Service Virtualization / DevTest

Lab/Troubleshooting Guide for Unmanaged Kubernetes Deployment

Support Guide

Version 1.1

This document ***is provided as*** is as an example of how to setup ***a DEV OR LAB environment only***.
This document is not designed or intended to be followed as a guide or UAT or PROD deployments as it
uses older version of software and disables security for ease of installation and configuration.

If you have problems with your environment, you will need to post question in the communities or consult your system /Kubernetes admins and experts for help.

Broadcom support cases regarding this documented will not be addressed.

Broadcom, the pulse logo, and Connecting everything are among the trademarks of Broadcom and/or its affiliates in the United States, certain other countries, and/or the EU.

Copyright © 2021 by Broadcom. All Rights Reserved.

The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, please visit www.broadcom.com.

Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

Table of Contents

Purpose

The purpose of this document is to help the end user setup a LAB or DEV environment for DevTest 10.7+ in an On-prem Kubernetes environment.

This document is NOT designed to be used as a guide to build a UAT or Prod Environment.

This document does not cover security and network concerns.

This document is provide ***STRICKLY AS IS*** and problems arising during its use will not be supported through normal Broadcom Support cases.
If you the end user needs help they can post question in the user communities here:
Service Virtualization (broadcom.com)
Or consult with their own internal System / Kubernetes Admin or experts.

Environment / Assumptions / Pre-requisites

System Requirements for the lab

The directions for this lab will require 5 Systems. 4 Linux machines and 1 Windows machine.
In this example, we will be using 4 Centos 7.5 machines and one windows 10 machine.
Note: It is assumed that the systems are all on the same network and have network connectivity and name resolution.

Each server has 4 CPU Cores /16 Gigs of ram/ 500 MBS of free diskspace

Server 1:
OS: Centos 7.9
Role: Stand Alone Server
Name: lvnktest001132.bpc.broadcom.net
IP: 10.173.36.18
Installed components:
MySQL 5.7.36 MySQL Community Server
DevTest 10.7 IAAM
DevTest 10.7 Enterprise Dashboard
MySQL jdbc driver version 5.1.45

Server 2:
OS: Centos 7.9
Role: Kubernetes Cluster Master
Name: lvnktest009975.bpc.broadcom.net
IP: 10.173.35.94
Installed components:
Docker version 1.31.1
Kubernetes version 1.22.3
Helm version 2.17.0
MySQL jdbc driver version 5.1.45

Server 3:
OS: Centos 7.9
Role: Kubernetes Worker Node 1
Name: lvnktest009976.bpc.broadcom.net
IP: 10.173.32.168
Installed components:
Docker version 1.31.1
Kubernetes version 1.22.3

Server 4:
OS: Centos 7.9
Role: Kubernetes Worker Node 2
Name: lvnktest009977.bpc.broadcom.net
IP: 10.173.38.42
Installed components:
Docker version 1.31.1
Kubernetes version 1.22.3

Workstation 1:
OS: Windows 10
Role: User Workstation
Name: lvnktest001897.bpc.broadcom.net
IP: 10.173.42.55
Installed components:
DevTest Workstation 10.7
MySQL Workbench

Prerequisites

- Kubernetes Unmanaged Cluster Setup
- Minimum Kubernetes version 1.21 required
- Minimum Docker 20.10 version required.
- Helm 2.17.0 required.
    - ***Helm 3.x is not supported.***
    - See command section for helm commands
    - See reference section for download information.
- Permissions to create Roles and Roles Binding in the namespace.
    - From Kubernetes 1.6 onwards, **RBAC policies are enabled by default**
- ***Enterprise Dashboard / IAAM should be running in On-premise or in cloud VM instances, i.e., EC2 for AWS or Compute for GCP, and connectivity with the Kubernetes Cluster should be established.***
- All Kubernetes (k8s) services are exposed as NodePorts by default.
    - Ensure all the ports that are exposed as NodePorts in worker nodes are available.
    - By default port range available in Kubernetes Cluster is ***30000-32767.***
- Workstation connects to the Registry Database to store test suite data.
    - Registry Database should be accessible to the Workstation Machines
- Kubernetes cluster should have 4 cores CPU,32 GB RAM, 200 GB disk space dedicated for DevTest
- The Devtest Docker containers are Linux-based and ***cannot be run on Windows Docker instances***.
- Each worker node will need a user and group created called devtest with a ***UID 1010*** as this is hardcoded into the images or the temp data directories need to have permissions set to ***777***
- Internet access must be enabled on all nodes, because required packages for kubernetes cluster will be downloaded from official repository.

Non-supported Use Cases

- Custom SSL certificates. You cannot use your own keystore file to connect SSL-enabled components.
- Setting up pods across different namespaces.
- Setting up pods across different clusters.
- Replicas for pods.
- Broker as pod.
- Docker containers for Windows.
- DevTest Workstation is not available as a Docker image.
- Enterprise Dashboard is not available as a Docker image.

Default URL after installation

- Enterprise Dashboard
    - http://<ED_IP>:1506/#/login
- IAAM
    - https://<IAAM_IP>:51111/
- Registry
    - http://<ClusterMasterIP>:32025/
- Portal
    - https:// <ClusterMasterIP>:32050/
- VSC
    - https:// <ClusterMasterIP>:32060/

Size of Docker Images

Docker Image Name	Image Size
config-server	625 MB
lisa	2.6 GB
portal	2.2 GB
virtual-service-catalog	993 MB

RAM Usage of Docker Containers

Container Name	Container RAM Size
Config-server	225 MB
Registry	2048 MB
Coordinator	512 MB
Simulator	512 MB
VSE	512 MB
Portal	2048 MB
Virtual Service Catalog	512 MB

Kubernetes Environment information.

- Install directory
  - /devtest_install
- Unix user and group
  - devtest
- Helm namespace
  - devtest107
- Kubernetes namespace
  - devtest-107
- Yaml Chart release_name
  - devtest107
- Kubernetes Cluster Master IP
  - 10.173.35.94
- Kubernetes PV directories
  - /tmp/data1
  - /tmp/data2
  - /tmp/data3
  - /tmp/data4
  - /tmp/data5

Kubernetes Persistent Volumes (PV)

Stateful resources in Kubernetes require ***Persistent Volumes (PV)*** for each of the components.
There are two ways PVs may be provisioned: ***Static and Dynamic***.
In any managed cluster like GCP or AWS, the default Storage class supports *ReadOnly* and *ReadWriteOnce* access modes. **Projects** PV needs to be configured with *ReadWriteMany* access mode.
Users may need to create a storage class to support it.
Please check Persistent Volumes for more reference.
Blank *storageClassName* takes default storageClass defined by the cluster.
By default
Registry DB PV is sized to 1 GB,
Projects PV is sized to 2 GB,
VSC PV is sized to 1 GB
VSE PV is sized to 2GB.
In case, the user wants to change the storage size for VSC or VSE PVs, the below lines need to be added in *custom-values.yml* file of *vsc/vse* section.

```
storage:
  storageClassName: ""
  requestSize: <new size>Gi
```

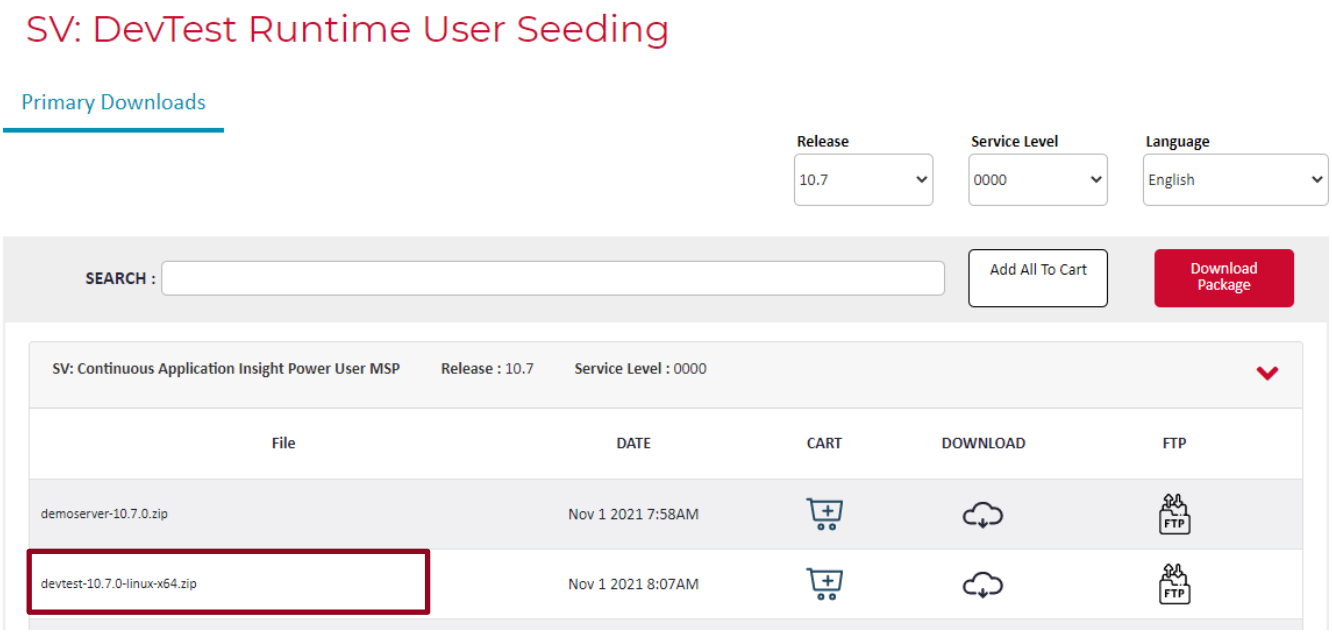
In case, Projects PV size and storage class needs to be changed, change it in *registry* section of *custom-values.yml*

```
storage:
  storageClassName: ""
  requestSizeForLisaDB: <new size>Gi
  requestSizeForProjects: <new size>Gi
```

Kubernetes Example Installation

Enterprise Dashboard / IAAM Standalone Server install

1) Download MySql Java Connector 5.1.45 Platform Independent (Architecture Independent), Compressed TAR Archive
    a. MySQL :: Download Connector/J
2) Download DevTest ***10.7*** installation media
    a. Access the following URL and enter valid credentials to download Broadcom software: https://support.broadcom.com/download-center/download-center.html.
    b. Type Service Virtualization into the search bar and click the SERVICE VIRTUALIZATION
    c. Click on the Devtest Run Time User Seeding link
    d. Download the devtest-10.7.0-Linux-x64.zip and optional Demoserver-10.7.0.zip



    e. Create an ***/installdir*** folder on the stand alone server and set the fire permissions to ***777***
    f. Transfer the mysql connector,demoserver and Devtest installer to the ***/installdir*** folder
    g. Extract the contents of the compressed files
    h. Disable the system firewall

```
systemctl disable firewalld
systemctl stop firewalld
```

    i. Install MySQL 5.7
        i. The below is an example script that will install MySql 5.7 create a devtest user and database for this example.

```
#!/usr/bin/bash

mysqlRootPass="R@cer123"

echo ' -> Removing previous mysql server installation'
systemctl stop mysqld.service && yum remove -y *mysql-community* && yum
remove -y mysql57-community-release.noarch && yum remove -y mysql80-
community-release*.noarch && rm -rf /var/lib/mysql && rm -rf
```

```
/var/log/mysqld.log && rm -rf /etc/my.cnf

echo ' -> Installing mysql server (community edition)'
yum localinstall -y https://dev.mysql.com/get/mysql57-community-
release-el7-7.noarch.rpm
yum install -y mysql-community-server

echo ' -> Starting mysql server (first run)'
systemctl enable mysqld.service
systemctl start mysqld.service
tempRootDBPass="`grep 'temporary.*root@localhost' /var/log/mysqld.log |
tail -n 1 | sed 's/.*root@localhost: //'`"

echo ' -> Setting up new mysql server root password'
systemctl stop mysqld.service
rm -rf /var/lib/mysql/*logfile*
# wget -O /etc/my.cnf "https://my-site.com/downloads/mysql/512MB.cnf"
/usr/bin/cp -f ./my5.7.cnf /etc/my.cnf
systemctl start mysqld.service
mysqladmin -u root -password="$tempRootDBPass" password
"$mysqlRootPass"
mysql -u root -password="$mysqlRootPass" <<EOSQL
    CREATE USER 'root'@'%' IDENTIFIED BY '${mysqlRootPass}';
    GRANT ALL ON *.* TO 'root'@'%' WITH GRANT OPTION;
    FLUSH PRIVILEGES;
    CREATE USER 'devtest'@'%' IDENTIFIED BY '${mysqlRootPass}';
    GRANT ALL ON *.* TO 'devtest'@'%' WITH GRANT OPTION;
    FLUSH PRIVILEGES;
    CREATE DATABASE DevTest_ED
    default character set utf8
    default collate utf8_unicode_ci;
    CREATE DATABASE DevTest_Reg
    default character set utf8
    default collate utf8_unicode_ci;
    CREATE DATABASE DevTest_IAM
    default character set utf8
    default collate utf8_unicode_ci;
EOSQL

systemctl status mysqld.service
      echo " -> MySQL server installation completed, root password:
      $mysqlRootPass";
```

j. Run the Devtest 10.7 installer to install IAAM and Enterprise dashboard.
   i. Below is an example VAR file to be used with the following silent installer command
   ii. ./devtest_linux_x64.sh –q –varfile ./response_10.7_MYSQL_Linux.varfile
      1. Lines in Yellow need to be modified

```
# install4j response file for DevTest Solutions 10.7.0
authenticateOnly=false
autoAddUsers=true
caAgreementChoice=2
createDesktopLinkAction$Boolean=false
dbHost=lvnktest001132
dbName=DevTest_IAM
dbPort$Long=3306
dbType$Integer=3
dbUsername=devtest
dbPassword=R@cer123
demoServerZipFile=/installdir/DevTestDemoServer.zip
```

```
driverPath=/installdir/mysql-connector-java-5.1.45/mysql-
connector-java-5.1.45-bin.jar
enterpriseDashboardServer=lvnktest001132\:1506
iamServerUrl=https\://lvnktest001132\:51111/auth
installDemoServer=yes
installEnterpriseDashboard=Yes
installServer=no
installWorkstation=No
# If installing SERVER COMPONENTS, we strongly recommend that
the lisaDataDir is BLANK or is the same as the install
directory.
# If installing WORKSTATION ONLY, the lisaDataDir is used to
define the data directory location.
# When an administrator is installing WORKSTATION ONLY on
behalf of another user, it is recommended to use
%userprofile%\\DevTest or ~/DevTest. Note, the end user needs
to have write permission for the data directory.
# Workstation data directory components: _local.properties,
logging.properties, de-identify.xml, hotdeploy, library for
customizations, projects with assets, example folders
lisaDataDir=/opt/CA/DevTest_10.7
plaCompanyDomain=Broadcom.com
plaEnterpriseSiteId=XXXXXX
plaInternalIdentifier=Support Lab
plaProxyPassword.encoded=
plaProxyUri=
plaProxyUsed=false
plaProxyUsername=
plaUsed=true
sys.adminRights$Boolean=true
sys.component.38273$Boolean=true
sys.installationDir=/opt/CA/DevTest_10.7
sys.languageId=en
sys.programGroupDisabled$Boolean=false
sys.symlinkDir=/usr/local/bin
```

iii. Copy the MySQL Connector jar file to the <Devtest>/lib/shared and <Devtest>/lib/dradis folders

```
cp /installdir/mysql-connector-java-5.1.45/mysql-connector-java-
5.1.45-bin.jar /opt/CA/DevTest_10.7/lib/shared
cp /installdir/mysql-connector-java-5.1.45/mysql-connector-java-
5.1.45-bin.jar /opt/CA/DevTest_10.7/lib/dradis
```

iv. Edit the <Devtest>***/dradis.properties*** to add the following:

```
dradis.db.driverClass=com.mysql.jdbc.Driver
dradis.db.url=jdbc:mysql://[SERVERNAME]:3306/DevTest_ED
dradis.db.user=devtest
dradis.db.password=R@cer123
```

v. Edit the <Devtest>***/site.properties*** to add the following:

```
lisadb.pool.common.driverClass=com.mysql.jdbc.Driver
lisadb.pool.common.url=jdbc:mysql://[SERVERNAME]:3306/DevTest_Reg
lisadb.pool.common.user=devtest
lisadb.pool.common.password=R@cer123
dradis.db.password=R@cer123
```

vi. Edit the <Devtest>***/lisa.properties*** to add the following:

```
## Configuring IAM
```

```
iam.server.url=https://[SERVERNAME]:51111/auth
## ========================================
## ========================================
devtest.enterprisedashboard.host=[SERVERNAME]
devtest.enterprisedashboard.port=1506
devtest.enterprisedashboard.https.enabled=false
# Should we start the internal Derby DB instance in the registry?
lisadb.internal.enabled=false
# Should we start the internal Derby DB instance in the Enterprise
Dashboard?
cicdb.internal.enabled=false
```

vii. Start the Devtest IAAM and ED services and make sure they are working as expected.
   1. Default URLs
      a. Enterprise dashboard
         i. http:/ [SERVERNAME]:1506/#/login
      b. IAAM
         i. https:// [SERVERNAME]:51111/

Default username and password admin/admin

Kubernetes Master and worker node install

Common Installation steps to be done on Master Node and Worker nodes.

1) Disable the system firewall.

```
systemctl disable firewalld
systemctl stop firewalld
```

2) Kubernetes doesn't support "Swap". Disable Swap on all nodes using below command and also to make it permanent comment out the swap entry in /etc/fstab file.

```
sed -i '/swap/s/^/#/' /etc/fstab
swapoff -a
```

3) Disable or set SELinux to permissive to or the containers will not start due to permission errors

```
setenforce 0
sed -i -e 's/SELINUX=enforcing/SELINUX=permissive/g' /etc/selinux/config
```

Command to check: sestatus

4) Create the devtest user in group at the OS level

```
groupadd --gid 1010 devtest
useradd -u 1010 -g devtest devtest
```

5) Create the Kubernetes Persistent volume directories, change permissions and set ownership

```
mkdir -p /tmp/data1 /tmp/data2 /tmp/data3 /tmp/data4 /tmp/data5
chmod -R 766 /tmp/data1 /tmp/data2 /tmp/data3 /tmp/data4 /tmp/data5
chown -R devtest /tmp/data1 /tmp/data2 /tmp/data3 /tmp/data4 /tmp/data5
```

6) Enable Kubernetes repository on master and all worker nodes

   a. Create a repo file for kubernetes and append the lines given below.

```
/etc/yum.repos.d/kubernetes.repo

[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
        https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
```

7) add the following line to **/etc/modules-load.d/k8s.conf**

```
br_netfilter
```

8) run the following command

```
modprobe br_netfilter
```

9) Add the following line to **/etc/sysctl.d/k8s.conf**

```
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
```

10) run the following command

```
sysctl --system
```

11) Install the required packages on master and all worker nodes

   a. Install "docker" and "kubeadm" packages using yum command.

```
yum -y install docker kubeadm kubelet kubectl
```

12) Add the following line to **/etc/sysconfig/docker-network**

```
DOCKER_NETWORK_OPTIONS="--iptables=false --ip-masq=false"
```

13) Add the following line to **/etc/sysconfig/docker-storage**

```
DOCKER_STORAGE_OPTIONS="--storage-driver overlay2"
```

14) Add the following line to **/etc/sysconfig/kubelet**

```
KUBELET_EXTRA_ARGS="--cgroup-driver=systemd"
```

15) Start and Enable docker and kubelet services on master and all worker nodes

```
systemctl start docker && systemctl enable docker
systemctl enable kubelet
```

Steps to be done only on Master Node

1) Create the devtest_install directory and sub folders and change permissions

```
mkdir /devtest_install
mkdir /devtest_install/config
mkdir /devtest_install/dbdriver
chmod -R 777 /devtest_install
```

2) Copy the MySQL Connector Jar file, mysql-connector-java-5.1.45-bin.jar to the devtest_install/dbdriver folder from the ED?IAAM server

3) Copy the following files from the ED/IAAM Server to the /devtest_install/config folder

   a. lisa.properties
   b. local.properties
   c. logging.properties
   d. phoenix.properties
   e. site.properties

4) Initializing and setting up the kubernetes cluster on Master node

   a. Use "kubeadm" command to initialize the kubernetes cluster along with "apiserver-advertise-address" and "--pod-network-cidr" options. It is used to specify the IP address for kubernetes cluster communication and range of networks for the pods.

```
kubeadm init --apiserver-advertise-address <ClusterMasterIP> --pod-network-cidr=10.244.0.0/16
```

Eexample:

```
kubeadm init --apiserver-advertise-address 10.173.35.94 --pod-network-cidr=10.244.0.0/16
```

Output:

[init] using Kubernetes version: v1.11.2
[preflight] running pre-flight checks
[WARNING Firewalld]: firewalld is active, please ensure ports [6443 10250] are open or your cluster may not function correctly
I0811 21:10:04.905996 12195 kernel_validator.go:81] Validating kernel version
I0811 21:10:04.906058 12195 kernel_validator.go:96] Validating kernel config
[preflight/images] Pulling images required for setting up a Kubernetes cluster
[preflight/images] This might take a minute or two, depending on the speed of your internet connection
[preflight/images] You can also perform this action in beforehand using 'kubeadm config images pull'
[kubelet] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[preflight] Activating the kubelet service
[certificates] Generated ca certificate and key.
……
suppressed few messages

[bootstraptoken] creating the "cluster-info" ConfigMap in the "kube-public" namespace
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy
Your Kubernetes master has initialized successfully!

To start using your cluster, you need to run the following as a regular user:
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/
You can now join any number of machines by running the following on each node as root:

kubeadm join 10.173.35.94:6443 --token pxavv6.zwqgdlivwfgbaaud --discovery-token-ca-cert-hash sha256:0cd1e77fd1514a6ec60e3c67c678c0d88ac80b18ff8184271ecef1ccdc01ee55

5) Kubernetes cluster initialization is completed, Copy the join token highlighted in blue color from the "kubeadm init" command output and store it somewhere, it is required while joining the worker nodes.
6) Copy /etc/kubernetes/admin.conf and Change Ownership only on Master node
    a. Once kubernetes cluster is initialized, copy "/etc/kubernetes/admin.conf" and change ownership. You will see this same instructions in the output of "kubeadm init" command.

```
mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

7) Install Network add-on to enable the communication between the pods only on Master node
    a. We have lot of network add-on available to enable the network communication with different functionality, Here I have used flannel network provider. Flannel is an overlay network provider that can be used with Kubernetes. You can refer more add-on from here.

```
kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
```

8) Use "kubectl get nodes" command to ensure the kubernetes master node status is ready.
Wait for few minutes until the status of the kubernetes master turn ***ready state.***

```
kubectl get nodes
NAME                STATUS    ROLES     AGE       VERSION
kubernetes-master   Ready     master    14m       v1.11.1
```

9) Download Helm 2.1.7 to the /devtest_install directory
10) Extract the helm files

```
tar -xvf helm-v2.17.0-linux-amd64.tar.gz
```

11) Move the helm file into place

```
mv ./linux-amd64/helm /usr/local/bin/helm
```

12) Taint the master node to allow Tiller pod to be installed

```
kubectl taint nodes --all node-role.kubernetes.io/master-
```

13) Create the tiller service account

```
kubectl create serviceaccount --namespace kube-system tiller
```

14) Create Tiller Cluster role

```
kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --
serviceaccount=kube-system:tiller
```

15) Run the helm init command

```
/usr/local/bin/helm init
```

16) Update Master with new role

```
kubectl patch deploy --namespace kube-system tiller-deploy -p
'{"spec":{"template":{"spec":{"serviceAccount":"tiller"}}}}'
```

17) Apply the tiller account to the new Tiller pod

```
/usr/local/bin/helm init --service-account tiller --upgrade
```

Steps to be done on only Worker Nodes

1) Join all worker nodes with kubernetes master node

    a. Login into all worker nodes and use the join token what you have copied earlier to join all the worker nodes with kubernetes master node as below

    ```
    kubeadm join 10.173.35.94:6443 --token pxavv6.zwqgdlivwfgbaaud --discovery-token-ca-cert-hash sha256:0cd1e77fd1514a6ec60e3c67c678c0d88ac80b18ff8184271ecef1ccdc01ee55
    ```

    Note: To regenerate a join token run the following command on the Master Node:
    kubeadm token create --print-join-command

To validate the nodes have joined successfully run the ***kubectl get nodes*** on the Master Node.

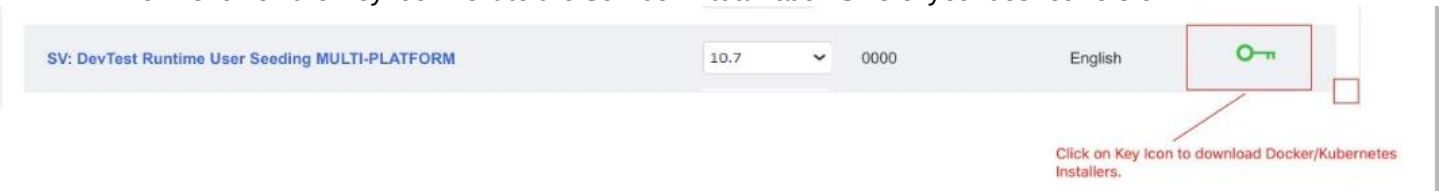
DevTest Kubernetes Nodes Ports Install done on Master Node

NOTE: All command and files are expected to be installed and run from the ***<devtest_install>/*** install directory

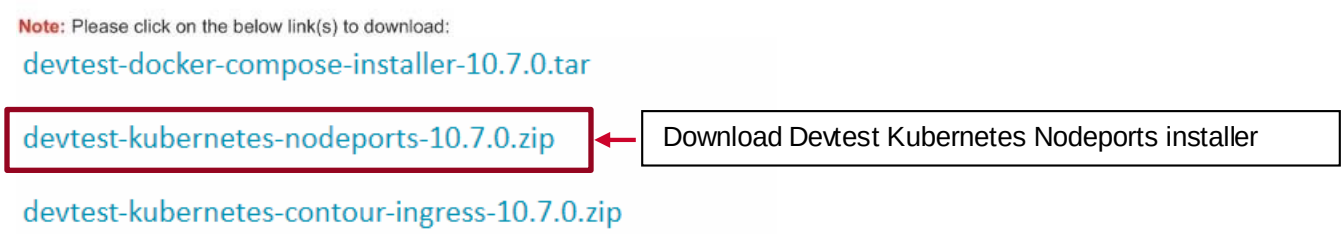
1) Download MySql Java Connector 5.1.45 Platform Independent (Architecture Independent), Compressed TAR Archive
    a. MySQL :: Download Connector/J
    b. Extract mysql-connector-java-5.1.45-bin.jar to the ***<devtest_install>/dbdriver*** folder
2) Copy the following files from the ED/IAAM Server to the /devtest_install/config folder
    a. lisa.properties
    b. local.properties
    c. logging.properties
    d. phoenix.properties
    e. site.properties
3) edit the sites.properties file to update the database connection as follows highlighted in yellow

```
lisadb.pool.common.driverClass=com.mysql.jdbc.Driver
lisadb.pool.common.url=jdbc:mysql://lvnktest001132:3306/DevTest_Reg?useSSL=false
lisadb.pool.common.user=devtest
lisadb.pool.common.password_enc=1098e0ccc3d5f4ddf7882d6fb73c89b17d25fba668113a10
1ca1b5d7b6460a0e7d2afef46a39af2b2
```

4) Download DevTest ***10.7*** installation media
    a. Access the following URL and enter valid credentials to download Broadcom software: https://support.broadcom.com/download-center/download-center.html.
    b. Type Service Virtualization into the search bar and click the SERVICE VIRTUALIZATION
    c. Click on the Key icon next to the Service Virtualization SKU of your desired version.



    d. Download the Devtest Kubernetes Nodeports Installer.



e. Unzip the installer. You will be able to see the following files:

```
staff    160 Oct  5 00:55 Examples
staff   5947 Oct  5 16:42 custom-values-SSL-Template.yaml
staff   5821 Oct  5 16:42 custom-values-nonSSL-Template.yaml
staff  12124 Oct  5 16:43 devtest-0.2.0.tgz
```

5) Make a copy of custom-values-nonSSL-Template.yaml file with name custom-values.yml and update the username and password with the user mail id and access token respectively as mentioned in the previous step. Below is the sample ***custom-values.yml before changes.***

```
# DevTest Service Account
serviceAccount:
  enabled: true
  name: devtest

imageCredentials:
  secretName: servicevirtual-cred
  registry: sv-docker.packages.broadcom.com
  username: <customer_name>
  password: <access_token>

config-server:
  enabled: true
  fullnameOverride: <release_name>-config-server
  image:
    repository: sv-docker.packages.broadcom.com/sv/config-server
    pullPolicy: IfNotPresent
    tag: latest

registry:
  enabled: true
  fullnameOverride: <release_name>-registry
  databaseDriver: db-driver
  dataFileConfigMap: devtest-config
  storage:
    requestSizeForProjects: 2Gi
    storageClassNameForProjects: ""
  image:
    repository: sv-docker.packages.broadcom.com/sv/lisa
    pullPolicy: IfNotPresent
    tag: latest
  container:
    env:
      IAAM_URL: "https://<IAAM/IP>:51111/auth"
      ENTERPRISE_DASHBOARD_SERVICE_NAME: "<enterprise_dashboard/ip>"
      ENTERPRISE_DASHBOARD_SERVICE_PORT: <enterprise_dashboard_port>
      ENTERPRISE_DASHBOARD_SERVICE_HTTPS_ENABLED: false
      CONFIG_SERVER_URL: "http://<release_name>-config-server:8888"
      REGISTRY_POD_PORT: 32020
      REGISTRY_NAME: "Registry"
      REGISTRY_URL: "tcp://<KubMasterIP>:32020/$(REGISTRY_NAME)"
      EXTRA_JAVA_OPTS: "-Dlisa.webserver.port=32025 -
Dlisa.threadDump.generate=false"
  service:
    type: NodePort
    registryPort: 32020
    registryNodePort: 32020
    invokePort: 32025
```

```
    invokeNodePort: 32025
    dbPort: 32030
    dbNodePort: 32030

coordinator:
  enabled: true
  fullnameOverride: <release_name>-coordinator
  databaseDriver: db-driver
  dataFileConfigMap: devtest-config
  registry:
    fullName: <release_name>-registry
  image:
    repository: sv-docker.packages.broadcom.com/sv/lisa
    pullPolicy: IfNotPresent
    tag: latest
  container:
    port: 32035
    env:
      JAVA_OPTS: “”
      CONFIG_SERVER_URL: "http://<release_name>-config-server:8888"
      REGISTRY_SERVICE_NAME: "<release_name>-registry"
      REGISTRY_INVOKE_SERVICE_PORT: "32025"
      REGISTRY_URL: "tcp://<KubMasterIP>:32020/Registry"
      COORDINATOR_URL: "tcp://<KubMasterIP>:32035/$(COORDINATOR_NAME)"
      REGISTRY_WEBSERVER_HTTPS_ENABLED: false
      REGISTRY_WEBSERVER_HOST: "<KubMasterIP>"
  service:
    type: NodePort
    port: 32035
    nodePort: 32035

simulator:
  enabled: true
  fullnameOverride: <release_name>-simulator
  databaseDriver: db-driver
  dataFileConfigMap: devtest-config
  registry:
    fullName: <release_name>-registry
  image:
    repository: sv-docker.packages.broadcom.com/sv/lisa
    pullPolicy: IfNotPresent
    tag: latest
  container:
    port: 32040
    env:
      JAVA_OPTS: ""
      CONFIG_SERVER_URL: "http://<release_name>-config-server:8888"
      REGISTRY_SERVICE_NAME: "<release_name>-registry"
      REGISTRY_INVOKE_SERVICE_PORT: "32025"
      REGISTRY_URL: "tcp://<KubMasterIP>:32020/Registry"
      SIMULATOR_URL: "tcp://<KubMasterIP>:32040/$(SIMULATOR_NAME)"
      REGISTRY_WEBSERVER_HTTPS_ENABLED: false
      REGISTRY_WEBSERVER_HOST: "<KubMasterIP>"
  service:
    type: NodePort
    port: 32040
    nodePort: 32040

portal:
  enabled: true
  fullnameOverride: <release_name>-portal
```

```
    databaseDriver: db-driver
    dataFileConfigMap: devtest-config
    registry:
      fullName: <release_name>-registry
    image:
      repository: sv-docker.packages.broadcom.com/sv/portal
      pullPolicy: IfNotPresent
      tag: latest
    container:
      env:
        REGISTRY_SERVICE_NAME: "<release_name>-registry"
        REGISTRY_INVOKE_SERVICE_PORT: "32025"
        JAVA_OPTS: "-DphoenixConfig.resHubServiceLoggingLevel=FULL -
Dregistry.host=<KubMasterIP> -Dregistry.port=32020 -
Dregistry.portal.port=32025 -
DlisaAutoConnect=tcp://<KubMasterIP>:32020/Registry"
        CONFIG_SERVER_URL: "http://<release_name>-config-server:8888"
        REGISTRY_WEBSERVER_HTTPS_ENABLED: false
        REGISTRY_WEBSERVER_HOST: "<KubMasterIP>"
    service:
      type: NodePort
      nodePort: 32050

  vse:
    enabled: true
    fullnameOverride: <release_name>-vse
    databaseDriver: db-driver
    dataFileConfigMap: devtest-config
    registry:
      fullName: <release_name>-registry
    storage:
      storageClassName: ""
    image:
      repository: sv-docker.packages.broadcom.com/sv/lisa
      pullPolicy: IfNotPresent
      tag: latest
    container:
      port: 32055
      env:
        JAVA_OPTS: ""
        REGISTRY_SERVICE_NAME: "<release_name>-registry"
        REGISTRY_INVOKE_SERVICE_PORT: "32025"
        REGISTRY_URL: "tcp://<KubMasterIP>:32020/Registry"
        VSE_URL: "tcp://<KubMasterIP>:32055/VSE"
        REGISTRY_WEBSERVER_HTTPS_ENABLED: false
        REGISTRY_WEBSERVER_HOST: "<KubMasterIP>"
    headlessService:
      port: 32055
    service:
      type: NodePort
      port: 32055
      nodePort: 32055

  virtual-service-catalog:
    enabled: true
    fullnameOverride: <release_name>-vsc
    dataFileConfigMap: devtest-config
    image:
      repository: sv-docker.packages.broadcom.com/sv/virtual-service-catalog
      pullPolicy: IfNotPresent
      tag: latest
```

```
container:
  env:
    IAAM_URL: "https://<iaam_hostname>:51111/auth"
service:
  type: NodePort
  nodePort: 32060
```

6) Replace all the references of <release-name> in the custom-values.yml with the release name you want.
Example:
From: fullnameOverride: <release_name>-config-server
To: fullnameOverride: devtest107-config-server

7) Change NodePorts of all the services according to the Ports freely available in the cluster.
   a. If using default ports no change is required.

8) Replace all references of <KubMasterIP> with Cluster Master Node IP.
Example:
From: REGISTRY_URL: "tcp://<KubMasterIP>:32020/Registry"
To: REGISTRY_URL: "tcp:// 10.173.35.94:32020/Registry"

9) Modify the the following om the **Registry section** of the ***custom-values.yml*** with the IP address and port of the standalone Enterprise Dashboard and IAAM:
<ENTERPRISE_DASHBOARD_SERVICE_NAME>
<ENTERPRISE_DASHBOARD_SERVICE_POR>
<IAAM_URL>

```
ENTERPRISE_DASHBOARD_SERVICE_NAME: "<ED_ip>"
ENTERPRISE_DASHBOARD_SERVICE_PORT: <ED_port_number> (Default port is 1506)
IAAM_URL: https://<iaam_IP>:51111/auth
```

10) Modify the **IAAM_URL** in the **Virtual Service Catalog section** in the **custom-values.yml** with the IP address and port of the standalone IAAM:

```
IAAM_URL: https://<iaam_IP>:51111/auth
```

11) Create a namespace where devtest components will be deployed.

```
kubectl create ns <namespace>
```
Example:
kubectl create ns devtest-107

12) Create Database configmap

```
kubectl create configmap db-driver --from-file=./dbdriver/mysql-connector-java-5.1.45-bin.jar -n devtest-107
```

13) Create Configuration files configmap

```
kubectl create configmap devtest-config --from-file=./config --namespace devtest-107
```

14) Create a pv.yaml file in the <devtest_install> folder
   a. Example pv.yaml below

```
apiVersion: v1
kind: PersistentVolume
metadata:
    name: pv1
spec:
    accessModes:
      - ReadWriteMany
    capacity:
      storage: 2Gi
    hostPath:
      path: /tmp/data1
---
```

```
apiVersion: v1
kind: PersistentVolume
metadata:
    name: pv2
spec:
    accessModes:
      - ReadWriteOnce
      - ReadOnlyMany
    capacity:
      storage: 1Gi
    hostPath:
      path: /tmp/data2
---
apiVersion: v1
kind: PersistentVolume
metadata:
    name: pv3
spec:
    accessModes:
      - ReadWriteOnce
      - ReadOnlyMany
    capacity:
      storage: 1Gi
    hostPath:
      path: /tmp/data3
---
apiVersion: v1
kind: PersistentVolume
metadata:
    name: pv4
spec:
    accessModes:
      - ReadWriteOnce
      - ReadOnlyMany
    capacity:
      storage: 2Gi
    hostPath:
      path: /tmp/data4
```

15) Use below command to create PVs:

```
kubectl apply -f pv.yaml
```

16) Deploy the DevTest helm charts using the following command

a. helm upgrade <release-name> ./devtest-X.X.X.tgz --install --values ./custom-values.yaml --namespace <namespace>
Example:
helm upgrade devtest107 ./devtest-0.2.0.tgz --install --values ./custom-values.yaml --namespace devtest-107

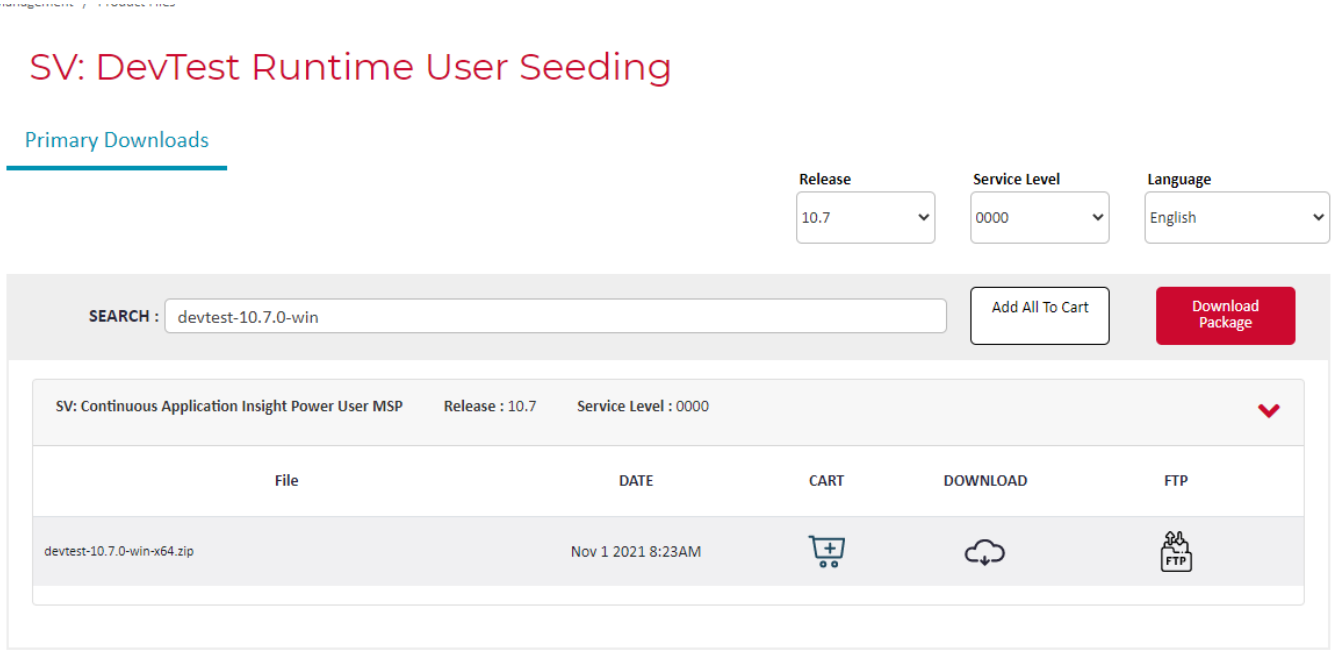
17) Check Deployment status

a. kubectl get pods --namespace <namespace>
Example:
kubectl get pods --namespace devtest-107

b. kubectl describe pods --namespace <namespace>
Example:
kubectl describe pods --namespace devtest-107

Workstation installation

1) Download MySql Java Connector 5.1.45 Platform Independent (Architecture Independent), Compressed TAR Archive
    a. MySQL :: Download Connector/J
2) Download DevTest ***10.7*** installation media
    a. Access the following URL and enter valid credentials to download Broadcom software: https://support.broadcom.com/download-center/download-center.html.
    b. Type Service Virtualization into the search bar and click the SERVICE VIRTUALIZATION
    c. Click on the Devtest Run Time User Seeding link
    d. Download the devtest-10.7.0-win-x64.zip



    e. Create an **c:\installdir** folder on the workstation machine
    f. Transfer the mysql connector and Devtest installer to the c:\installdir folder
    g. Extract the contents of the compressed files
    h. Run the workstation installer devtest_win_x64.exe and choose workstation only
    i. Copy the MySQL connector file, **mysql-connector-java-5.1.45-bin.jar**, to the **<DevTest_Install>\lib\shared** directory
    j. When starting the workstation point to the Cluster IP registry URL http://<ClusterMasterIP>:32025/

Devtest Administration

Adding additional Kubernetes Simulators /Coordinators /Virtual Service Engine (VSE)

When doing an on-prem Kubernetes install a compressed file is used to setup the environment. This file has a name such as ***devtest-0.2.0.tgz.***
To add an additional simulator, coordinator or VSE it is required to extract the files from devtest-0.2.0.tgz and modify them. The steps are below.

NOTE:
Steps 6,7,8 are ONLY required when increasing the number of VSE.
These steps are not required for simulators or coordinator

1) Untar tgz file using the command **tar –zxf <tgz>** in the previously created devtest install directory
2) Go to devtest charts directory using command **cd devtest/charts**
3) Copy the component that needs to be duplicated.IE simulator or coordinator or VSE
   example:
   copy simulator chart directory to simulator2 using command **cp –R simulator simulator2**
4) Update name in file **<component>/*Chart.yaml*** using command ***vi* <component>/*Chart.yaml*** and change name to **<component>**2. See highlighted example changes below:

```
apiVersion: v1
appVersion: 10.6.1
description: A Helm chart for a DevTest Simulator
name: simulator2
version: 0.2.0
```

5) Duplicate **<component>** configuration in **custom-values.yaml** file and change duplicate configuration.
   See highlighted example changes below:
   Example for simulator

```
simulator2:
  enabled: true
  fullnameOverride: devtest107-simulator2
  databaseDriver: db-driver
  dataFileConfigMap: devtest-config
  registry:
    fullName: devtest107-registry
  image:
    repository: sv-docker.broadcom.com/sv/lisa
    pullPolicy: IfNotPresent
    tag: 10.7.1
  container:
    port: 32041
    env:
      JAVA_OPTS: "-Dlisa.threadDump.generate=false"
      CONFIG_SERVER_URL: "http://devtest10-config-server:8888"
      REGISTRY_SERVICE_NAME: "devtest107-registry"
      REGISTRY_INVOKE_SERVICE_PORT: "32025"
      REGISTRY_URL: "tcp://10.173.35.94:32020/Registry"
      SIMULATOR_URL: "tcp://10.173.35.94:32041/Simulator2"
      REGISTRY_WEBSERVER_HTTPS_ENABLED: false
      REGISTRY_WEBSERVER_HOST: "10.173.35.94"
  service:
    type: NodePort
```

```
    port: 32041
    nodePort: 32041
```

6) On the worker nodes add a new ***/tmp/dataX*** directory for each new component added.

   a. Example:
      /tmp/data5
   b. Change the permissions to 776
   c. Change the owner to devtest

7) Add a new section in the pv.yaml pointing to the new directory structure

```
apiVersion: v1
kind: PersistentVolume
metadata:
    name: pv5
spec:
    accessModes:
      - ReadWriteOnce
      - ReadOnlyMany
    capacity:
      storage: 2Gi
    hostPath:
      path: /tmp/data5
```

8) Update the pvc information by running

```
kubectl apply –f pv.yaml
```

9) After doing these changes, run the helm upgrade command pointing to ***devtest*** directory instead of tgz file.
   Example:

```
helm upgrade devtest107 ./devtest –install –values ./custom-values.yaml –
namespace devtest-107
```

Steps to Upgrade DevTest On-Prem Kubernetes installation.

In this example, we will cover upgrading a 10.6 Kubernetes on-prem environment to a 10.7 DevTest environment.

Trouble shooting and commands

Helm Commands

1) Command to check the helm version
    a. Helm version
2) Command to check all of the releases for a specified namespace
    a. helm list –namespace <helmnamespace>
3) apply changes from custom-values.yaml
    a. helm upgrade <helmnamespace> ./devtest-0.X.X.tgz –install –values ./custom-values.yaml –namespace <namespace>
    Example:
    helm upgrade devtest107 ./devtest-0.1.7.tgz --install --values ./custom-values.yaml --namespace devtest-107
4) helm upgrade command to use folder instead of tgz file
    a. helm upgrade <helmnamespace> ./devtest –install –values ./custom-values.yaml –namespace <namespace>
    Example:
    helm upgrade devtest107 ./devtest --install --values ./custom-values.yaml --namespace devtest-107
5) remove helm deployment
    a. helm delete –purge <helmnamespace>
    Example:
    helm delete --purge devtest107
6) To restart pods this is only available ***before helm 3.X***
    a. helm upgrade --recreate-pods <helmnamespace> ./devtest-0.1.7.tgz --install --values ./custom-values.yaml --namespace <namespace>
    Example:
    helm upgrade --recreate-pods devtest106 ./devtest-0.1.7.tgz --install --values ./custom-values.yaml --namespace devtest-106
7) Command to update service account user
    a. helm init --service-account tiller --upgrade

Kubernetes Commands

1) Command to apply new persistent volume information

```
kubectl apply –f pv.yaml
```

2) Command to list pods on the cluster master

```
kubectl get po -n kube-system
```

3) Command to list devtest pods in <namespace>

```
kubectl get pods -n <namespace>
Example:
kubectl get pods -n devtest-107
```

4) Command to list devtest pods details in <namespace>

```
kubectl get pods -o wide -n <namespace>
Example:
kubectl get pods -o wide -n devtest-107
```

5) Command to open command pod command prompt
   a) kubectl exec -it <podname> -n <namespace> -- /bin/sh

      Example:
      kubectl exec -it devtest107-portal-0 -n devtest-107 -- /bin/sh

      NOTE: Logs are stored in **/home/devtest/lisatmp_XX.XX**

6) Command to delete configmap for database driver and config files
   a) kubectl delete configmap <ConfigMapName> --namespace <namespace>

      Example Database Driver:
      kubectl delete configmap db-driver --namespace devtest-107
      Example Config files::
      kubectl delete configmap devtest-config --namespace devtest-107

7) Command to create configmap for database driver and config files
   a) kubectl create configmap <ConfigMapName> --from-file=<fullpathtofile>-n <namespace>

      Example Database Driver:
      kubectl create configmap db-driver --from-file=./dbdriver/mysql-connector-java-5.1.45-bin.jar -n devtest-107
      Example Config files:
      kubectl create configmap devtest-config --from-file=./config -n devtest-107

8) Command to check a pods logs for the last x number of hours
   a) kubectl logs --since=Xh <podname> -n <namespace>
      Example:
      kubectl logs --since=1h devtest107-registry-0 -n devtest-107

9) Command to Taint the primary node to allow Tiller pod install
   a) kubectl taint nodes --all node-role.kubernetes.io/master-

10) Command to remove Taint from the Cluster Master so devtest pods are not deployed there
    a) kubectl taint nodes $(hostname) node-role.kubernetes.io/master:NoSchedule

11) Command to list kubernetes persistent volumes
    a) kubectl get pv

12) Command to create service account

a) kubectl create serviceaccount --namespace <anemspace> <serviceAccoutName>

Example:
kubectl create serviceaccount --namespace kube-system tiller

13) Command to create role binding

```
kubectl create clusterrolebinding tiller-cluster-rule --clusterrole=cluster-admin --serviceaccount=kube-system:tiller
```

14) Command to deploy service account user

```
kubectl patch deploy --namespace kube-system tiller-deploy -p '{"spec":{"template":{"spec":{"serviceAccount":"tiller"}}}}'
```

15) Command to regenerate a join token for worker nodes:

```
kubeadm token create --print-join-command
```

16) Command to list nodes in cluster

```
kubectl get nodes
```

17) Command to get configuration information for a pod

a) kubectl describe pod <poodname> -n <namespace>

Example:
kubectl describe pod devtest107-portal-0 -n devtest-107

18) Command to copy folder from Pod.

a) kubectl cp <some-namespace>/<some-pod>:/tmp/foo /tmp/bar
Example:
kubectl cp devtest-107/devtest107-registry-0:/home/devtest ./devtest

19) Command to list ports

a) kubectl get svc -o wide -n <namespace>
Example:
kubectl get svc -o wide -n devtest-107

20) Command to list configmaps in a name space

a) kubectl get configmap --namespace <namespace>
Example:
kubectl get configmap --namespace devtest-107

Problems and solutions

We added a new component but it stays in a pending state:

After adding a new component VSE, Simulator or Coordinator the new pod is not deployed and stays in a p[ending state.
If you check the pod description you see an error such as:
Events:

Type	Reason	Age	From	Message
----	------	----	----	-------
Warning	FailedScheduling	13s (x3 over 2m15s)	default-scheduler	0/3 nodes are available: 3 pod has unbound immediate PersistentVolumeClaims.

Solution:
on the nodes make sure to create a new /tmp/datax directory for the module, set the permissions to 777 and change the owner to devtest.

Next update the pv.yaml file and add a new entry for this directory.

When accessing IAM server from Kuberentes environment is it redirecting to the local host.

The Kubernetes environment require an on premise IAM server and ED server.
By default when you install IAM and ED they default to a setting of using the localhost for the configuration.

To correct for this, make sure ***the lisa.properties/local.properties*** on IAM and ED server has the below properties configured with Hostname or IpAddress instead of localhost.

. **iam.server.url**=https://<IAM_Server_Hosname> or <FQDN> or IpAddr:51111/auth

. **devtest.enterprisedashboard.host**=<ED_Server_Hostname> or <FQDN> or IpAddr

Note: The ServerName/FQDN?IPaddr need to be able to be resolved from the Kubernetes environment.
And idealy should match what is used in the custom-values.yaml

example:
IAAM_URL: ***"https://Support_Server.Braodcom.com:51111/auth"***

ENTERPRISE_DASHBOARD_SERVICE_NAME: "***Support_Server.Braodcom.com***"

ENTERPRISE_DASHBOARD_SERVICE_PORT: 1506

CONFIG_SERVER_URL: "http://eps-lisa-devtest-1006-config-server:8888"

REGISTRY_POD_PORT: 32020

Reference Links

- Getting started | Kubernetes
- Orientation and setup | Docker Documentation
- ***helm 2.17***
    - 6 Easy Steps to Install Helm Kubernetes Package Manager on Linux | CyberITHub


BROADCOM®