM you want to use and the other one is for the rest of the VMs (we are using flavor medium) for the CDAP it is large. That is my local environment, on the carrier it is xlarge. Just to show better: cdap VMs are using m1.large and the rest of DCAE VMs are using m1.medium.



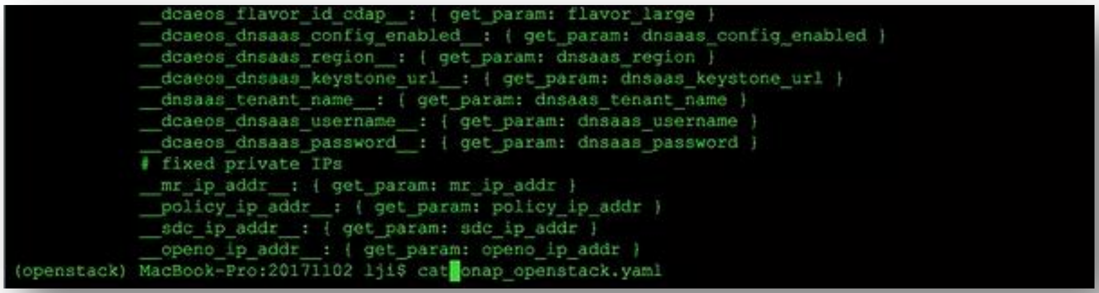
▢	dcaecdap02	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.20 • 10.12.6.2	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 58 minutes	Create Snapshot
▢	dcaecdap06	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.18 • 10.12.5.231	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 58 minutes	Create Snapshot
▢	dcaecdap01	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.21 • 10.12.5.241	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 58 minutes	Create Snapshot
▢	dcaecdap00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.17 • 10.12.5.247	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 58 minutes	Create Snapshot
▢	dcaecdap05	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.10 • 10.12.5.255	m1.large	onap_key_6WA0	Active	nova	None	Running	11 hours, 58 minutes	Create Snapshot
▢	dcaedoks00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.8 • 10.12.5.238	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 4 minutes	Create Snapshot
▢	dcaedokp00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.14 • 10.12.5.44	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 9 minutes	Create Snapshot
▢	dcaecnsl00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.12 • 10.12.5.240	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 12 minutes	Create Snapshot
▢	dcaecnsl01	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.6 • 10.12.5.214	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 12 minutes	Create Snapshot
▢	dcaecnsl02	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.7 • 10.12.5.234	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 12 minutes	Create Snapshot
			oam_onap_6WA0								

Depends on what kind of analytic application you want to include in the DCAE, for you demo you don’t need to have all the components loaded. For example if you are not using the CDAP version of the TCA you don’t have to deploy CDAP cluster at all, and to change those kind of things you need to go into the installer script where everything evokes from, you can find out where CDAP cluster is deployed and take those out so you don’t have that. Another trick I sometimes play is I just set this parameter (dcaeos_flavor_id_cdap: {get_param: flavor_large} to a flavor small and all it does to your openstack environment is it will invoke VM but nothing inside work of course so it is not recommended of course because you may break the integrity of the whole system however if you are sure of what you are doing that could be – you can use that kind of method to reduce resources used for your demo. Of course it is not for the production and anything like that but that is another place that you can customize. Since we are here, we will go over the parameters.

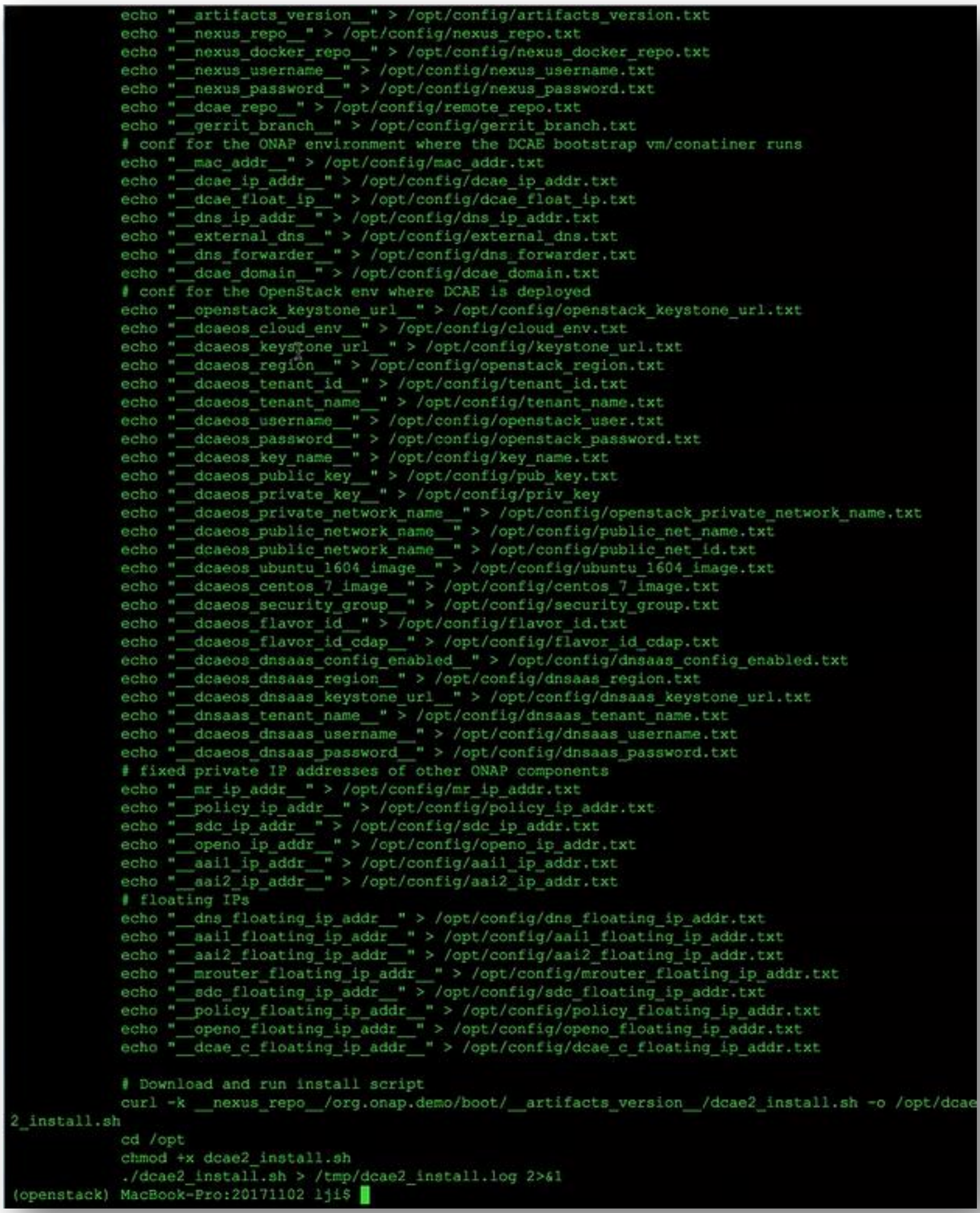


```
            base: { get_param: vm_base_name }
      key_name: { get_resource: vm_key }
      networks:
        - port: { get_resource: dcae_c_private_port }
      #security_groups:
      #  - { get_resource: onap_sg }
      user_data_format: RAW
      user_data:
        str_replace:
          params:
            __rand_str__: { get_resource: random-str }
            # repo related
            __artifacts_version__: { get_param: artifacts_version }
            __docker_version__: { get_param: dcae_docker }
            __nexus_repo__: { get_param: nexus_repo }
            __nexus_docker_repo__: { get_param: nexus_docker_repo }
            __nexus_username__: { get_param: nexus_username }
            __nexus_password__: { get_param: nexus_password }
            __dcae_repo__: { get_param: dcae_repo }
            __gerrit_branch__: { get_param: dcae_branch }
            # conf for the ONAP environment where the DCAE bootstrap vm/conatiner runs
            __mac_addr__: { get_attr: [dcae_c_private_port, mac_address] }
            __dcae_ip_addr__: { get_param: dcae_ip_addr }
            __dcae_float_ip__: { get_attr: [dcae_c_floating_ip, floating_ip_address] }
            __dns_ip_addr__: { get_param: dns_ip_addr }
            __external_dns__: { get_param: external_dns }
            __dns_forwarder__: { get_param: dns_forwarder }
            __dcae_domain__: { get_param: dcae_domain }
            # conf for VMs DCAE is to bringup
            __openstack_keystone_url__: { get_param: keystone_url }
            __dcae_keystone_url__: { get_param: dcae_keystone_url }
            __dcaeos_cloud_env__: { get_param: cloud_env }
            __dcaeos_keystone_url__: { get_param: dcae_keystone_url }
            __dcaeos_region__: { get_param: openstack_region }
            __dcaeos_tenant_id__: { get_param: openstack_tenant_id }
            __dcaeos_tenant_name__: { get_param: openstack_tenant_name }
            __dcaeos_security_group__:
              str_replace:
                template: 'onap_sg_rand'
                params:
                  rand: { get_resource: random-str }
            #__dcaeos_security_group__: { get_attr: [onap_sg, name] }
            __dcaeos_username__: { get_param: openstack_username }
            __dcaeos_password__: { get_param: openstack_api_key }
            __dcaeos_key_name__: { get_resource: vm_key }
            __dcaeos_public_key__: { get_param: dcae_public_key }
            __dcaeos_private_key__: { get_param: dcae_private_key }
            __dcaeos_private_network_name__: { get_attr: [oam_onap, name] }
            __dcaeos_public_network_name__: { get_param: public_net_name }
            __dcaeos_ubuntu_1604_image__: { get_param: ubuntu_1604_image }
            __dcaeos_centos_7_image__: { get_param: dcae_centos_7_image }
            __dcaeos_flavor_id__: { get_param: flavor_medium }
            __dcaeos_flavor_id_cdap__: { get_param: flavor_large }
            __dcaeos_dnsaas_config_enabled__: { get_param: dnsaas_config_enabled }
            __dcaeos_dnsaas_region__: { get_param: dnsaas_region }
            __dcaeos_dnsaas_keystone_url__: { get_param: dnsaas_keystone_url }
            __dnsaas_tenant_name__: { get_param: dnsaas_tenant_name }
            __dcaeos_dnsaas_username__: { get_param: dnsaas_username }
            __dcaeos_dnsaas_password__: { get_param: dnsaas_password }
            # fixed private IPs
            __mr_ip_addr__: { get_param: mr_ip_addr }
            __policy_ip_addr__: { get_param: policy_ip_addr }
            __sdc_ip_addr__: { get_param: sdc_ip_addr }
            __openo_ip_addr__: { get_param: openo_ip_addr }
(openstack) MacBook-Pro:20171102 lji$ more
```

So you can see here is where all those parameters are written into the DCAE bootstrap.



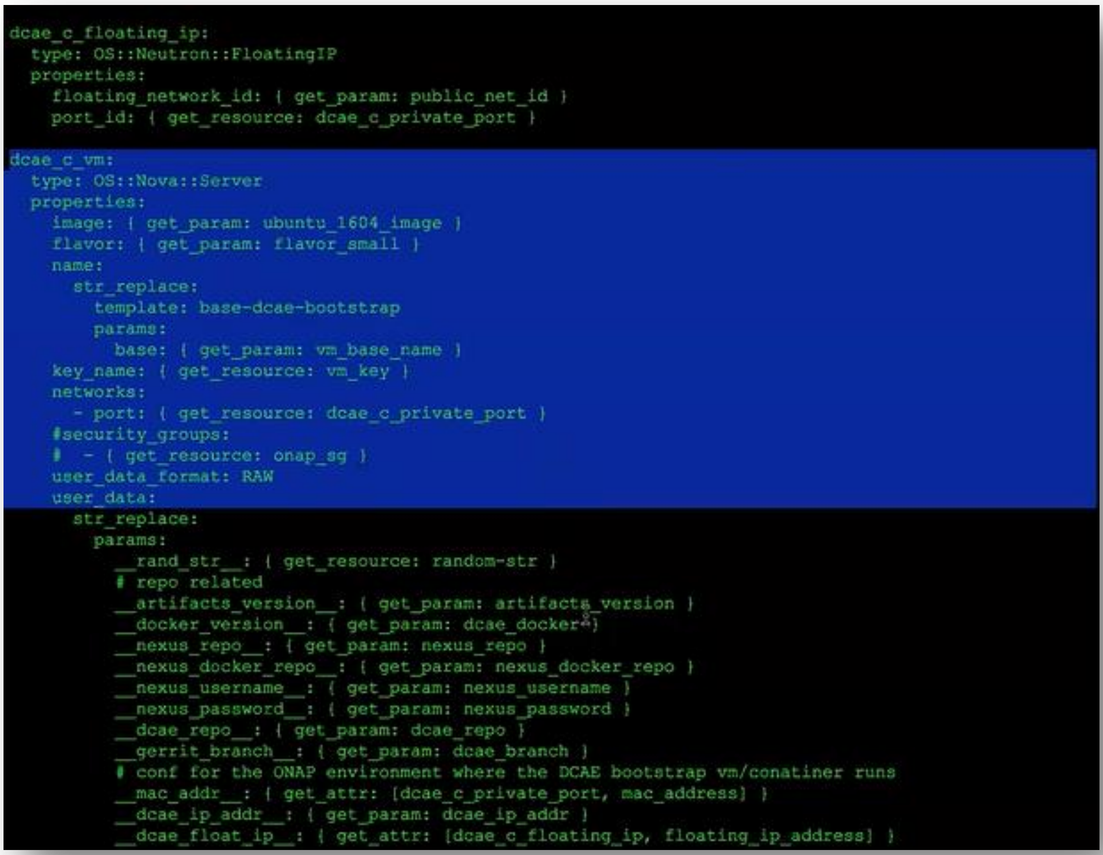
```
            __dcaeos_flavor_id_cdap__: { get_param: flavor_large }
            __dcaeos_dnsaas_config_enabled__: { get_param: dnsaas_config_enabled }
            __dcaeos_dnsaas_region__: { get_param: dnsaas_region }
            __dcaeos_dnsaas_keystone_url__: { get_param: dnsaas_keystone_url }
            __dnsaas_tenant_name__: { get_param: dnsaas_tenant_name }
            __dcaeos_dnsaas_username__: { get_param: dnsaas_username }
            __dcaeos_dnsaas_password__: { get_param: dnsaas_password }
            # fixed private IPs
            __mr_ip_addr__: { get_param: mr_ip_addr }
            __policy_ip_addr__: { get_param: policy_ip_addr }
            __sdc_ip_addr__: { get_param: sdc_ip_addr }
            __openo_ip_addr__: { get_param: openo_ip_addr }
(openstack) MacBook-Pro:20171102 lji$ cat onap_openstack.yaml
```



```
            echo "__artifacts_version__" > /opt/config/artifacts_version.txt
            echo "__nexus_repo__" > /opt/config/nexus_repo.txt
            echo "__nexus_docker_repo__" > /opt/config/nexus_docker_repo.txt
            echo "__nexus_username__" > /opt/config/nexus_username.txt
            echo "__nexus_password__" > /opt/config/nexus_password.txt
            echo "__dcae_repo__" > /opt/config/remote_repo.txt
            echo "__gerrit_branch__" > /opt/config/gerrit_branch.txt
            # conf for the ONAP environment where the DCAE bootstrap vm/conatiner runs
            echo "__mac_addr__" > /opt/config/mac_addr.txt
            echo "__dcae_ip_addr__" > /opt/config/dcae_ip_addr.txt
            echo "__dcae_float_ip__" > /opt/config/dcae_float_ip.txt
            echo "__dns_ip_addr__" > /opt/config/dns_ip_addr.txt
            echo "__external_dns__" > /opt/config/external_dns.txt
            echo "__dns_forwarder__" > /opt/config/dns_forwarder.txt
            echo "__dcae_domain__" > /opt/config/dcae_domain.txt
            # conf for the OpenStack env where DCAE is deployed
            echo "__openstack_keystone_url__" > /opt/config/openstack_keystone_url.txt
            echo "__dcaeos_cloud_env__" > /opt/config/cloud_env.txt
            echo "__dcaeos_keystone_url__" > /opt/config/keystone_url.txt
            echo "__dcaeos_region__" > /opt/config/openstack_region.txt
            echo "__dcaeos_tenant_id__" > /opt/config/tenant_id.txt
            echo "__dcaeos_tenant_name__" > /opt/config/tenant_name.txt
            echo "__dcaeos_username__" > /opt/config/openstack_user.txt
            echo "__dcaeos_password__" > /opt/config/openstack_password.txt
            echo "__dcaeos_key_name__" > /opt/config/key_name.txt
            echo "__dcaeos_public_key__" > /opt/config/pub_key.txt
            echo "__dcaeos_private_key__" > /opt/config/priv_key
            echo "__dcaeos_private_network_name__" > /opt/config/openstack_private_network_name.txt
            echo "__dcaeos_public_network_name__" > /opt/config/public_net_name.txt
            echo "__dcaeos_public_network_name__" > /opt/config/public_net_id.txt
            echo "__dcaeos_ubuntu_1604_image__" > /opt/config/ubuntu_1604_image.txt
            echo "__dcaeos_centos_7_image__" > /opt/config/centos_7_image.txt
            echo "__dcaeos_security_group__" > /opt/config/security_group.txt
            echo "__dcaeos_flavor_id__" > /opt/config/flavor_id.txt
            echo "__dcaeos_flavor_id_cdap__" > /opt/config/flavor_id_cdap.txt
            echo "__dcaeos_dnsaas_config_enabled__" > /opt/config/dnsaas_config_enabled.txt
            echo "__dcaeos_dnsaas_region__" > /opt/config/dnsaas_region.txt
            echo "__dcaeos_dnsaas_keystone_url__" > /opt/config/dnsaas_keystone_url.txt
            echo "__dnsaas_tenant_name__" > /opt/config/dnsaas_tenant_name.txt
            echo "__dcaeos_dnsaas_username__" > /opt/config/dnsaas_username.txt
            echo "__dcaeos_dnsaas_password__" > /opt/config/dnsaas_password.txt
            # fixed private IP addresses of other ONAP components
            echo "__mr_ip_addr__" > /opt/config/mr_ip_addr.txt
            echo "__policy_ip_addr__" > /opt/config/policy_ip_addr.txt
            echo "__sdc_ip_addr__" > /opt/config/sdc_ip_addr.txt
            echo "__openo_ip_addr__" > /opt/config/openo_ip_addr.txt
            echo "__aai1_ip_addr__" > /opt/config/aai1_ip_addr.txt
            echo "__aai2_ip_addr__" > /opt/config/aai2_ip_addr.txt
            # floating IPs
            echo "__dns_floating_ip_addr__" > /opt/config/dns_floating_ip_addr.txt
            echo "__aai1_floating_ip_addr__" > /opt/config/aai1_floating_ip_addr.txt
            echo "__aai2_floating_ip_addr__" > /opt/config/aai2_floating_ip_addr.txt
            echo "__mrouter_floating_ip_addr__" > /opt/config/mrouter_floating_ip_addr.txt
            echo "__sdc_floating_ip_addr__" > /opt/config/sdc_floating_ip_addr.txt
            echo "__policy_floating_ip_addr__" > /opt/config/policy_floating_ip_addr.txt
            echo "__openo_floating_ip_addr__" > /opt/config/openo_floating_ip_addr.txt
            echo "__dcae_c_floating_ip_addr__" > /opt/config/dcae_c_floating_ip_addr.txt

            # Download and run install script
            curl -k __nexus_repo__/org.onap.demo/boot/__artifacts_version__/dcae2_install.sh -o /opt/dcae
2_install.sh
            cd /opt
            chmod +x dcae2_install.sh
            ./dcae2_install.sh > /tmp/dcae2_install.log 2>&1
(openstack) MacBook-Pro:20171102 lji$ 
```

All these are either parameters provided from the env file while you are running the stack creation or there are some resource that found-up from this heat. All these files, all these parameters are very straight forward (onap_openstack.yaml) the logic here is:

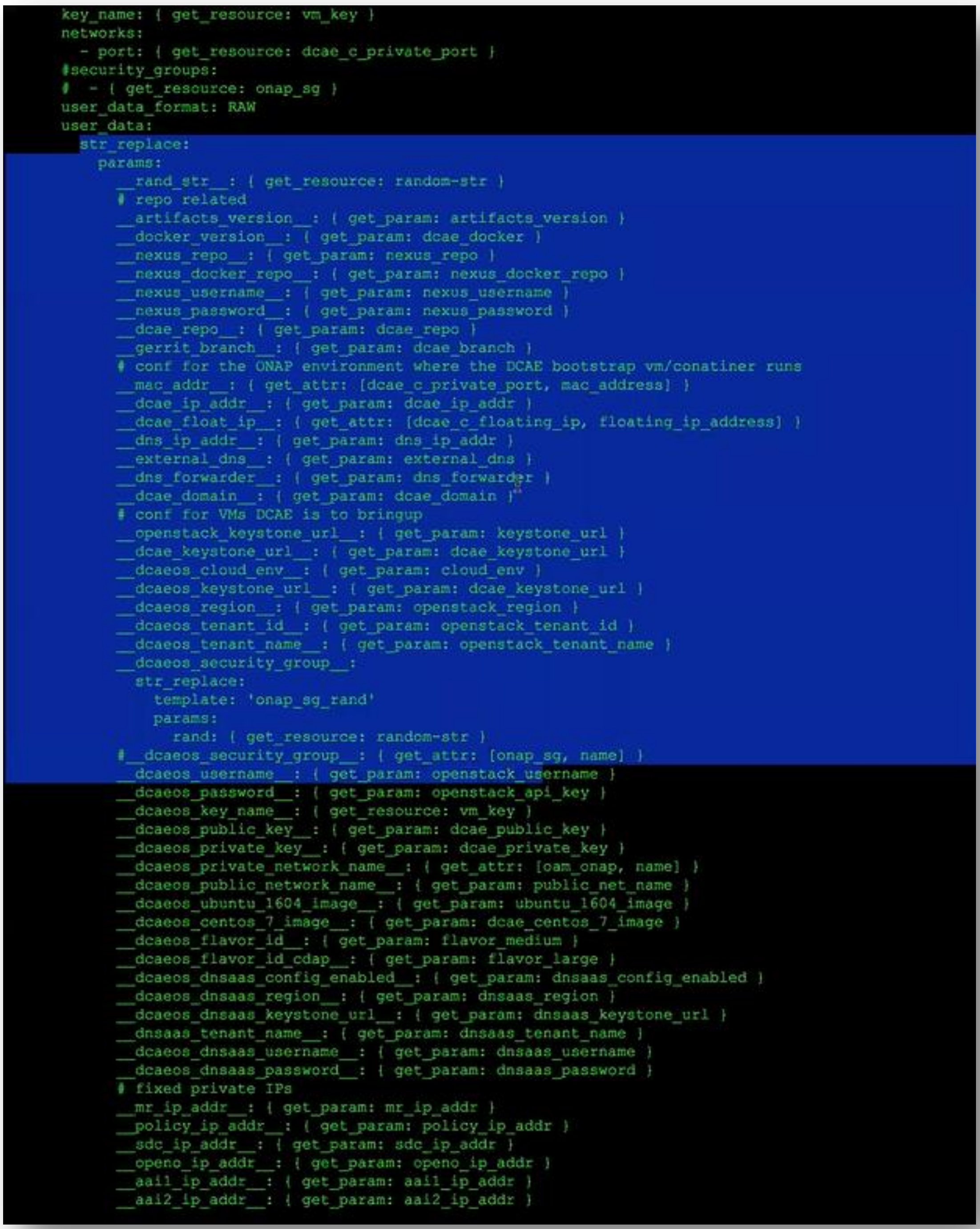
The bootstrap VM:



```
dcae_c_floating_ip:
  type: OS::Neutron::FloatingIP
  properties:
    floating_network_id: { get_param: public_net_id }
    port_id: { get_resource: dcae_c_private_port }

dcae_c_vm:
  type: OS::Nova::Server
  properties:
    image: { get_param: ubuntu_1604_image }
    flavor: { get_param: flavor_small }
    name:
      str_replace:
        template: base-dcae-bootstrap
        params:
          base: { get_param: vm_base_name }
    key_name: { get_resource: vm_key }
    networks:
      - port: { get_resource: dcae_c_private_port }
    #security_groups:
    #  - { get_resource: onap_sg }
    user_data_format: RAW
    user_data:
      str_replace:
        params:
          __rand_str__: { get_resource: random-str }
          # repo related
          __artifacts_version__: { get_param: artifacts_version }
          __docker_version__: { get_param: dcae_docker }
          __nexus_repo__: { get_param: nexus_repo }
          __nexus_docker_repo__: { get_param: nexus_docker_repo }
          __nexus_username__: { get_param: nexus_username }
          __nexus_password__: { get_param: nexus_password }
          __dcae_repo__: { get_param: dcae_repo }
          __gerrit_branch__: { get_param: dcae_branch }
          # conf for the ONAP environment where the DCAE bootstrap vm/conatiner runs
          __mac_addr__: { get_attr: [dcae_c_private_port, mac_address] }
          __dcae_ip_addr__: { get_param: dcae_ip_addr }
          __dcae_float_ip__: { get_attr: [dcae_c_floating_ip, floating_ip_address] }
```

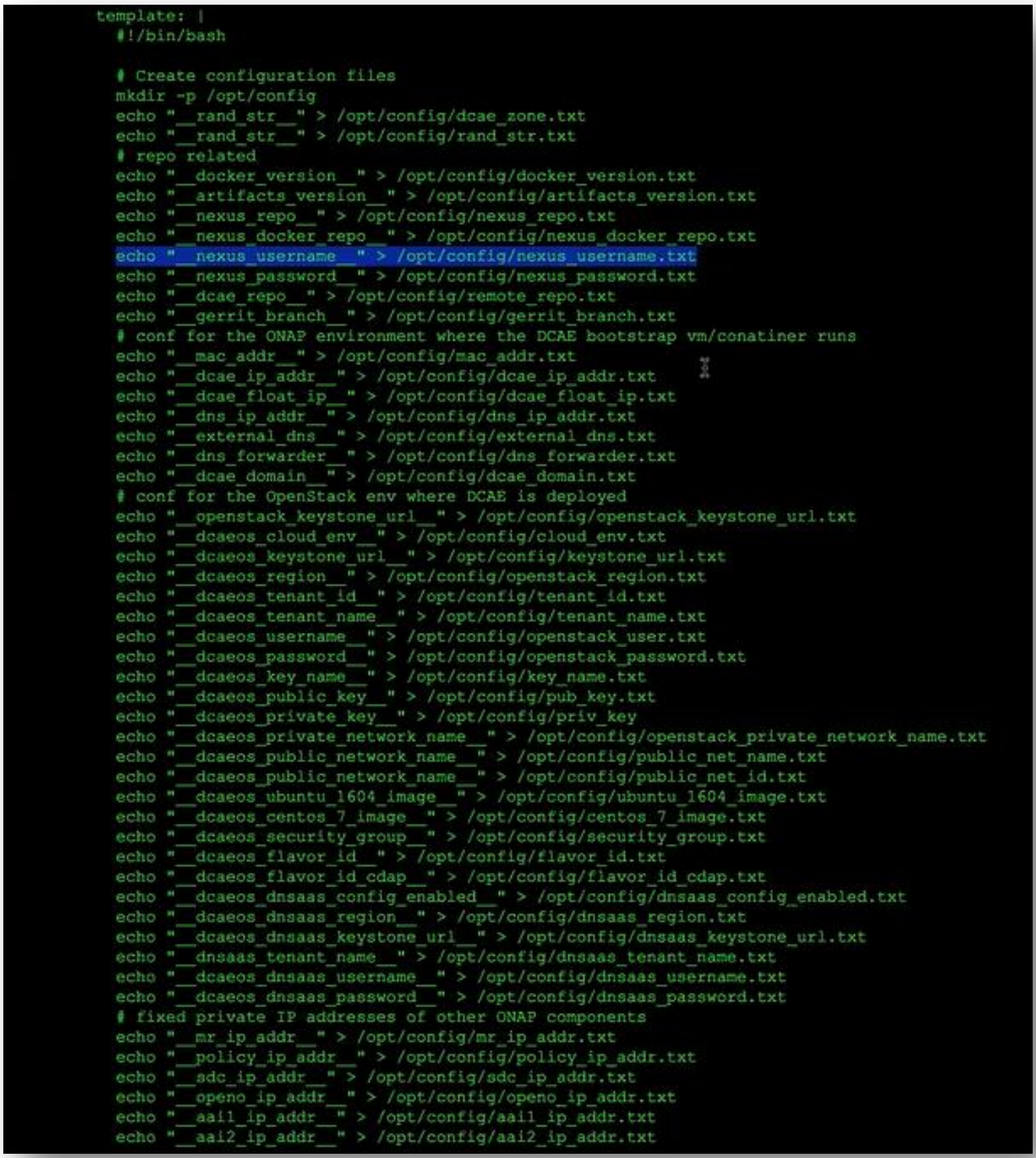
dcae_c_vm:
  type: OS::Nova::Server
  properties:

this is the cloud init part VM initialization, so it will get all those values



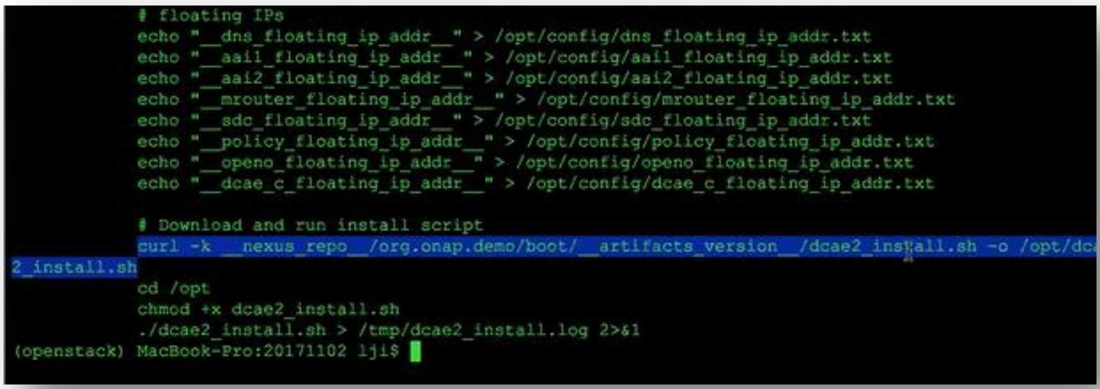
```
      key_name: { get_resource: vm_key }
      networks:
        - port: { get_resource: dcae_c_private_port }
      #security_groups:
      #  - { get_resource: onap_sg }
      user_data_format: RAW
      user_data:
        str_replace:
          params:
            __rand_str__: { get_resource: random-str }
            # repo related
            __artifacts_version__: { get_param: artifacts_version }
            __docker_version__: { get_param: dcae_docker }
            __nexus_repo__: { get_param: nexus_repo }
            __nexus_docker_repo__: { get_param: nexus_docker_repo }
            __nexus_username__: { get_param: nexus_username }
            __nexus_password__: { get_param: nexus_password }
            __dcae_repo__: { get_param: dcae_repo }
            __gerrit_branch__: { get_param: dcae_branch }
            # conf for the ONAP environment where the DCAE bootstrap vm/conatiner runs
            __mac_addr__: { get_attr: [dcae_c_private_port, mac_address] }
            __dcae_ip_addr__: { get_param: dcae_ip_addr }
            __dcae_float_ip__: { get_attr: [dcae_c_floating_ip, floating_ip_address] }
            __dns_ip_addr__: { get_param: dns_ip_addr }
            __external_dns__: { get_param: external_dns }
            __dns_forwarder__: { get_param: dns_forwarder }
            __dcae_domain__: { get_param: dcae_domain }
            # conf for VMs DCAE is to bringup
            __openstack_keystone_url__: { get_param: keystone_url }
            __dcae_keystone_url__: { get_param: dcae_keystone_url }
            __dcaeos_cloud_env__: { get_param: cloud_env }
            __dcaeos_keystone_url__: { get_param: dcae_keystone_url }
            __dcaeos_region__: { get_param: openstack_region }
            __dcaeos_tenant_id__: { get_param: openstack_tenant_id }
            __dcaeos_tenant_name__: { get_param: openstack_tenant_name }
            __dcaeos_security_group__:
              str_replace:
                template: 'onap_sg_rand'
                params:
                  rand: { get_resource: random-str }
            #__dcaeos_security_group__: { get_attr: [onap_sg, name] }
            __dcaeos_username__: { get_param: openstack_username }
            __dcaeos_password__: { get_param: openstack_api_key }
            __dcaeos_key_name__: { get_resource: vm_key }
            __dcaeos_public_key__: { get_param: dcae_public_key }
            __dcaeos_private_key__: { get_param: dcae_private_key }
            __dcaeos_private_network_name__: { get_attr: [oam_onap, name] }
            __dcaeos_public_network_name__: { get_param: public_net_name }
            __dcaeos_ubuntu_1604_image__: { get_param: ubuntu_1604_image }
            __dcaeos_centos_7_image__: { get_param: dcae_centos_7_image }
            __dcaeos_flavor_id__: { get_param: flavor_medium }
            __dcaeos_flavor_id_cdap__: { get_param: flavor_large }
            __dcaeos_dnsaas_config_enabled__: { get_param: dnsaas_config_enabled }
            __dcaeos_dnsaas_region__: { get_param: dnsaas_region }
            __dcaeos_dnsaas_keystone_url__: { get_param: dnsaas_keystone_url }
            __dnsaas_tenant_name__: { get_param: dnsaas_tenant_name }
            __dcaeos_dnsaas_username__: { get_param: dnsaas_username }
            __dcaeos_dnsaas_password__: { get_param: dnsaas_password }
            # fixed private IPs
            __mr_ip_addr__: { get_param: mr_ip_addr }
            __policy_ip_addr__: { get_param: policy_ip_addr }
            __sdc_ip_addr__: { get_param: sdc_ip_addr }
            __openo_ip_addr__: { get_param: openo_ip_addr }
            __aai1_ip_addr__: { get_param: aai1_ip_addr }
            __aai2_ip_addr__: { get_param: aai2_ip_addr }
```

str_replace:
        params:
into some variables and it will just write those variables (echo "__) down to disk files inside that VM,



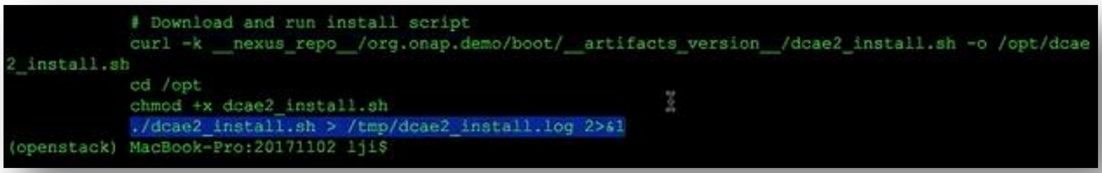
```
        template: |
          #!/bin/bash

          # Create configuration files
          mkdir -p /opt/config
          echo "__rand_str__" > /opt/config/dcae_zone.txt
          echo "__rand_str__" > /opt/config/rand_str.txt
          # repo related
          echo "__docker_version__" > /opt/config/docker_version.txt
          echo "__artifacts_version__" > /opt/config/artifacts_version.txt
          echo "__nexus_repo__" > /opt/config/nexus_repo.txt
          echo "__nexus_docker_repo__" > /opt/config/nexus_docker_repo.txt
          echo "__nexus_username__" > /opt/config/nexus_username.txt
          echo "__nexus_password__" > /opt/config/nexus_password.txt
          echo "__dcae_repo__" > /opt/config/remote_repo.txt
          echo "__gerrit_branch__" > /opt/config/gerrit_branch.txt
          # conf for the ONAP environment where the DCAE bootstrap vm/conatiner runs
          echo "__mac_addr__" > /opt/config/mac_addr.txt
          echo "__dcae_ip_addr__" > /opt/config/dcae_ip_addr.txt
          echo "__dcae_float_ip__" > /opt/config/dcae_float_ip.txt
          echo "__dns_ip_addr__" > /opt/config/dns_ip_addr.txt
          echo "__external_dns__" > /opt/config/external_dns.txt
          echo "__dns_forwarder__" > /opt/config/dns_forwarder.txt
          echo "__dcae_domain__" > /opt/config/dcae_domain.txt
          # conf for the OpenStack env where DCAE is deployed
          echo "__openstack_keystone_url__" > /opt/config/openstack_keystone_url.txt
          echo "__dcaeos_cloud_env__" > /opt/config/cloud_env.txt
          echo "__dcaeos_keystone_url__" > /opt/config/keystone_url.txt
          echo "__dcaeos_region__" > /opt/config/openstack_region.txt
          echo "__dcaeos_tenant_id__" > /opt/config/tenant_id.txt
          echo "__dcaeos_tenant_name__" > /opt/config/tenant_name.txt
          echo "__dcaeos_username__" > /opt/config/openstack_user.txt
          echo "__dcaeos_password__" > /opt/config/openstack_password.txt
          echo "__dcaeos_key_name__" > /opt/config/key_name.txt
          echo "__dcaeos_public_key__" > /opt/config/pub_key.txt
          echo "__dcaeos_private_key__" > /opt/config/priv_key
          echo "__dcaeos_private_network_name__" > /opt/config/openstack_private_network_name.txt
          echo "__dcaeos_public_network_name__" > /opt/config/public_net_name.txt
          echo "__dcaeos_public_network_name__" > /opt/config/public_net_id.txt
          echo "__dcaeos_ubuntu_1604_image__" > /opt/config/ubuntu_1604_image.txt
          echo "__dcaeos_centos_7_image__" > /opt/config/centos_7_image.txt
          echo "__dcaeos_security_group__" > /opt/config/security_group.txt
          echo "__dcaeos_flavor_id__" > /opt/config/flavor_id.txt
          echo "__dcaeos_flavor_id_cdap__" > /opt/config/flavor_id_cdap.txt
          echo "__dcaeos_dnsaas_config_enabled__" > /opt/config/dnsaas_config_enabled.txt
          echo "__dcaeos_dnsaas_region__" > /opt/config/dnsaas_region.txt
          echo "__dcaeos_dnsaas_keystone_url__" > /opt/config/dnsaas_keystone_url.txt
          echo "__dnsaas_tenant_name__" > /opt/config/dnsaas_tenant_name.txt
          echo "__dcaeos_dnsaas_username__" > /opt/config/dnsaas_username.txt
          echo "__dcaeos_dnsaas_password__" > /opt/config/dnsaas_password.txt
          # fixed private IP addresses of other ONAP components
          echo "__mr_ip_addr__" > /opt/config/mr_ip_addr.txt
          echo "__policy_ip_addr__" > /opt/config/policy_ip_addr.txt
          echo "__sdc_ip_addr__" > /opt/config/sdc_ip_addr.txt
          echo "__openo_ip_addr__" > /opt/config/openo_ip_addr.txt
          echo "__aai1_ip_addr__" > /opt/config/aai1_ip_addr.txt
          echo "__aai2_ip_addr__" > /opt/config/aai2_ip_addr.txt
```

so this is the heat template, so at the end you will see the steps



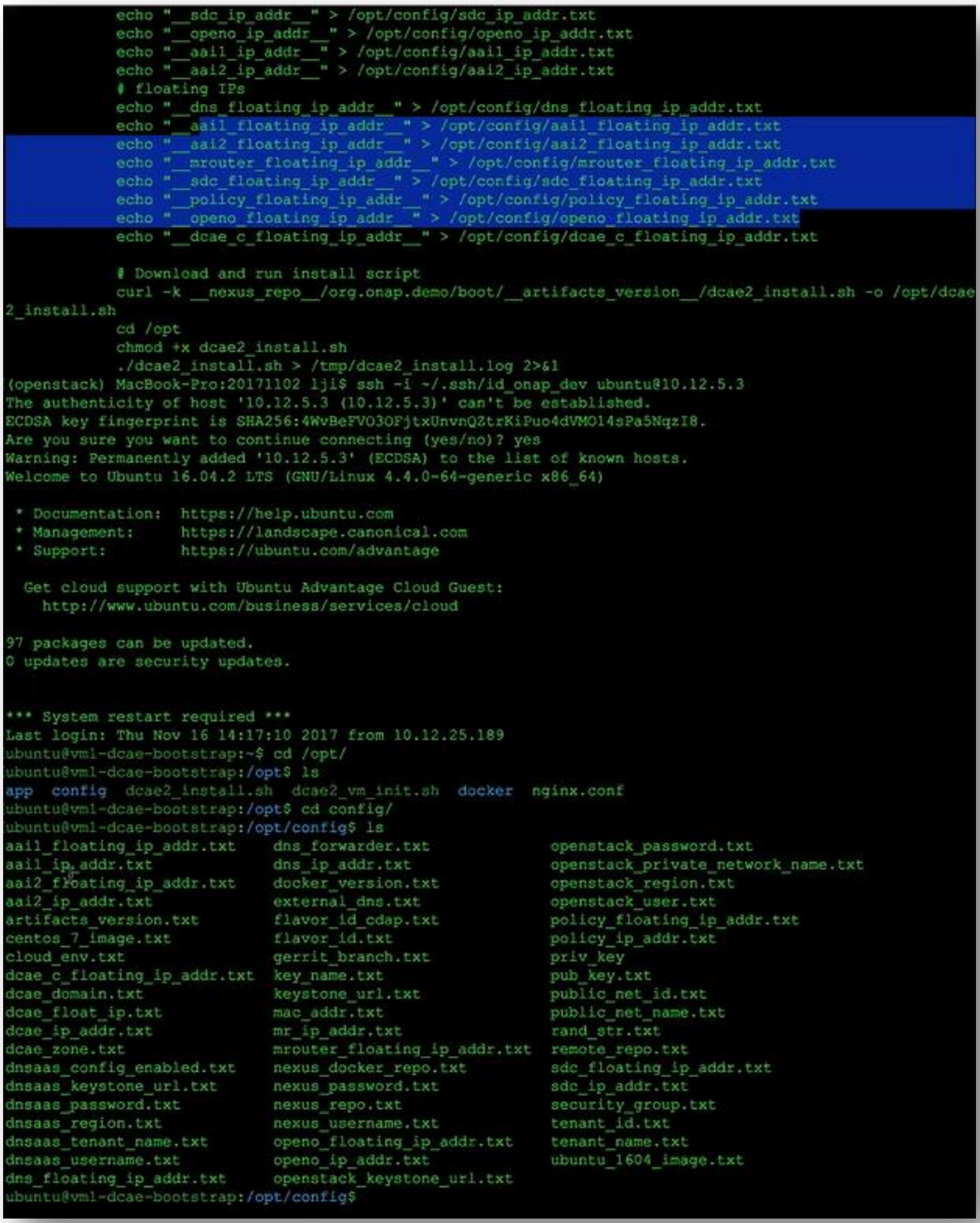
```
            # floating IPs
            echo "__dns_floating_ip_addr__" > /opt/config/dns_floating_ip_addr.txt
            echo "__aai1_floating_ip_addr__" > /opt/config/aai1_floating_ip_addr.txt
            echo "__aai2_floating_ip_addr__" > /opt/config/aai2_floating_ip_addr.txt
            echo "__mrouter_floating_ip_addr__" > /opt/config/mrouter_floating_ip_addr.txt
            echo "__sdc_floating_ip_addr__" > /opt/config/sdc_floating_ip_addr.txt
            echo "__policy_floating_ip_addr__" > /opt/config/policy_floating_ip_addr.txt
            echo "__openo_floating_ip_addr__" > /opt/config/openo_floating_ip_addr.txt
            echo "__dcae_c_floating_ip_addr__" > /opt/config/dcae_c_floating_ip_addr.txt

            # Download and run install script
            curl -k __nexus_repo__/org.onap.demo/boot/__artifacts_version__/dcae2_install.sh -o /opt/dc
2_install.sh
            cd /opt
            chmod +x dcae2_install.sh
            ./dcae2_install.sh > /tmp/dcae2_install.log 2>&1
(openstack) MacBook-Pro:20171102 ljis$
```

(curl –k __nexus_repo__/org.onap.demo/boot/__artifacts_version__/dcae2_install.sh –o /opt/dcae2_install.sh) it is downloading the scripts and running installation script (./dcae2_install.sh > /tmp/dcae2_install.log 2>&1) so that is all the bootstrap VM does.



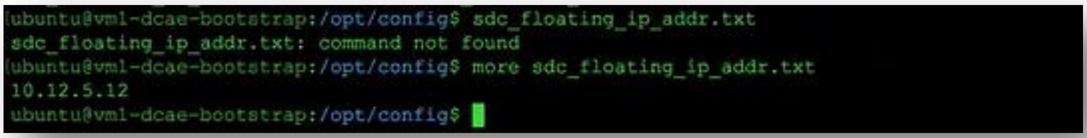
```
            # Download and run install script
            curl -k __nexus_repo__/org.onap.demo/boot/__artifacts_version__/dcae2_install.sh -o /opt/dcae
2_install.sh
            cd /opt
            chmod +x dcae2_install.sh
            ./dcae2_install.sh > /tmp/dcae2_install.log 2>&1
(openstack) MacBook-Pro:20171102 lji$
```

And now I am going to (ssh –i ~/.ssh/id_onap_dev ubusntu@10.12.5.3) inside the bootstrap VM to show the results of the execution of the cloud init.
You can see that all echo lines becomes disk files.



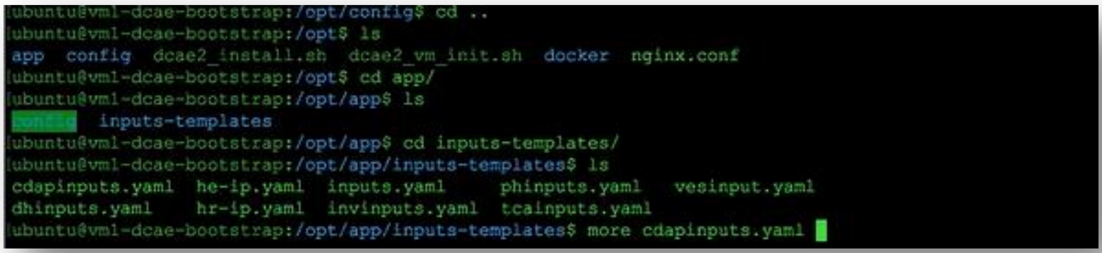
```
            echo "__sdc_ip_addr__" > /opt/config/sdc_ip_addr.txt
            echo "__openo_ip_addr__" > /opt/config/openo_ip_addr.txt
            echo "__aai1_ip_addr__" > /opt/config/aai1_ip_addr.txt
            echo "__aai2_ip_addr__" > /opt/config/aai2_ip_addr.txt
            # floating IPs
            echo "__dns_floating_ip_addr__" > /opt/config/dns_floating_ip_addr.txt
            echo "__aai1_floating_ip_addr__" > /opt/config/aai1_floating_ip_addr.txt
            echo "__aai2_floating_ip_addr__" > /opt/config/aai2_floating_ip_addr.txt
            echo "__mrouter_floating_ip_addr__" > /opt/config/mrouter_floating_ip_addr.txt
            echo "__sdc_floating_ip_addr__" > /opt/config/sdc_floating_ip_addr.txt
            echo "__policy_floating_ip_addr__" > /opt/config/policy_floating_ip_addr.txt
            echo "__openo_floating_ip_addr__" > /opt/config/openo_floating_ip_addr.txt
            echo "__dcae_c_floating_ip_addr__" > /opt/config/dcae_c_floating_ip_addr.txt

            # Download and run install script
            curl -k __nexus_repo__/org.onap.demo/boot/__artifacts_version__/dcae2_install.sh -o /opt/dcae
2_install.sh
            cd /opt
            chmod +x dcae2_install.sh
            ./dcae2_install.sh > /tmp/dcae2_install.log 2>&1
(openstack) MacBook-Pro:20171102 lji$ ssh -i ~/.ssh/id_onap_dev ubuntu@10.12.5.3
The authenticity of host '10.12.5.3 (10.12.5.3)' can't be established.
ECDSA key fingerprint is SHA256:4WvBeFVO3OFjtxUnvnQZtrKiPuo4dVMO14sPa5NqzI8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.12.5.3' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-64-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

97 packages can be updated.
0 updates are security updates.


*** System restart required ***
Last login: Thu Nov 16 14:17:10 2017 from 10.12.25.189
ubuntu@vm1-dcae-bootstrap:~$ cd /opt/
ubuntu@vm1-dcae-bootstrap:/opt$ ls
app  config  dcae2_install.sh  dcae2_vm_init.sh  docker  nginx.conf
ubuntu@vm1-dcae-bootstrap:/opt$ cd config/
ubuntu@vm1-dcae-bootstrap:/opt/config$ ls
aai1_floating_ip_addr.txt    dns_forwarder.txt              openstack_password.txt
aai1_ip_addr.txt             dns_ip_addr.txt                openstack_private_network_name.txt
aai2_floating_ip_addr.txt    docker_version.txt             openstack_region.txt
aai2_ip_addr.txt             external_dns.txt               openstack_user.txt
artifacts_version.txt        flavor_id_cdap.txt             policy_floating_ip_addr.txt
centos_7_image.txt           flavor_id.txt                  policy_ip_addr.txt
cloud_env.txt                gerrit_branch.txt              priv_key
dcae_c_floating_ip_addr.txt  key_name.txt                   pub_key.txt
dcae_domain.txt              keystone_url.txt               public_net_id.txt
dcae_float_ip.txt            mac_addr.txt                   public_net_name.txt
dcae_ip_addr.txt             mr_ip_addr.txt                 rand_str.txt
dcae_zone.txt                mrouter_floating_ip_addr.txt   remote_repo.txt
dnsaas_config_enabled.txt    nexus_docker_repo.txt          sdc_floating_ip_addr.txt
dnsaas_keystone_url.txt      nexus_password.txt             sdc_ip_addr.txt
dnsaas_password.txt          nexus_repo.txt                 security_group.txt
dnsaas_region.txt            nexus_username.txt             tenant_id.txt
dnsaas_tenant_name.txt       openo_floating_ip_addr.txt     tenant_name.txt
dnsaas_username.txt          openo_ip_addr.txt              ubuntu_1604_image.txt
dns_floating_ip_addr.txt     openstack_keystone_url.txt
ubuntu@vm1-dcae-bootstrap:/opt/config$
```

Let's take a look at the sdc_floating_ip_addr.txt – that is how all the associations are being made (more sdc_floating_ip_addr.txt -> 10.12.5.12).



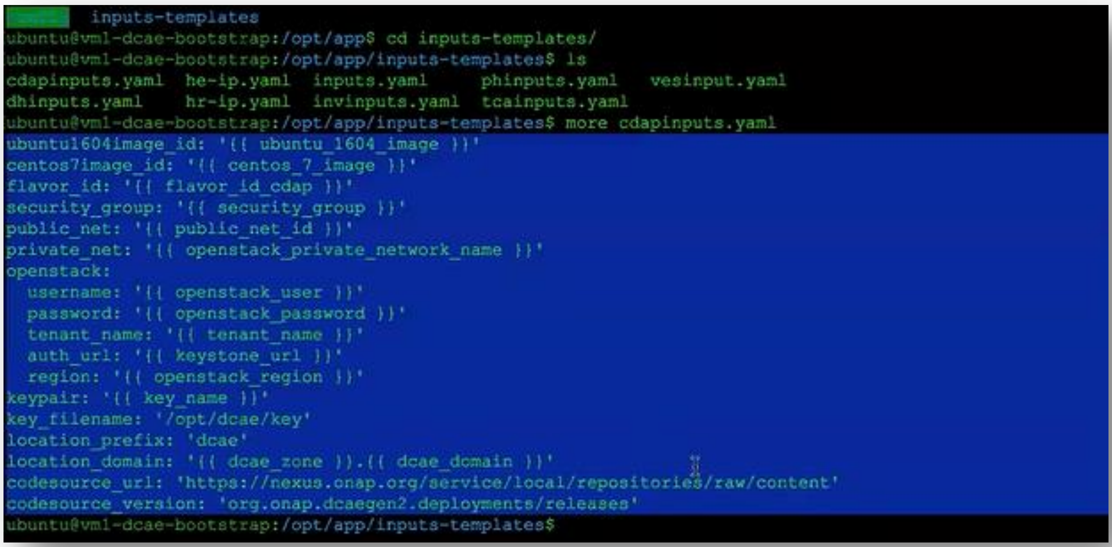
```
ubuntu@vm1-dcae-bootstrap:/opt/config$ sdc_floating_ip_addr.txt
sdc_floating_ip_addr.txt: command not found
ubuntu@vm1-dcae-bootstrap:/opt/config$ more sdc_floating_ip_addr.txt
10.12.5.12
ubuntu@vm1-dcae-bootstrap:/opt/config$ 
```

And when we are downloading the input templates (/opt/app/input-templates more cdapinputs.yaml).



```
ubuntu@vm1-dcae-bootstrap:/opt/config$ cd ..
ubuntu@vm1-dcae-bootstrap:/opt$ ls
app  config  dcae2_install.sh  dcae2_vm_init.sh  docker  nginx.conf
ubuntu@vm1-dcae-bootstrap:/opt$ cd app/
ubuntu@vm1-dcae-bootstrap:/opt/app$ ls
config  inputs-templates
ubuntu@vm1-dcae-bootstrap:/opt/app$ cd inputs-templates/
ubuntu@vm1-dcae-bootstrap:/opt/app/inputs-templates$ ls
cdapinputs.yaml  he-ip.yaml  inputs.yaml     phinputs.yaml   vesinput.yaml
dhinputs.yaml    hr-ip.yaml  invinputs.yaml  tcainputs.yaml
ubuntu@vm1-dcae-bootstrap:/opt/app/inputs-templates$ more cdapinputs.yaml
```

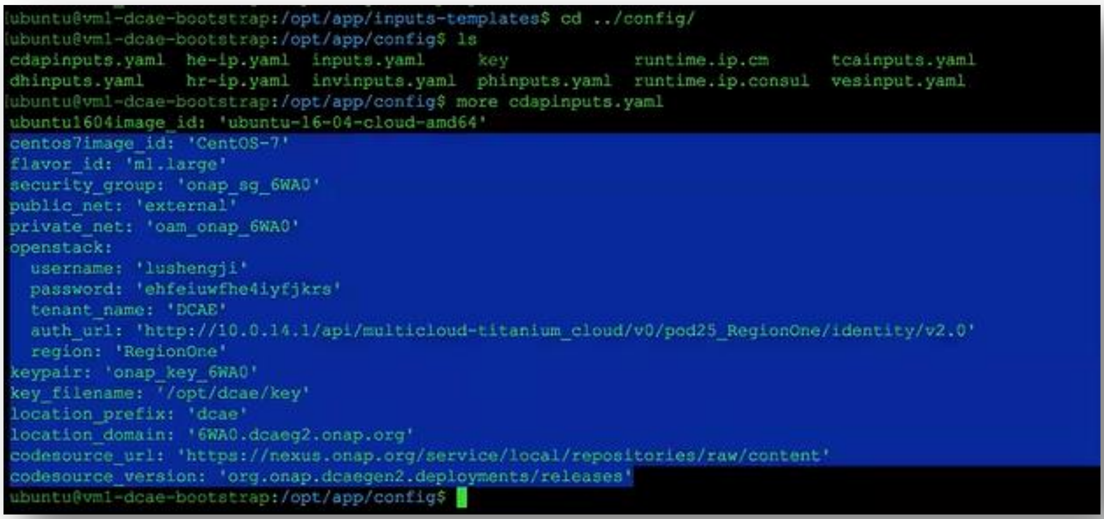
That is an original form of the inputs file (ubuntu1604_id: '{{Ubuntu_1604_image }}')



```
config  inputs-templates
ubuntu@vm1-dcae-bootstrap:/opt/app$ cd inputs-templates/
ubuntu@vm1-dcae-bootstrap:/opt/app/inputs-templates$ ls
cdapinputs.yaml  he-ip.yaml  inputs.yaml     phinputs.yaml   vesinput.yaml
dhinputs.yaml    hr-ip.yaml  invinputs.yaml  tcainputs.yaml
ubuntu@vm1-dcae-bootstrap:/opt/app/inputs-templates$ more cdapinputs.yaml
ubuntu1604image_id: '{{ ubuntu_1604_image }}'
centos7image_id: '{{ centos_7_image }}'
flavor_id: '{{ flavor_id_cdap }}'
security_group: '{{ security_group }}'
public_net: '{{ public_net_id }}'
private_net: '{{ openstack_private_network_name }}'
openstack:
  username: '{{ openstack_user }}'
  password: '{{ openstack_password }}'
  tenant_name: '{{ tenant_name }}'
  auth_url: '{{ keystone_url }}'
  region: '{{ openstack_region }}'
keypair: '{{ key_name }}'
key_filename: '/opt/dcae/key'
location_prefix: 'dcae'
location_domain: '{{ dcae_zone }}.{{ dcae_domain }}'
codesource_url: 'https://nexus.onap.org/service/local/repositories/raw/content'
codesource_version: 'org.onap.dcaegen2.deployments/releases'
ubuntu@vm1-dcae-bootstrap:/opt/app/inputs-templates$
```

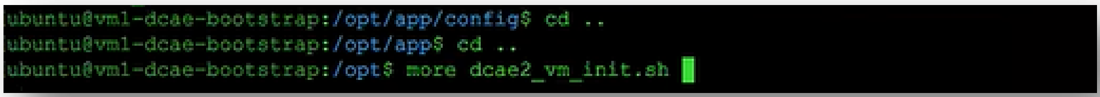
you can see all those things are changed into templates and for example how these templates are resolved is for example: template centos_7_image and then expansion script will look in this config directory finding a file called centos_7_image without extension and will replace what is in this file plugged into this jinja to templates and the results are copied into the different directory (cd ../config/ more cdapinputs.yaml).

Everything is detemplatized using the values provided from the config directory.



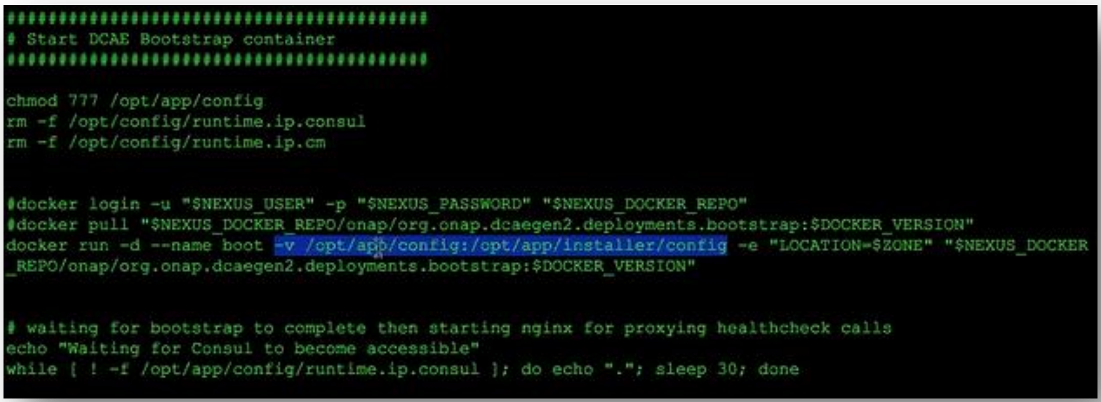
```
ubuntu@vm1-dcae-bootstrap:/opt/app/inputs-templates$ cd ../config/
ubuntu@vm1-dcae-bootstrap:/opt/app/config$ ls
cdapinputs.yaml  he-ip.yaml  inputs.yaml     key            runtime.ip.cm      tcainputs.yaml
dhinputs.yaml    hr-ip.yaml  invinputs.yaml  phinputs.yaml  runtime.ip.consul  vesinput.yaml
ubuntu@vm1-dcae-bootstrap:/opt/app/config$ more cdapinputs.yaml
ubuntu1604image_id: 'ubuntu-16-04-cloud-amd64'
centos7image_id: 'CentOS-7'
flavor_id: 'm1.large'
security_group: 'onap_sg_6WA0'
public_net: 'external'
private_net: 'oam_onap_6WA0'
openstack:
  username: 'lushengji'
  password: 'ehfeiuwfhe4iyfjkrs'
  tenant_name: 'DCAE'
  auth_url: 'http://10.0.14.1/api/multicloud-titanium_cloud/v0/pod25_RegionOne/identity/v2.0'
  region: 'RegionOne'
keypair: 'onap_key_6WA0'
key_filename: '/opt/dcae/key'
location_prefix: 'dcae'
location_domain: '6WA0.dcaeg2.onap.org'
codesource_url: 'https://nexus.onap.org/service/local/repositories/raw/content'
codesource_version: 'org.onap.dcaegen2.deployments/releases'
ubuntu@vm1-dcae-bootstrap:/opt/app/config$
```

So if you want to change when running the bootstrap docker container this directly is mapped from disc directory on the host VM into the docker.



```
ubuntu@vm1-dcae-bootstrap:/opt/app/config$ cd ..
ubuntu@vm1-dcae-bootstrap:/opt/app$ cd ..
ubuntu@vm1-dcae-bootstrap:/opt$ more dcae2_vm_init.sh
```

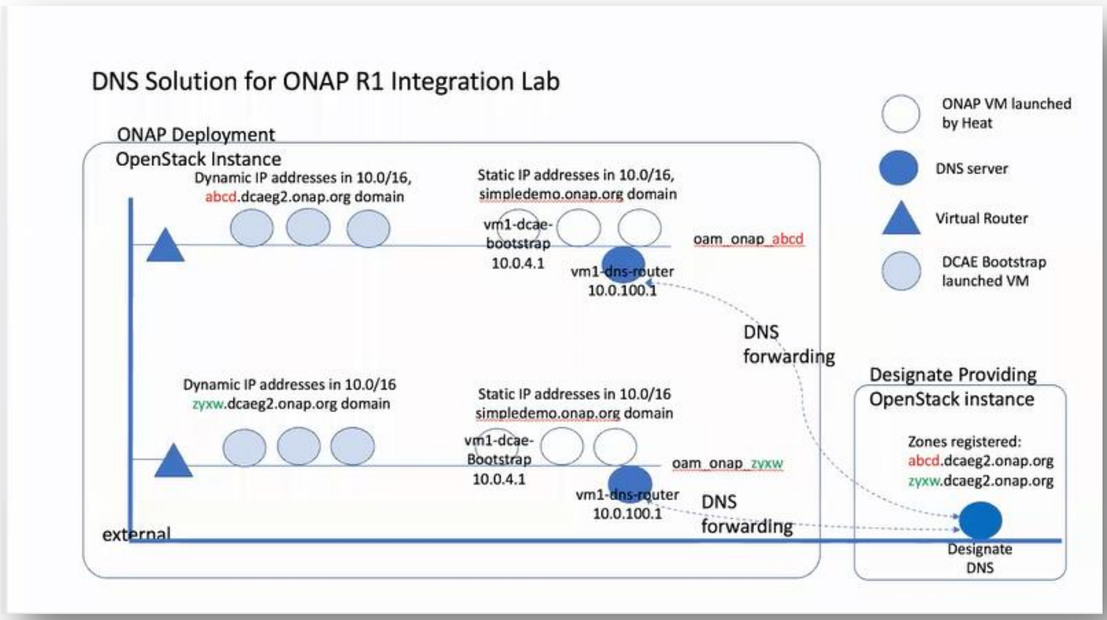
So you can see at the end when we are running it, we map (docker run –d –name boot –v /opt/app/config:/opt/app/installer/config) the volumes local disc opt/app/config into what is inside of the docker container opt/app/installer/config.



```
###########################################
# Start DCAE Bootstrap container
###########################################

chmod 777 /opt/app/config
rm -f /opt/config/runtime.ip.consul
rm -f /opt/config/runtime.ip.cm


#docker login -u "$NEXUS_USER" -p "$NEXUS_PASSWORD" "$NEXUS_DOCKER_REPO"
#docker pull "$NEXUS_DOCKER_REPO/onap/org.onap.dcaegen2.deployments.bootstrap:$DOCKER_VERSION"
docker run -d --name boot -v /opt/app/config:/opt/app/installer/config -e "LOCATION=$ZONE" "$NEXUS_DOCKER
_REPO/onap/org.onap.dcaegen2.deployments.bootstrap:$DOCKER_VERSION"


# waiting for bootstrap to complete then starting nginx for proxying healthcheck calls
echo "Waiting for Consul to become accessible"
while [ ! -f /opt/app/config/runtime.ip.consul ]; do echo "."; sleep 30; done
```

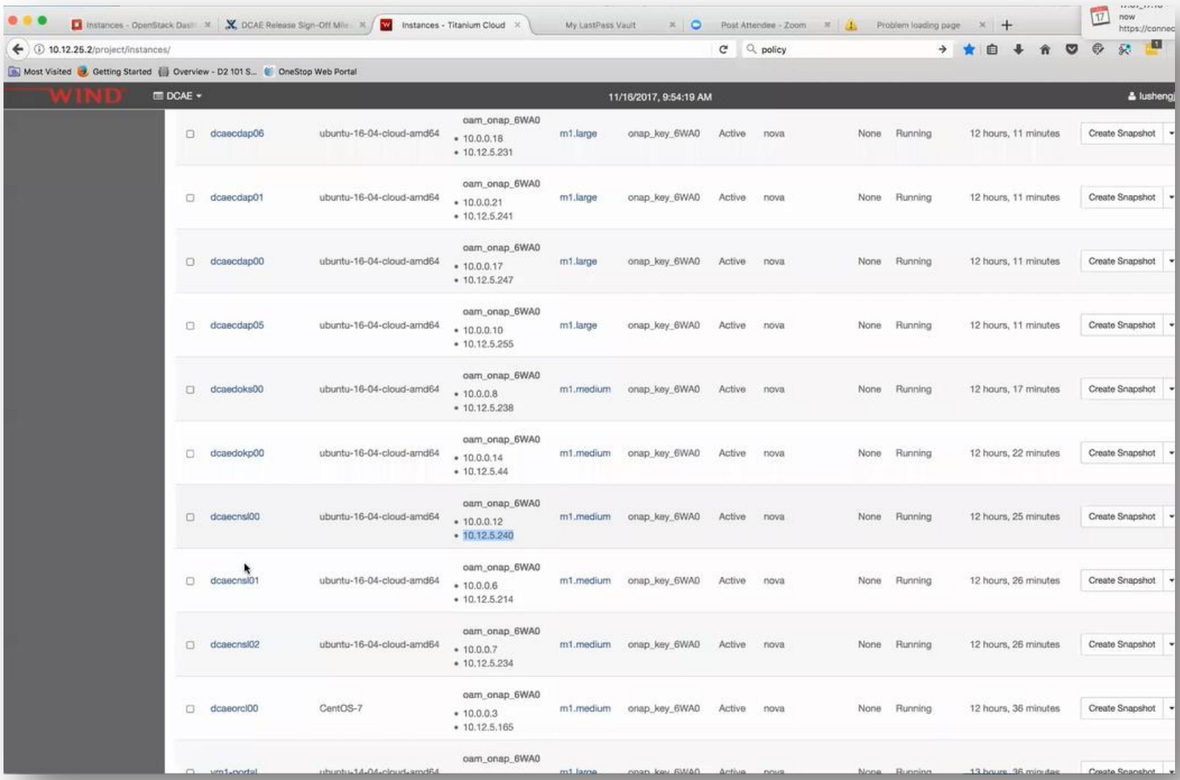
So for example when it is running it is waiting for AAI to become ready if you want to customize you can login to the bootstrap VM and customize the files inside the app config. In the input files you can place things over there and when the docker container is ready to run it is gonna map into the docker container and we see all modified the values than it will run the stuff like you specify.

DNS as you know because in DCAE we use DHCP assigned flow the dynamic IP addresses which is different from what the heat solution uses – that uses static IP addresses on the intra ONAP communication there used fixed prealocated IP address. For DCAE we use in the production environment, we started with DHCP assigned dynamic IP addresses and using host names or DNS based solution for different VMS to find each other. DNS is time and scale proven technique for this kind of problem if you deal with really large deployment crossing different regions, different data centers multiple of zones DNS is a reasonable solution. When we take that and move to ONAP – here there is a gap how to dynamically update IP address to DNS name association, that means designate – there is not designate inside the pod25 we worked with Windriver trying to find out the solution but because the pod25 environment is there so called carrier grade cloud solution – it has a cluster of controllers and again you have a leader election relation those kind of things it becomes very difficult to add the component that is originally not there, so they were saying they have to bring down whole system to perform such a task it would be very interruption for the integration, so the final solution is it is a little bit really like this I had to explain so many times actually to a picture here.


DNS Solution for ONAP R1 Integration Lab
ONAP Deployment
OpenStack Instance
Dynamic IP addresses in 10.0/16,
abcd.dcaeg2.onap.org domain
Static IP addresses in 10.0/16,
simpledemo.onap.org domain
vm1-dcae-bootstrap
10.0.4.1
vm1-dns-router
10.0.100.1
oam_onap_abcd
DNS forwarding
Dynamic IP addresses in 10.0/16
zyxw.dcaeg2.onap.org domain
Static IP addresses in 10.0/16
simpledemo.onap.org domain
vm1-dcae-Bootstrap
10.0.4.1
vm1-dns-router
10.0.100.1
oam_onap_zyxw
DNS forwarding
external
ONAP VM launched by Heat
DNS server
Virtual Router
DCAE Bootstrap launched VM
Designate Providing OpenStack instance
Zones registered:
abcd.dcaeg2.onap.org
zyxw.dcaeg2.onap.org
Designate DNS

So we have setup a separate Openstack instance. It is new, They have to had designate so they put designated inside. This is a larger pod25. Through heat , the heat first bring-up the ONAP VMs including the bootstrap VM and they are all put onto the private network called oam_onap_abcd is a run id and 10.0 addresses are all fixed and one of them these nodes is the DNS router (vm1-dns-router 10.0.100.1). So all those VMs they use this DNS router as their default DNS resolution server – this guy knows the static mapping for example 100.1 is DNS.simpledemo.onap.org or something like that. And DCAE's VMs they are span-up and they are connected to the same private network and they use the same DNS server as a default resolution server. What we configured is on this DNS resolution server added a forwarder entry that is if any host names or any domains DNS server does not about, it is gonna ask another guy it is a DNS server that is running behind the DNS designate service inside of this separate Openstack and this guy is forwarder's is google's 8.8.8.8 so when the communication is from abcd.dcagen2.onap.org (dynamic addresses) to here (static addresses) domain so we use a simpledemo.onap.org domain so DNS will be request going to this guy and he knows it is on his private domain, so he will return the private IP address and the communication is done like that. IF VM from static IP addresses wants to refer to a VMs with dynamic ip addresses, for example he wants to know how to that host name, so that is name resolution will go to this guy (vm1-dns-router 10.0.100.1) and abcd.dcaegen2.onap.org is a separate domain which I don't know, then vm1-dns-router 10.0.100.1 will gonna forward this request to Designate DNS – this guy knows because we saw in the script it registered this domain with this DNS – he would return the IP address of abcd.dcaegen2.onap.org where the VM is running to the requester. So this way both groups they can communicate with each other using host names – that is the DNS solution at least for the integration lab. And it works because one of the design goal they setup the heat solution is you can have multiple installations of ONAP within same environment, so this does work because different

DNS groups they do register different zones with the designate. Let me just quickly show you how you can check after you started DCAE how to check the status,



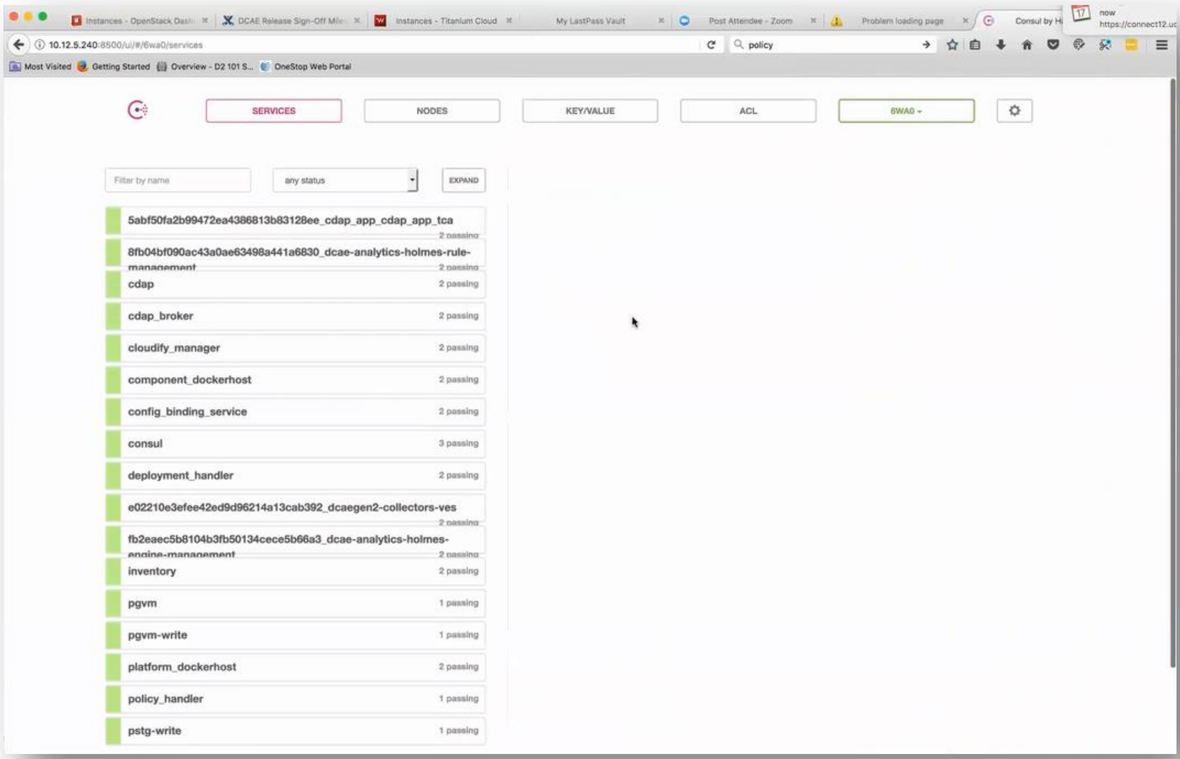
10.12.25.2/project/instances/

DCAE

11/16/2017, 9:54:19 AM

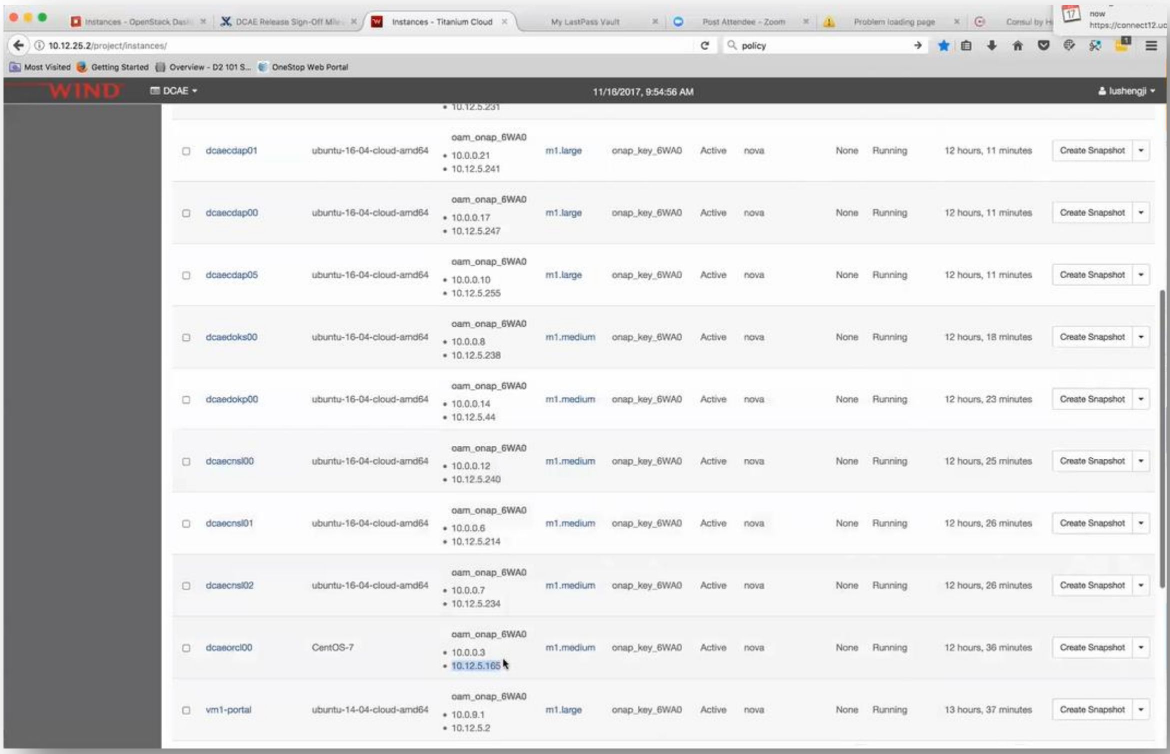
☐	dcaecdap06	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.18 • 10.12.5.231	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 11 minutes	Create Snapshot
☐	dcaecdap01	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.21 • 10.12.5.241	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 11 minutes	Create Snapshot
☐	dcaecdap00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.17 • 10.12.5.247	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 11 minutes	Create Snapshot
☐	dcaecdap05	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.10 • 10.12.5.255	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 11 minutes	Create Snapshot
☐	dcaedoks00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.8 • 10.12.5.238	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 17 minutes	Create Snapshot
☐	dcaedokp00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.14 • 10.12.5.44	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 22 minutes	Create Snapshot
☐	dcaecnsl00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.12 • 10.12.5.240	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 25 minutes	Create Snapshot
☐	dcaecnsl01	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.6 • 10.12.5.214	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 26 minutes	Create Snapshot
☐	dcaecnsl02	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.7 • 10.12.5.234	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 26 minutes	Create Snapshot
☐	dcaeorcl00	CentOS-7	oam_onap_6WA0 • 10.0.0.3 • 10.12.5.165	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 36 minutes	Create Snapshot
☐	vm1-portal	ubuntu-14-04-cloud-amd64	oam_onap_6WA0	m1.large	onap_key_6WA0	Active	nova	None	Running	13 hours, 36 minutes	Create Snapshot

if you go to any use of the consoles IP, - in dcaecns00 IP address 10.12.5.240 and go in your web browser to 10.12.5.240:8800, you can see the status of all DCAE components. Here is the TCA, Holmes, VES, all those things.

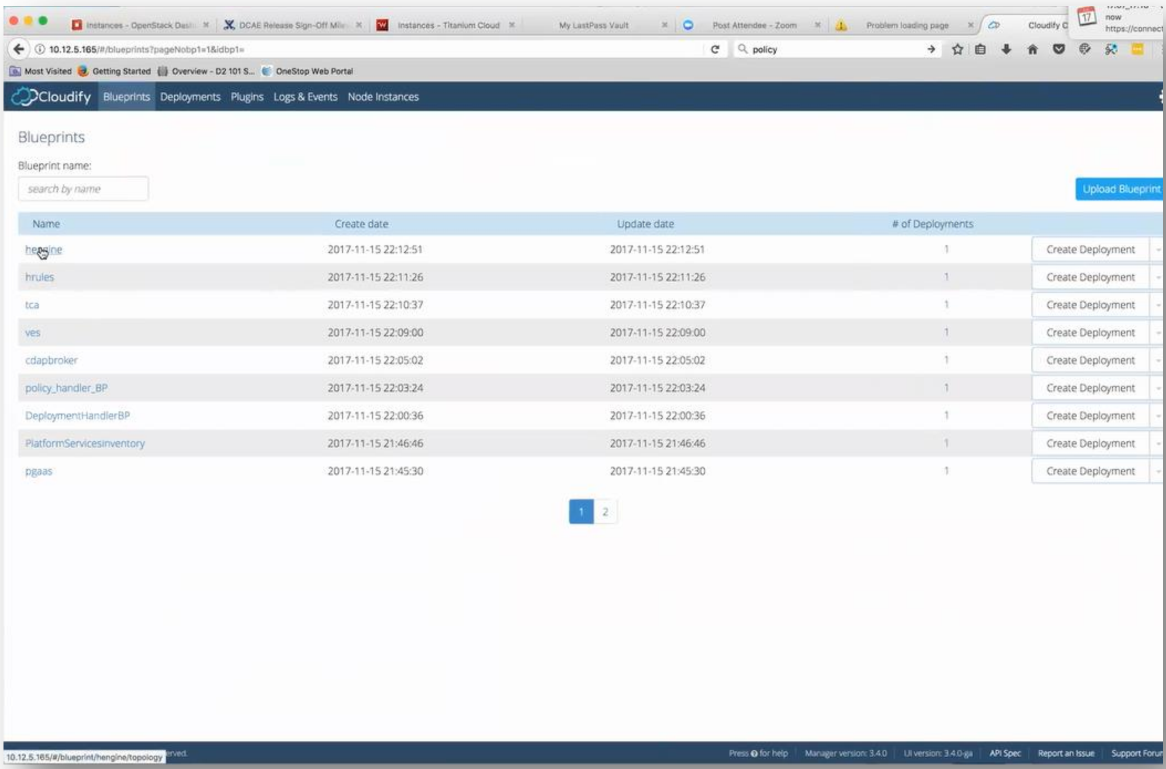


5abf50fa2b99472ea4386813b83128ee_cdap_app_cdap_app_tca	2 passing
8fb04bf090ac43a0ae63498a441a6830_dcae-analytics-holmes-rule-management	2 passing
cdap	2 passing
cdap_broker	2 passing
cloudify_manager	2 passing
component_dockerhost	2 passing
config_binding_service	2 passing
consul	3 passing
deployment_handler	2 passing
e02210e3efee42ed9d96214a13cab392_dcaegen2-collectors-ves	2 passing
fb2eaec5b8104b3fb50134cece5b66a3_dcae-analytics-holmes-engine-management	2 passing
inventory	2 passing
pgvm	1 passing
pgvm-write	1 passing
platform_dockerhost	2 passing
policy_handler	1 passing
pstg-write	1 passing

That is one component. You can also directly access qualify manager's UI: 10.12.5.165 to deploy an additional blueprints for example these are all deployed blueprints through bootstrap script.



dcaecdap01	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.21 • 10.12.5.241	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 11 minutes	Create Snapshot
dcaecdap00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.17 • 10.12.5.247	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 11 minutes	Create Snapshot
dcaecdap05	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.10 • 10.12.5.255	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 11 minutes	Create Snapshot
dcaedoks00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.8 • 10.12.5.238	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 18 minutes	Create Snapshot
dcaedokp00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.14 • 10.12.5.44	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 23 minutes	Create Snapshot
dcaecnsl00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.12 • 10.12.5.240	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 25 minutes	Create Snapshot
dcaecnsl01	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.6 • 10.12.5.214	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 26 minutes	Create Snapshot
dcaecnsl02	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.7 • 10.12.5.234	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 26 minutes	Create Snapshot
dcaeorcl00	CentOS-7	oam_onap_6WA0 • 10.0.0.3 • 10.12.5.165	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 36 minutes	Create Snapshot
vm1-portal	ubuntu-14-04-cloud-amd64	oam_onap_6WA0 • 10.0.9.1 • 10.12.5.2	m1.large	onap_key_6WA0	Active	nova	None	Running	13 hours, 37 minutes	Create Snapshot

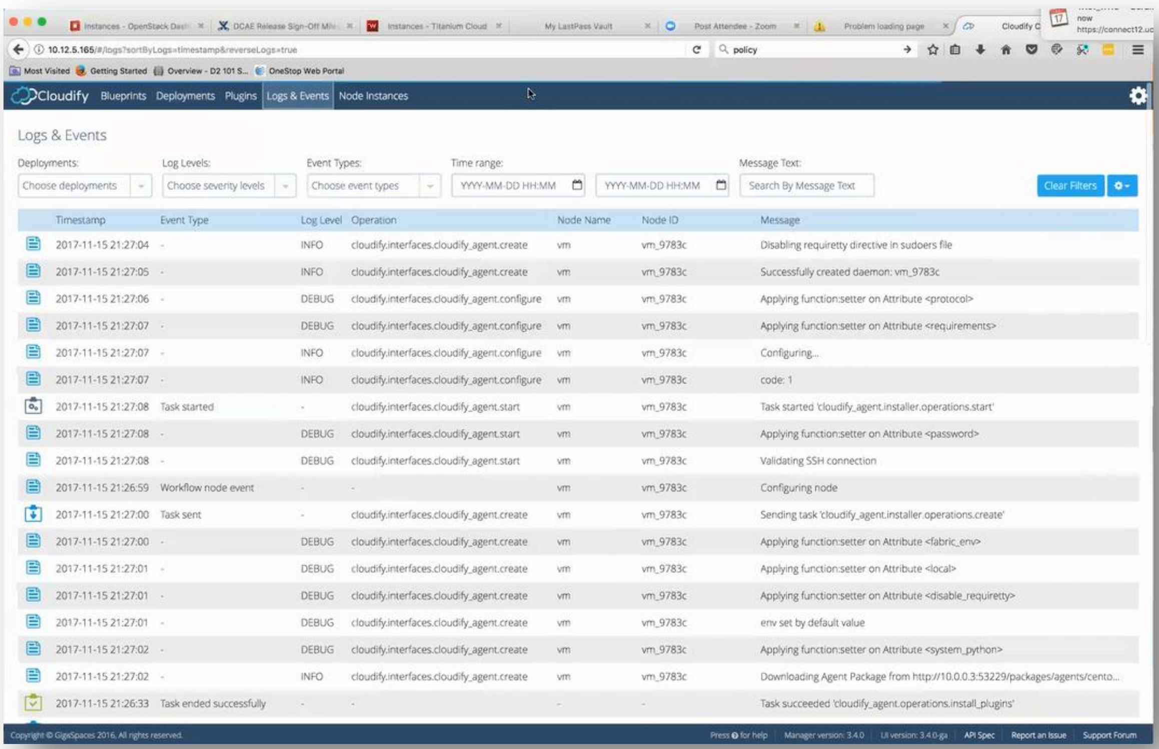


Blueprints

Blueprint name:

Name	Create date	Update date	# of Deployments	
hengine	2017-11-15 22:12:51	2017-11-15 22:12:51	1	Create Deployment
hrules	2017-11-15 22:11:26	2017-11-15 22:11:26	1	Create Deployment
tca	2017-11-15 22:10:37	2017-11-15 22:10:37	1	Create Deployment
ves	2017-11-15 22:09:00	2017-11-15 22:09:00	1	Create Deployment
cdapbroker	2017-11-15 22:05:02	2017-11-15 22:05:02	1	Create Deployment
policy_handler_BP	2017-11-15 22:03:24	2017-11-15 22:03:24	1	Create Deployment
DeploymentHandlerBP	2017-11-15 22:00:36	2017-11-15 22:00:36	1	Create Deployment
PlatformServicesInventory	2017-11-15 21:46:46	2017-11-15 21:46:46	1	Create Deployment
pgaas	2017-11-15 21:45:30	2017-11-15 21:45:30	1	Create Deployment

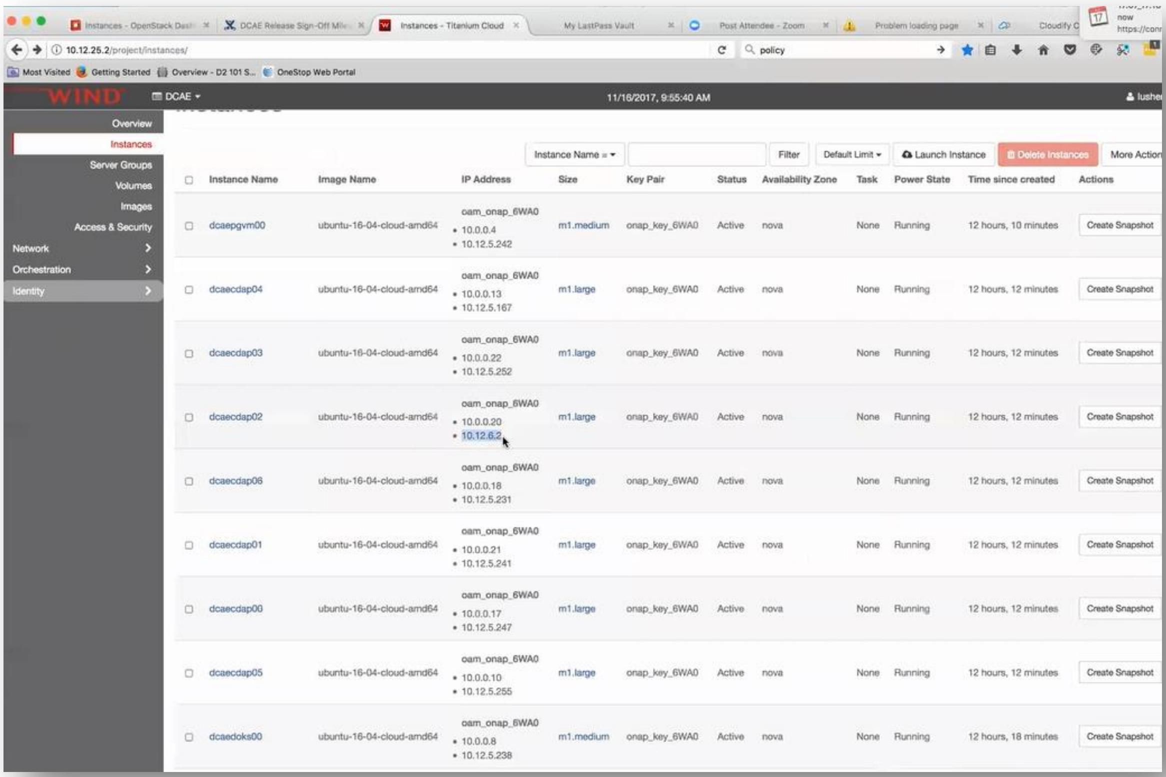
And it provides a really nice logging , you can check it up here.



Logs & Events

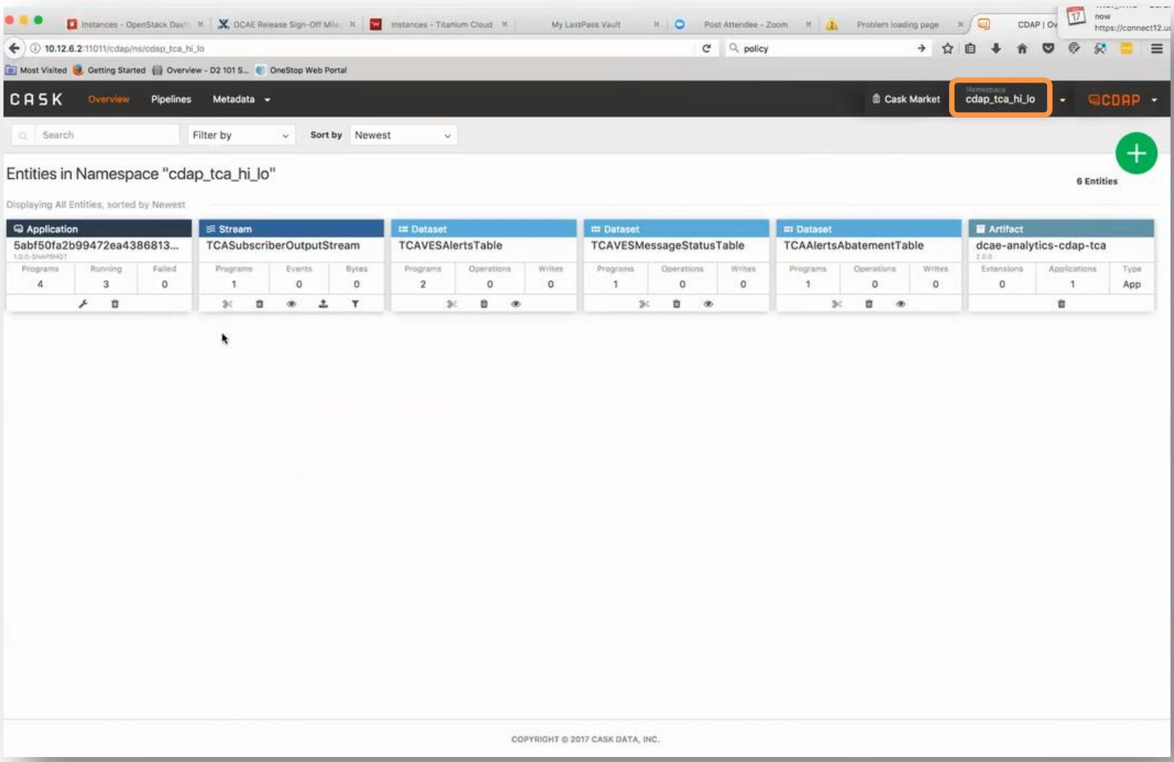
	Timestamp	Event Type	Log Level	Operation	Node Name	Node ID	Message
	2017-11-15 21:27:04	-	INFO	cloudify.interfaces.cloudify_agent.create	vm	vm_9783c	Disabling requiretty directive in sudoers file
	2017-11-15 21:27:05	-	INFO	cloudify.interfaces.cloudify_agent.create	vm	vm_9783c	Successfully created daemon: vm_9783c
	2017-11-15 21:27:06	-	DEBUG	cloudify.interfaces.cloudify_agent.configure	vm	vm_9783c	Applying function:setter on Attribute <protocol>
	2017-11-15 21:27:07	-	DEBUG	cloudify.interfaces.cloudify_agent.configure	vm	vm_9783c	Applying function:setter on Attribute <requirements>
	2017-11-15 21:27:07	-	INFO	cloudify.interfaces.cloudify_agent.configure	vm	vm_9783c	Configuring...
	2017-11-15 21:27:07	-	INFO	cloudify.interfaces.cloudify_agent.configure	vm	vm_9783c	code: 1
	2017-11-15 21:27:08	Task started	-	cloudify.interfaces.cloudify_agent.start	vm	vm_9783c	Task started 'cloudify_agent.installer.operations.start'
	2017-11-15 21:27:08	-	DEBUG	cloudify.interfaces.cloudify_agent.start	vm	vm_9783c	Applying function:setter on Attribute <password>
	2017-11-15 21:27:08	-	DEBUG	cloudify.interfaces.cloudify_agent.start	vm	vm_9783c	Validating SSH connection
	2017-11-15 21:26:59	Workflow node event	-	-	vm	vm_9783c	Configuring node
	2017-11-15 21:27:00	Task sent	-	cloudify.interfaces.cloudify_agent.create	vm	vm_9783c	Sending task 'cloudify_agent.installer.operations.create'
	2017-11-15 21:27:00	-	DEBUG	cloudify.interfaces.cloudify_agent.create	vm	vm_9783c	Applying function:setter on Attribute <fabric_env>
	2017-11-15 21:27:01	-	DEBUG	cloudify.interfaces.cloudify_agent.create	vm	vm_9783c	Applying function:setter on Attribute <local>
	2017-11-15 21:27:01	-	DEBUG	cloudify.interfaces.cloudify_agent.create	vm	vm_9783c	Applying function:setter on Attribute <disable_requiretty>
	2017-11-15 21:27:01	-	DEBUG	cloudify.interfaces.cloudify_agent.create	vm	vm_9783c	env set by default value
	2017-11-15 21:27:02	-	DEBUG	cloudify.interfaces.cloudify_agent.create	vm	vm_9783c	Applying function:setter on Attribute <system_python>
	2017-11-15 21:27:02	-	INFO	cloudify.interfaces.cloudify_agent.create	vm	vm_9783c	Downloading Agent Package from http://10.0.0.3:53229/packages/agents/cento...
	2017-11-15 21:26:33	Task ended successfully	-	-	-	-	Task succeeded 'cloudify_agent.operations.install_plugins'

For CDAP, it has a GUI, you can check , go to CDAP02 10.12.6.2:11011



Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
dcaepgvm00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.4 • 10.12.5.242	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 10 minutes	Create Snapshot
dcaecdap04	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.13 • 10.12.5.167	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 12 minutes	Create Snapshot
dcaecdap03	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.22 • 10.12.5.252	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 12 minutes	Create Snapshot
dcaecdap02	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.20 • 10.12.6.2	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 12 minutes	Create Snapshot
dcaecdap06	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.18 • 10.12.5.231	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 12 minutes	Create Snapshot
dcaecdap01	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.21 • 10.12.5.241	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 12 minutes	Create Snapshot
dcaecdap00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.17 • 10.12.5.247	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 12 minutes	Create Snapshot
dcaecdap05	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.10 • 10.12.5.255	m1.large	onap_key_6WA0	Active	nova	None	Running	12 hours, 12 minutes	Create Snapshot
dcaedoks00	ubuntu-16-04-cloud-amd64	oam_onap_6WA0 • 10.0.0.8 • 10.12.5.238	m1.medium	onap_key_6WA0	Active	nova	None	Running	12 hours, 18 minutes	Create Snapshot

– that is CDAP UI and we are not using default namescpace, you can see that there is a cdap namespace, see lots of information there.



Entities in Namespace "cdap_tca_hi_lo"

Displaying All Entities, sorted by Newest

6 Entities

Application	Stream	Dataset	Dataset	Dataset	Artifact
5abf50fa2b99472ea4386813... 1.0.0-SNAPSHOT	TCASubscriberOutputStream	TCAVESAlertsTable	TCAVESMessageStatusTable	TCAAlertsAbatementTable	dcae-analytics-cdap-tca 2.0.0
Programs 4, Running 3, Failed 0	Programs 1, Events 0, Bytes 0	Programs 2, Operations 0, Writes 0	Programs 1, Operations 0, Writes 0	Programs 1, Operations 0, Writes 0	Extensions 0, Applications 1, Type App

All these are opensource tools there are documentations in the public domain on how to use it. So that is about it, we are close to the end.

Question: If you are talking of designate here, are there plans to continue using designate here?
Well it is still under discussion, because here is the trade-off: If you are bounded to a static IP addresses and you use those, then the designate is not really needed because you don't have dynamic requirements to update the DNS records. But if you want to use dynamically assigned IP addresses then you need some way to dynamically register the host name to IP binding. That is where the designate comes in. But of course there are alternative technologies – you could stand-up a bind server and provide some API to it. We are evaluating exactly which way we want to go at this point but we know it is an experience. I guess lot of installations do not have a designate. Just from our own internal experience once you done it, it is not that hard but you know that is just from our perspective. Yes, we are evaluating what to do with that internally as well.

Question: Hi Lusheng, just quick question about the documentation because I went through the VNF requirements and in a chapter "Monitoring and management requirements" and there is mentioned about to VES Jason there is also some chapter about Google Protocol Buffer and Avro and I wonder if it should be kept because today as far as I understand there is only VES plain text collector and there is no Google Protocol Buffer or Avro supported yet.
OK do you mind if we follow this offline?, yes, OK I will write you an e-mail just to put the link to that part, ok thank you.

Chat question: is there a recording of this session? Yes, we are recording this and I am gonna put it in DCAE wiki page as well.