EECE 7290 – Software Defined Networking (Spring 2017), University of Massachusetts, Lowell

**Project** - SDN for Secure Video Streaming: CORD Based Secure Video Streaming

**Document** – Running HelloWorld service and creating new tenant service.

**Students** –

1. Aman Maldar
2. Priyanka Murthy

Date of submission – May 5, 2017

**Part 1 - Running the HelloWorld Service on CORD [1]**

This section gives the steps to run the example service in the CORD environment. The second half of the document provides the steps to create a new tenant service.

Assuming, the steps mentioned in document ‘CORD Environment Setup’ [2] are completed, we are ready to run HelloWorld service and make changes into the template. Following steps will run the example service in the production environment. Production environment already contains XOS, ONOS, OpenStack installed on it.

**Steps:**

1) ssh into the compute node created on the CloudLab.

* ssh username@ip\_address

ex- ssh [aman\_uml@128.104.222.127](mailto:aman_uml@128.104.222.127)

2) ssh into prod environment

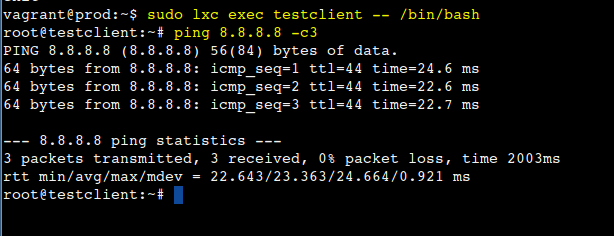
* ssh prod

3) The prod environment contains the test client, which can be used to run the services inside the CORD.

* sudo lxc exec testclient -- /bin/bash

4) Ping to see if all the services are up and running. Ping should be successful

* ping 8.8.8.8



5) Exit the testclient

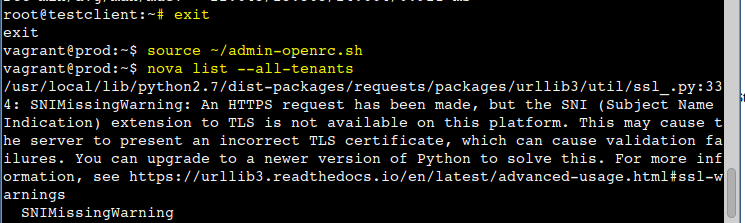
* exit

6) Provide the username to access the services.

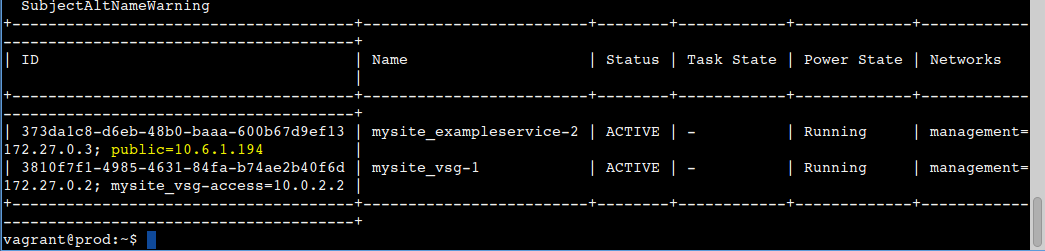
* source ~/admin-openrc.sh

7) See the list of all the running services.

* nova list --all-tenants



See the public IP address as shown below.

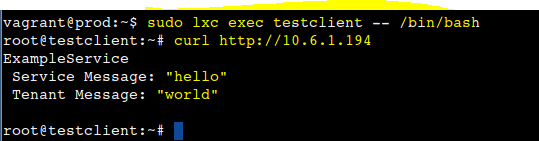


8) Enter the testclient again.

* sudo lxc exec testclient -- /bin/bash

9) Access the service by using curl command. You will see the results printed.

* curl http://10.6.1.194

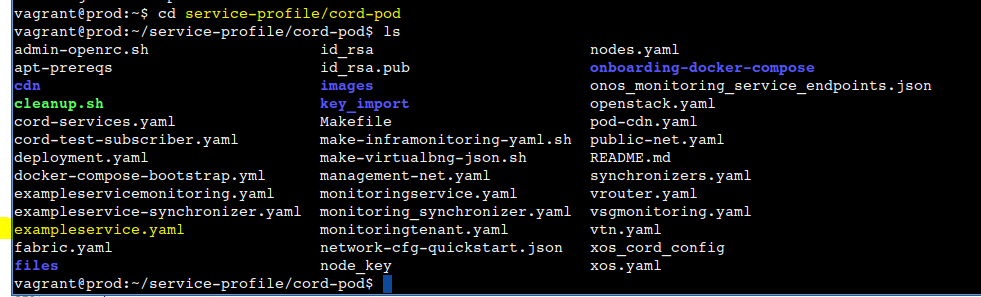


This is it! We can see the example service printing the message HelloWorld.

**Part 2 - Making changes to the HelloWorld Service.**

Follow the steps 1,2 from above.

Navigate to the folder /service-profile/cord-pod and see the files in the folder



We have to make changes in the file exampleservice.yaml

Open the file using editor and see the changes made at the end of file

service#exampleservice:

type: tosca.nodes.ExampleService

requirements:

- management:

node: management

relationship: tosca.relationships.UsesNetwork

properties:

view\_url: /admin/exampleservice/exampleservice/$id$/

kind: exampleservice

public\_key: { get\_artifact: [ SELF, pubkey, LOCAL\_FILE] }

private\_key\_fn: /opt/xos/services/exampleservice/keys/exampleservice\_rsa

service\_message: **Hello**

artifacts:

pubkey: /opt/xos/services/exampleservice/keys/exampleservice\_rsa.pub

tenant#exampletenant1:

type: tosca.nodes.ExampleTenant

properties:

tenant\_message: w**orld**

requirements:

- tenant:

node: service#exampleservice

relationship: tosca.relationships.TenantOfService

- dependency:

node: mysite\_exampleservice

relationship: tosca.relationships.DependsOn

tenant#exampletenant2:

type: tosca.nodes.ExampleTenant

properties:

tenant\_message: **universe**

requirements:

- tenant:

node: service#exampleservice

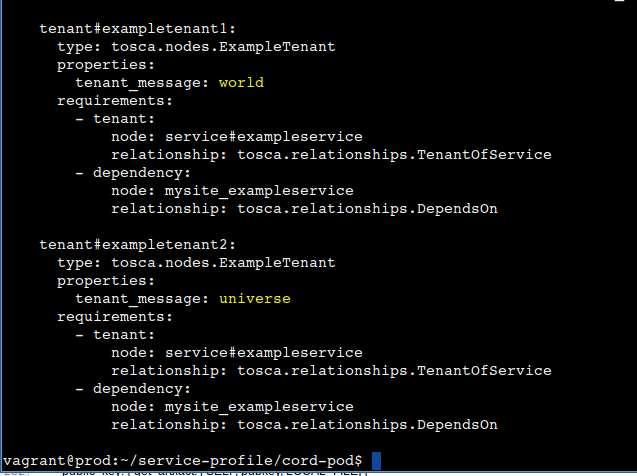
relationship: tosca.relationships.TenantOfService

- dependency:

node: mysite\_exampleservice

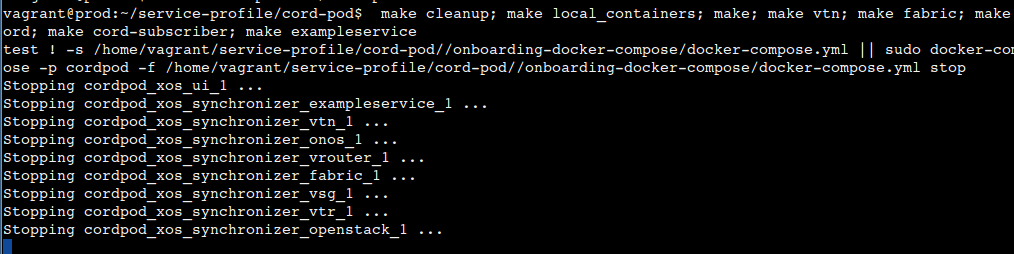
relationship: tosca.relationships.DependsOn

Observe the changes made



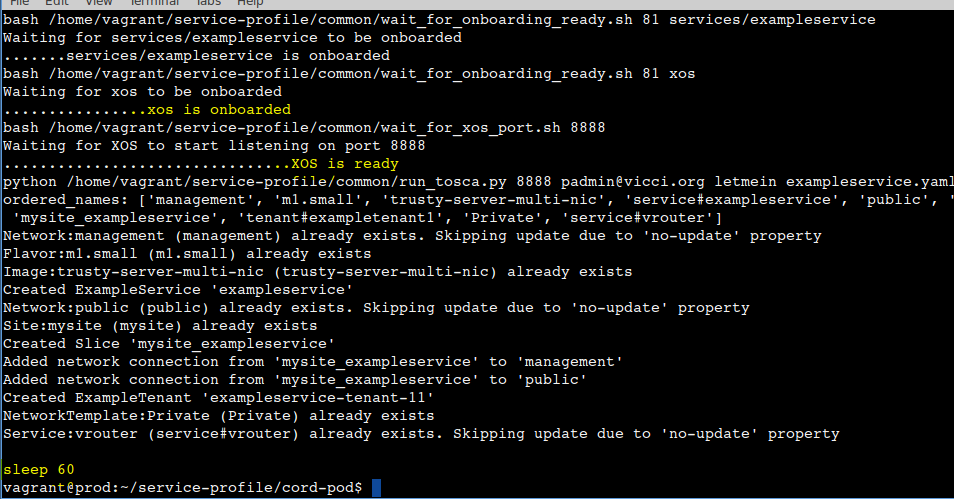
Once the above changes are made we have to run the development loop again. This takes approximately 20 minutes.

* make cleanup; make local\_containers; make; make vtn; make fabric; make cord; make cord-subscriber; make exampleservice

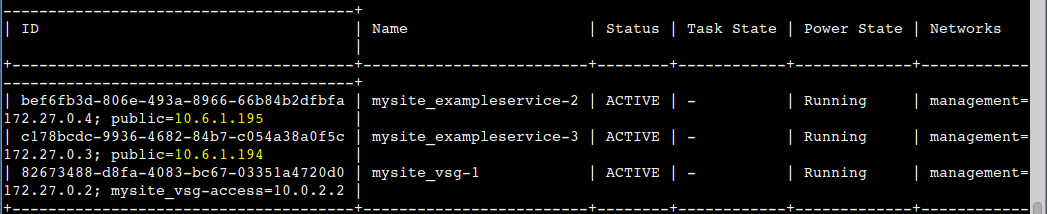


We are creating new tenant service. Two tenant s are prints 2 different messages. Parent service always runs first to print “Hello”. Two tenant services prints “world” and “universe” respectively.

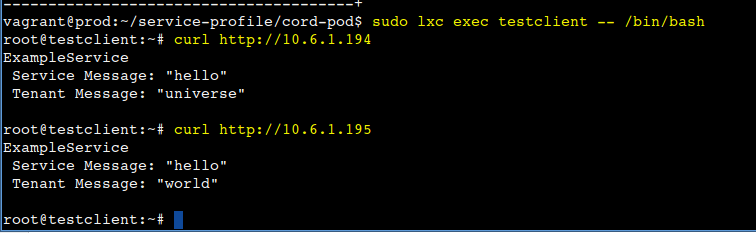
See the result of running the development loop.



Once the development loop is completed, again see the list of all the services. We can see newly created service



Do the curl again to see the results.



This is it; we have created new tenant service.

References-

[1] Running hello world - example service

<https://github.com/opencord/exampleservice/tree/master/xos>

[2] Project Documentation

<https://github.com/amanmaldar/EECE7290_Project>