# Linux Container Demo for Azure Service Fabric

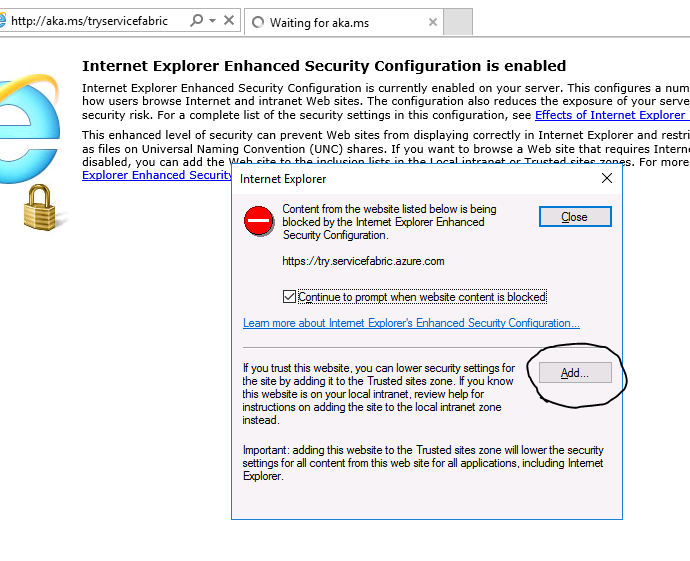
**Prerequisites for development machine**

**(Note: No need to perform these prerequisites steps for lab machines, please proceed for the lab)**

1. Get a physical machine or Azure VM with "Ubuntu 16.04 LTS" for your development machine.
2. Install the Service Fabric SDK (version 5.4.x and above)
3. Azure CLI 2.0

**Lab**

* Login to Windows Lab machines with user – msready2017, password – Msready2017lab.
* Open the Linux Lab manual present on Desktop. This will help you in copying and pasting the commands, instead of typing them (Make sure you use the **Linux** User manual and not Windows one).
* Click on the MobaXterm present on Desktop. Double click on the session for 192.168.1.11.
* Open the Internet Explorer and browse to <http://aka.ms/tryservicefabric>.
* If you get the security warning, click on Add -> Add and close.



* Login with Github or Facebook account and Join the Linux cluster.



* Open the Service Fabric Explorer by clicking on the link provided. You might be prompted for the security warning again. Please use same Add method mentioned above.

**Exercise 1: Deploy the container application**

In this exercise, we will deploy the container application to the service fabric cluster.

* Go back to MobaXterm and connect to the cluster, using the Azure cli –

*sfctl cluster select --endpoint http://<Cluster FQDN>:19080*

* Copy the package, register it and create application –

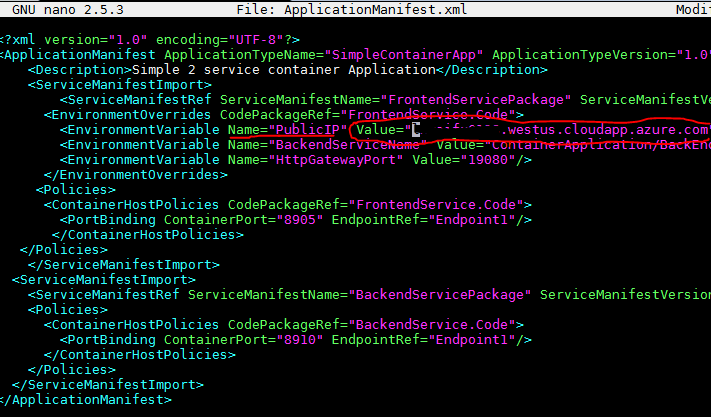
Make sure to change the FrontEndService Public IP to point to your cluster in ApplicationManifest.xml. (use nano editor).

*cd ~/LabLinux*

*cd SimpleContainerAppL*

*nano ApplicationManifest.xml*

Even though it mentions “Public IP”, you need to add the FQDN and not the IP per se. And do not add “http” or port number.



Use Ctrl+X to save and exit the nano editor.

*cd ..*

*sfctl application upload --path SimpleContainerAppL*

*sfctl application provision --application-type-build-path SimpleContainerAppL*

*sfctl application create --app-name fabric:/ContainerApplication --app-type SimpleContainerApp --app-version 1.0*

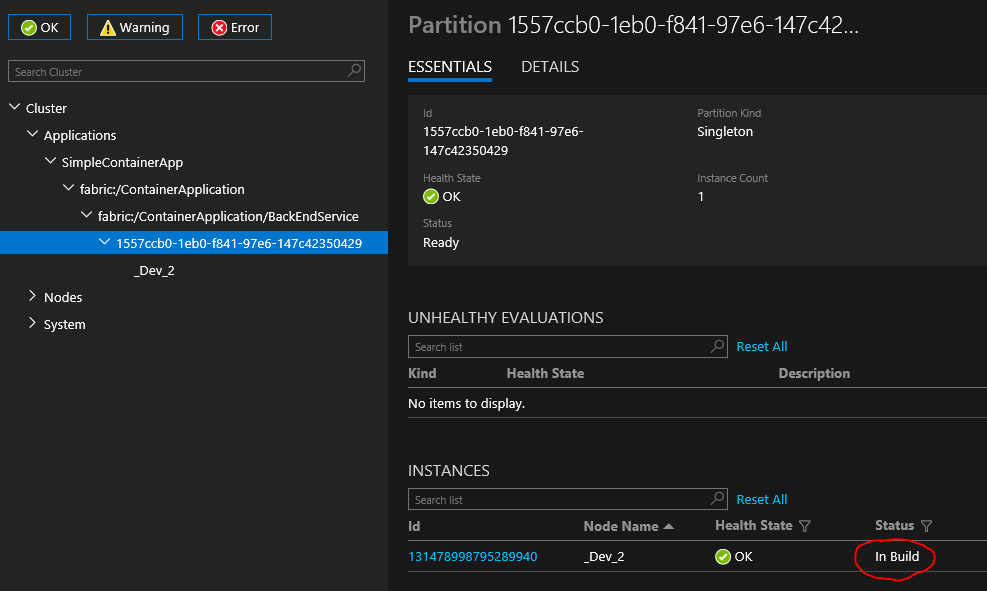
Note the version number is 1.0 here.

* Create the services (this can be created using Explorer or using the command below)

*sfctl service create --app-id ContainerApplication --name fabric:/ContainerApplication/BackEndService --service-type StatelessBackendService --stateless --instance-count 1 --singleton-scheme*

*sfctl service create --app-id ContainerApplication --name fabric:/ContainerApplication/FrontEndService --service-type StatelessFrontendService --stateless --instance-count 1 --singleton-scheme*

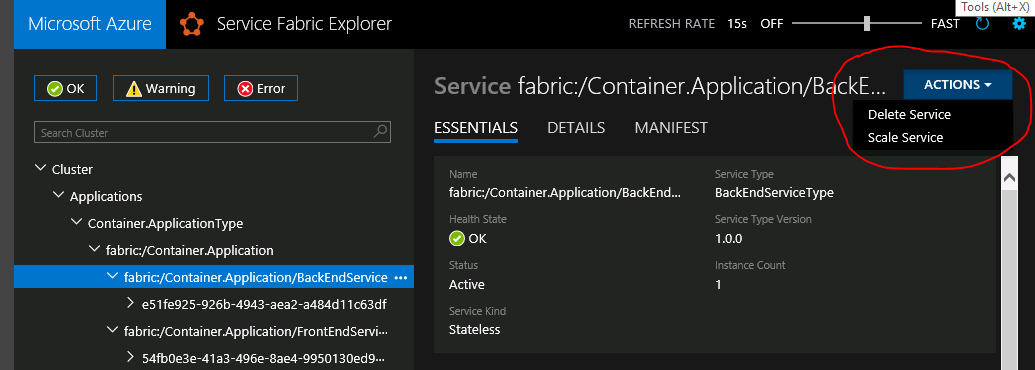
You would observe that the services show “In Build” status for some time.



Browse to application site – http://<Cluster FQDN>:8080

**Exercise 2: Scaling the application**

* Switch and login to Windows Machine. Browse to cluster endpoint.
* Scale up the number of instances of the Backend service using the Service Fabric Explorer. Click on the service within the application. On the right-hand side, you should see the “Actions” button. Click on Actions button and select “Scale service”. Change the number to 3 instances. Observe the number of instances in SFX.

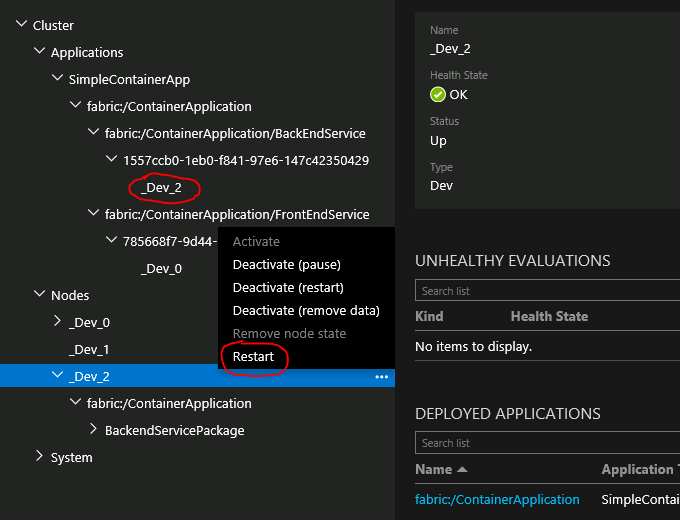


* Test by opening the IE in in-private mode and observing that the backend node showed are different with refresh.
* Using same method above scale down the number of instances to 2.

**Exercise 3: Failover to another node**

In this exercise, we will learn how to failover of node works in case node is down.

* Browse to service fabric explorer.
* Expand the nodes tree on left side.
* Click on the “…” for one of the node where your backend service is running.
* Select restart



* Observe how the backend service fails over from one node to another while the application is still up and running in the browser.

**Exercise 4: Rolling upgrade for the application**

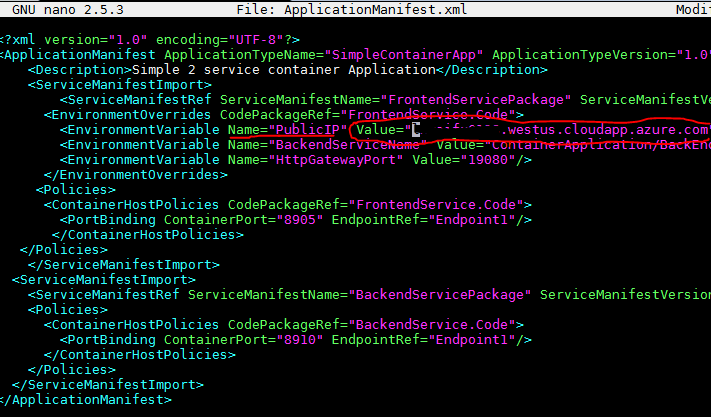
In this exercise, we will learn how to upgrade your application to newer version.

Make sure to change the FrontEndService Public IP to point to your cluster FQDN in ApplicationManifest.xml. (use nano editor)

*cd SimpleContainerAppLV2*

*nano ApplicationManifest.xml*

Even though it mentions “Public IP”, you need to add the FQDN and not the IP per se.



Use Ctrl+X to save and exit the nano editor.

*cd ..*

* Deploy the version 2 application by copying the package in Imagestore –

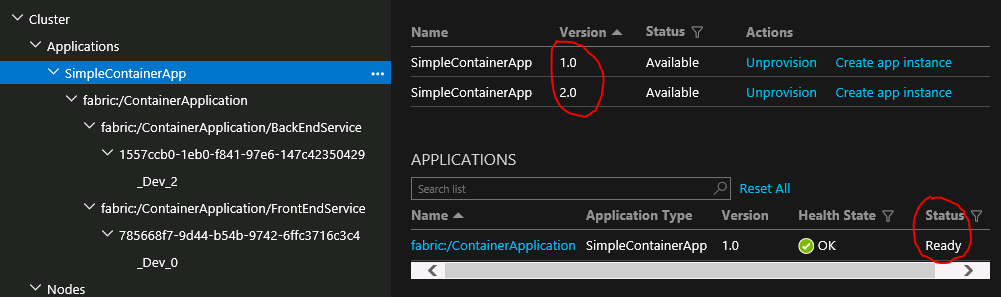
*sfctl application upload --path SimpleContainerAppLV2*

*sfctl application provision --application-type-build-path SimpleContainerAppLV2*

* Start Upgrade

*sfctl application upgrade --app-id fabric:/ContainerApplication --app-version 2.0 --parameters ‘{}’ --mode Monitored*

**\*\*Note\*\*** - You might get an error if you copied the command from manual on the VM, as it contains double quotes instead of single quote for parenthesis. Please make sure you use single quotes as above.



Check the Application on the browser, you should see the upgrade progress from 1.0 to 2.0.

**(Only Local cluster) Exercise 5: Debugging Lab**

1. Connect to the cluster, using the Azure cli –

*sfctl cluster select --endpoint http://localhost:19080*

* Copy the package, register it and create application –

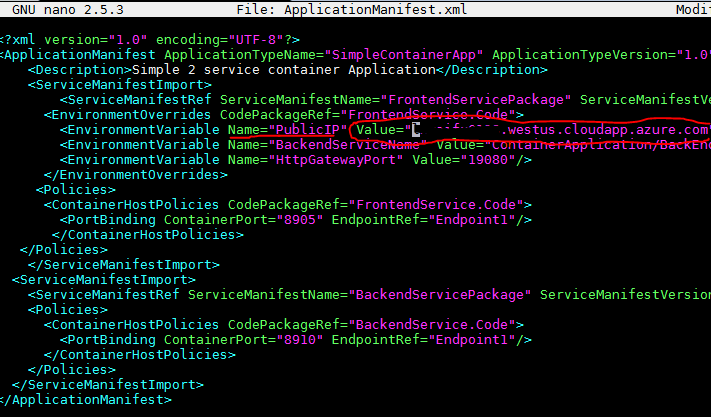
Make sure to change the FrontEndService Public IP to point to your localhost cluster in ApplicationManifest.xml. (use nano editor).

*cd ~/LabLinux*

*cd SimpleContainerAppL*

*nano ApplicationManifest.xml*

Even though it mentions “Public IP”, you need to add the localhost.



Use Ctrl+X to save and exit the nano editor.

*cd ..*

*sfctl application upload --path SimpleContainerAppL*

*sfctl application provision --application-type-build-path SimpleContainerAppL*

*sfctl application create --app-name fabric:/ContainerApplication --app-type SimpleContainerApp --app-version 1.0*

* Create the services (this can be created using Explorer or using the command below)

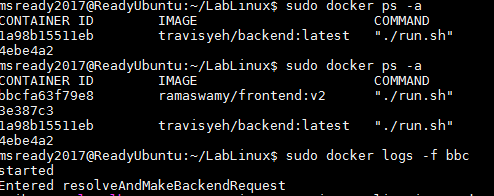
*sfctl service create --app-id ContainerApplication --name fabric:/ContainerApplication/BackEndService --service-type StatelessBackendService --stateless --instance-count 1 --singleton-scheme*

*sfctl service create --app-id ContainerApplication --name fabric:/ContainerApplication/FrontEndService --service-type StatelessFrontendService --stateless --instance-count 1 --singleton-scheme*

* Check the deployment on service fabric explorer for Linux, which will be <http://192.168.1.11:19080>.It will not load, so proceed to next steps to see what happened.

1. To debug docker logs get the CONTAINER ID with

*sudo docker ps -a*



Then get logs for the frontend container instance

*sudo docker logs -f [CONTAINER ID]*

Note the param “-f” in the command. Browse your site and you should see traces for the problem.

**Exercise 6: Additional Information**

This is an observation exercise only. Please browse to SFX and click on the cluster. Open the Manifest tab for your cluster. Check the details of the Nodes and Ports open on each node. Similarly observe the Manifest files for your Applications and Services.