Temenos Data Lake

Docker – K8s Model bank

User Guide

Table of Contents

[Introduction 3](#_Toc56007355)

[Windows Installation setup 4](#_Toc56007356)

[**Install Docker Desktop** 4](#_Toc56007357)

[**Enabling Kubernetes on Docker Desktop** 5](#_Toc56007358)

[**Configuring Resources in Docker Desktop** 5](#_Toc56007359)

[**Helm Installation** 6](#_Toc56007360)

[**Download the TDE zip** 6](#_Toc56007361)

[**Configuring Values YAML file** 6](#_Toc56007362)

[**Create Docker and helm chart** 7](#_Toc56007363)

[**Accessing the TDE Portal** 8](#_Toc56007364)

[Linux Installation setup 9](#_Toc56007365)

[**Docker Installation** 9](#_Toc56007366)

[**Configure the Kubernetes repository** 9](#_Toc56007367)

[**Kubernetes Installation** 10](#_Toc56007368)

[**Disabling firewall** 10](#_Toc56007369)

[**Disabling Swap** 10](#_Toc56007370)

[**Deploying Kubernetes control plane to the master node** 11](#_Toc56007371)

[**Installing Helm** 11](#_Toc56007372)

[**Download the TDE zip** 12](#_Toc56007373)

[**Configuring Values YAML file** 12](#_Toc56007374)

[**Create Docker and helm chart** 13](#_Toc56007375)

[**Accessing the TDE Portal** 14](#_Toc56007376)

Introduction

Temenos Data Lake uses Docker and Kubernetes to setup local environment. The pre-requisite is to install Docker, helm and enable Kubernetes in either Windows or Linux Operation system. It internally uses Mysql as operational database and depend on MSSQL/ORACLE database to be available as target database platforms.

Windows Installation setup

Description: As this is not part of UTP, this package will be extracted under Xtras (Non UTP Products) of Model Bank Demo system.

Scope:

This package only ensures the end to end integration of the software works. Demo data and packaging improvements will be done in the future builds.

This guide covers the installation of TDL on existing MSSQL server.

Pre-Requisites:

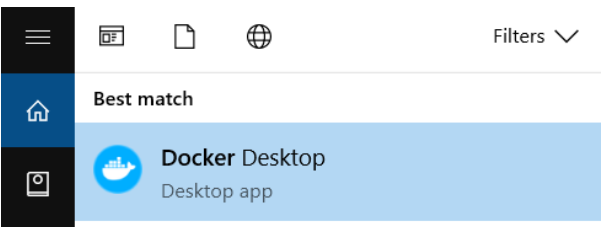
1. Transact should be up and running
2. DES should be setup either Docker or Standalone
3. Deploy irf-provider-container.war in Application server
4. Target database should be one of the below database stack such as ORACLE, MSSQL, Azure MSSQL and Postgres installed

**Install Docker Desktop**

Step 1: Download the Docker Desktop community version from Docker Hub using the link <https://hub.docker.com/?overlay=onboarding>

Step 2: Double click on “Docker Desktop Installer” executable file (.exe) to install the Docker Desktop. You can follow the instructions in “Installation Wizard” to accept the license, authorize the installer before proceeding with the installation.

Step 3: Once the installation is complete, you should be able to see “Docker Desktop” app, once you type “Docker” in your Windows search bar.



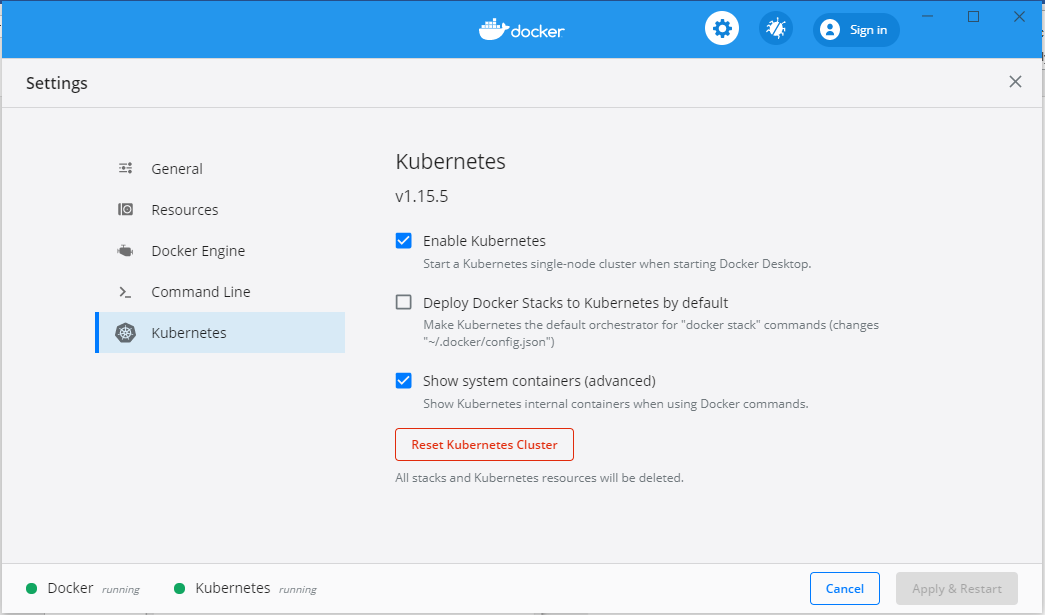
Step 4: Once click on “Docker Desktop” application, you should be able to see a whale icon in the status bar. In that case, the Docker Desktop is ready.

\

**Note:** If you have Oracle Virtual Box VMs, they will not work when Hyper-V is enabled.

**Enabling Kubernetes on Docker Desktop**

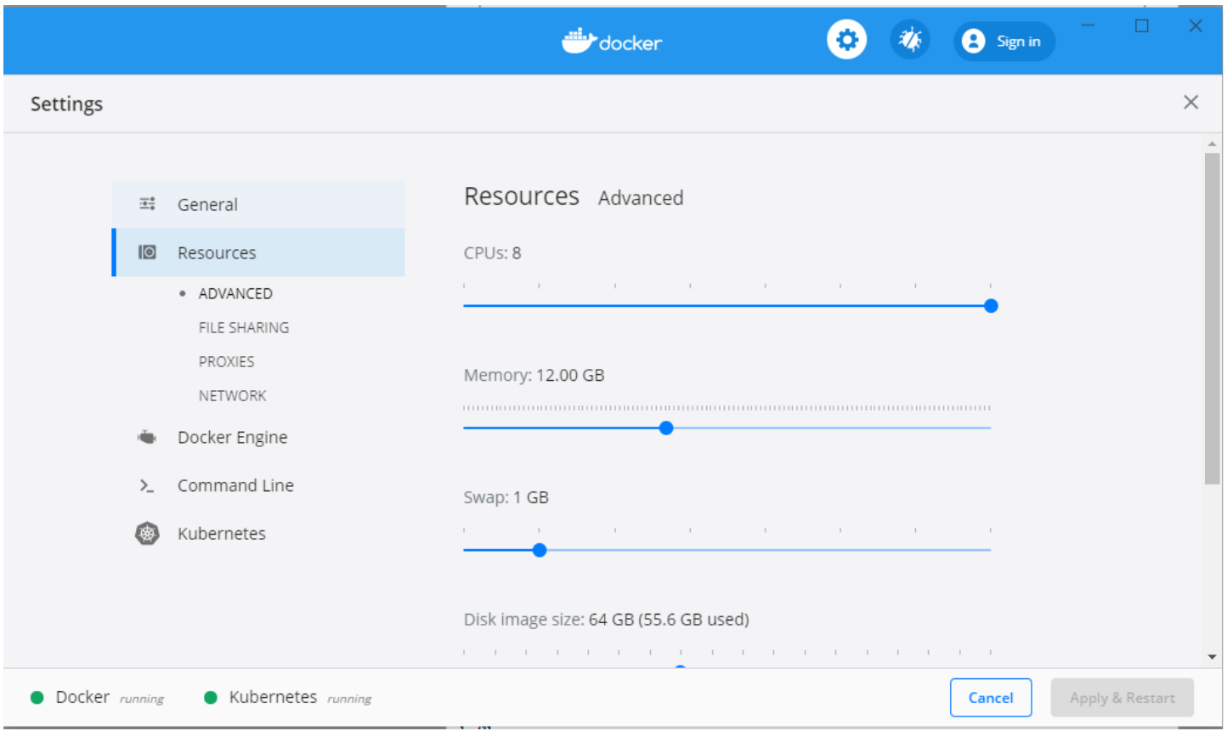
Once Docker is up and running you will see the Docker icon on your system tray. Right click on it and select settings. Within the settings form use the “Kubernetes” section to enable Kubernetes, then click on “Apply & Restart”. The installation will take a few minutes.



**Configuring Resources in Docker Desktop**

Please go to the path Docker Desktop -> Settings -> Advanced and set the CPUs and RAM appropriately according to the laptop/machine configuration.

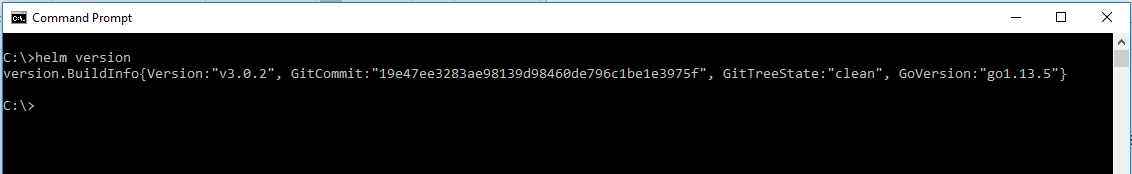
Remember that the Docker Desktop VM runs with the resources that are set. So, neither do not allocate very less RAM nor very high RAM also.



**Helm Installation**

Helm is the package manager for Kubernetes which allows you to install, upgrade and remove applications from your Kubernetes.

In order to install helm, download the Helm 3 executable for Windows from <https://github.com/helm/helm/releases>, save into a folder of your choice and add the executable’s file to your system’s “Path” variable. Test the installation by opening a command prompt and executing “helm version” which will return some basic details about your helm installation.

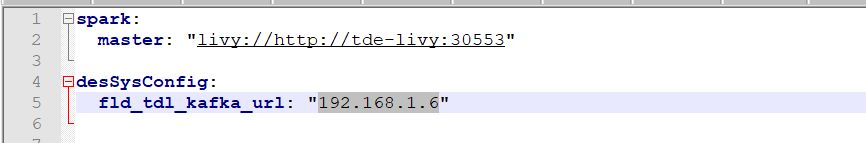


**Download the TDE zip**

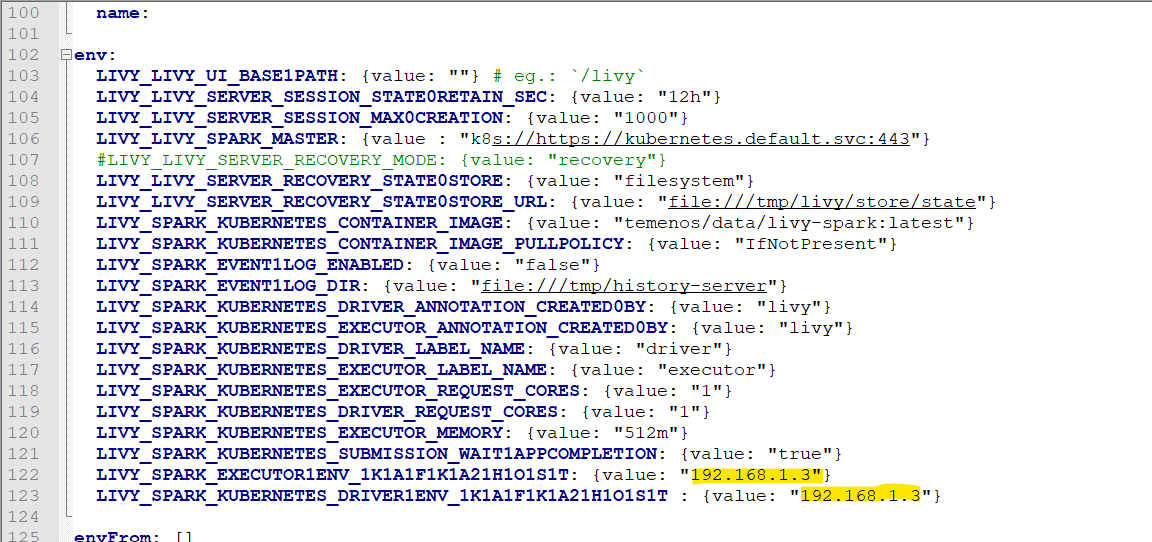
The DEV snapshot of TDE can be downloaded for 202008 release and unzipped into any directory of your choice<http://maven.temenosgroup.com/content/repositories/t24-snapshots/com/temenos/technology/data/Temenos_Data_Engineering/DEV.0.0-SNAPSHOT/><Temenos\_Data\_Engineering-DEV.0.0-202008>.zip

**Configuring Values YAML file**

1. Get inside the tdl-k8s directory and check the values.yaml file and values.admin-bootstrap.sample.yaml (under helm\tde-chart) contains the list of environment variables or values (as per Helm terminology) that are used in the Helm templates used for containerizing different TDE components. This file should be used as a template for your own configuration. Create a copy and rename it as “values.local.overrides.yaml” or use the existing onePlease refer to it and modify your “values.local.overrides.yaml” accordingly to the DES server ip address configured as below:



1. Please update the ip address in “values.yaml” where livy is running as below:



**Create Docker and helm chart**

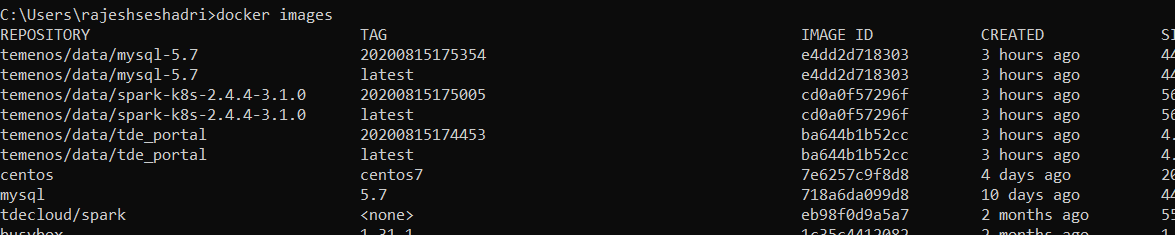
After overriding the properties, launch the docker tdl by running the batch file launch-tdl.bat

**Note**: Please open the launch-tdl.bat to verify the following highlighted line is specified in case of override defined

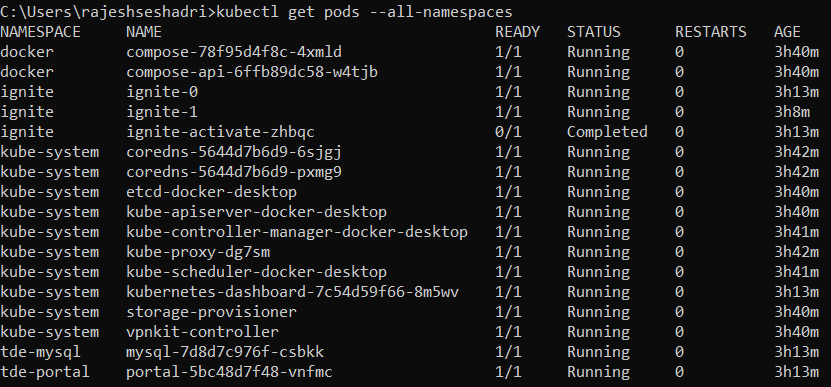
helm install tde . -f values.yaml -f values.admin-bootstrap.sample.yaml -f values.local.overrides.yaml --set clusterSizing.atlas=0

Run the script launch-tdl.bat. This may take some time.

1. To check the Docker images are started fine. Run this command “docker images”



1. To check Pods are started fine. Run this command “kubectl get pods --all-namespaces”



**Accessing the TDE Portal**

Use the following URL to access the TDE Portal:

<https://localhost:30443/>

You can log on as admin/admin or developer/560045

**Delete helm chart and Docker images**

Run the following command to delete the helm and docker images. **Use with caution, it will delete all the docker images,**

stop-tdl.bat

Linux Installation setup

Description: As this is not part of UTP, this package will be extracted under Xtras (Non UTP Products) of Model Bank Demo system.

Scope:

This package only ensures the end to end integration of the software works. Demo data and packaging improvements will be done in the future builds.

This guide covers the installation of TDL on existing MSSQL server.

Pre-Requisites:

1. Transact should be up and running
2. DES should be setup either Docker or Standalone
3. Deploy irf-provider-container.war in Application server
4. Target database should be one of the below database stack such as ORACLE, MSSQL, Azure MSSQL and Postgres installed

**Docker Installation**

Install docker from the RedHat repository and enable the service.

yum install -y yum-utils &&

yum-config-manager --enable rhel-7-server-extras-rpms &&

yum install docker -y &&

systemctl restart docker && systemctl enable docker

NB. Your RHEL installation needs to be registered before the command above can be issued.

**Configure the Kubernetes repository**

Kubernetes packages are not available in the default RHEL repositories but you can add them with the command below:

cat <<EOF > /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

EOF

**Kubernetes Installation**

Install kubeadm and enable the kubelet service

yum install -y kubeadm-1.16.10 kubectl-1.16.10 kubelet-1.16.10kubeadm-1.16.10 kubectl-systemctl restart kubelet && systemctl enable kubelet kubelet-1.16

**Disabling firewall**

systemctl firewalld

**Disabling Swap**

There is currently no support for swap space in Kubernetes containers. Disable swapping with the command below.

swapoff -a

Also, edit the /etc/fstab file by commenting the swap partition entry at the bottom

#

# /etc/fstab

# Created by anaconda on Tue Mar 20 17:54:11 2018

#

# Accessible filesystems, by reference, are maintained under '/dev/disk'

# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info

#

/dev/mapper/rhel-root / xfs defaults 0 0

UUID=cc2e2f0f-635f-464b-af84-a07f221a34bb /boot xfs defaults 0 0

/dev/mapper/rhel-home /home xfs defaults 0 0

# /dev/mapper/rhel-swap swap swap defaults -1 0

**Deploying Kubernetes control plane to the master node**

1. Initialise the master node

kubeadm init

1. Enable the current kubectl context for the current user.

mkdir -p $HOME/.kube &&

cp -i /etc/kubernetes/admin.conf $HOME/.kube/config &&

chown $(id -u):$(id -g) $HOME/.kube/config

1. Deploy the Weave Net plugin for overlay network plugin

export kubever=$(kubectl version | base64 | tr -d '\n')

kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$kubever"

You will need to ensure the default Weave Net IP range (10.32.0.0/12) does not overlap with your network. Please see [https://www.weave.works/blog/weave-net-kubernetes-integration](https://www.weave.works/blog/weave-net-kubernetes-integration%20) for further reference. An IP range calculator can be found at <http://jodies.de/ipcalc>. If correct, the command “kubectl get pods --all-namespaces” should produce the output below:

# kubectl get pods --all-namespaces

NAMESPACE NAME READY STATUS RESTARTS AGE

kube-system etcd-###.com 1/1 Running 0 2d

kube-system kube-apiserver-###.com 1/1 Running 0 2d

kube-system kube-controller-manager-###.com 1/1 Running 0 2d

kube-system kube-dns-86f4d74b45-wz554 3/3 Running 0 2d

kube-system kube-proxy-g94k8 1/1 Running 0 2d

kube-system kube-scheduler-###.com 1/1 Running 0 2d

kube-system weave-net-rszqs 2/2 Running 0 2d

1. [Optional]

If your cluster has no more than 3 nodes, you can enable the master for carrying worker duties with the following command

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

**Installing Helm**

1. Temenos Analytics releases for Kubernetes are deployed using Helm charts. Run the following commands to download & install Helm:

yum install -y wget &&

wget <https://storage.googleapis.com/kubernetes-helm/helm-v2.8.2-linux-amd64.tar.gz> &&

tar -xzf helm-v2.8.2-linux-amd64.tar.gz &&

mv linux-amd64/helm /usr/bin/helm

1. Switch to the manifests directory, create Helm’s service account with the command below and initialize.

kubectl apply -f kubernetes/cluster-principals/30.tiller-svc-acc.yaml &&

helm init --service-account tiller

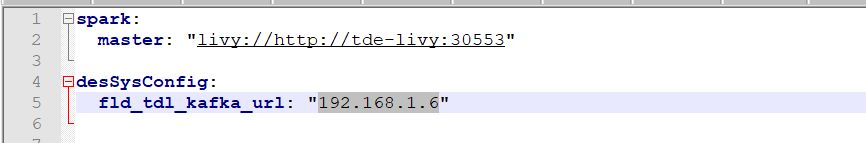
**Download the TDE zip**

The DEV snapshot of TDE can be downloaded for 202008 release and unzipped into any directory of your choice<http://maven.temenosgroup.com/content/repositories/t24-snapshots/com/temenos/technology/data/Temenos_Data_Engineering/DEV.0.0-SNAPSHOT/><Temenos\_Data\_Engineering-DEV.0.0-202008>.zip

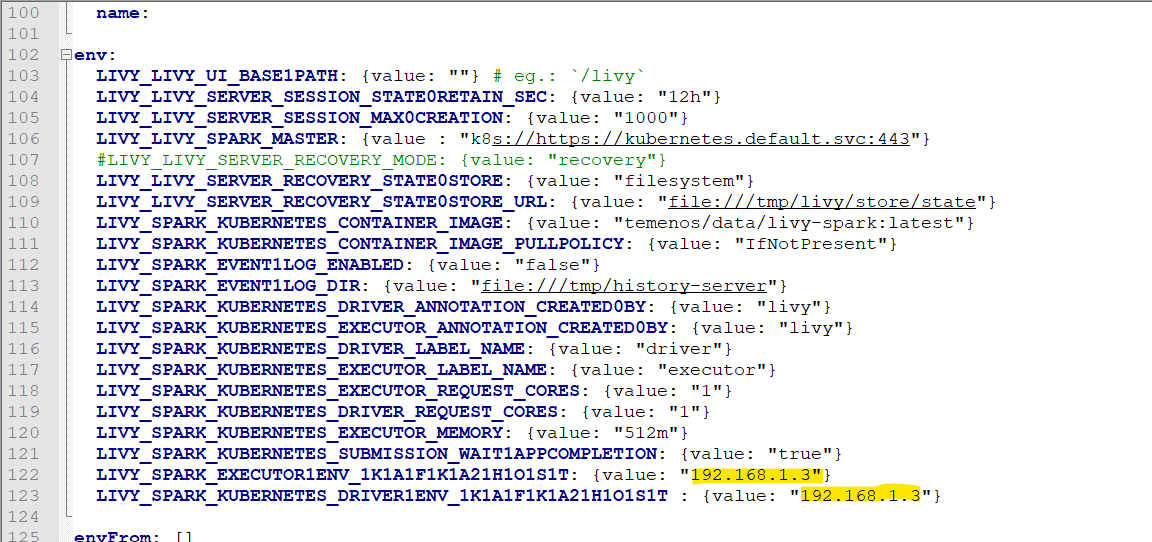
**Configuring Values YAML file**

Get inside the tdl-k8s directory, check the values.yaml file and values.admin-bootstrap.sample.yaml (under helm\tde-chart) contains the list of environment variables or values (as per Helm terminology) that are used in the Helm templates used for containerizing different TDE components. This file should be used as a template for your own configuration. Create a copy and rename it as “values.local.overrides.yaml”.

1. Get inside the tdl-k8s directory and check the values.yaml file and values.admin-bootstrap.sample.yaml (under helm\tde-chart) contains the list of environment variables or values (as per Helm terminology) that are used in the Helm templates used for containerizing different TDE components. This file should be used as a template for your own configuration. Create a copy and rename it as “values.local.overrides.yaml” or use the existing onePlease refer to it and modify your “values.local.overrides.yaml” accordingly to the DES server ip address configured as below:



1. Please update the ip address in “values.yaml” where livy is running as below:



**Create Docker and helm chart**

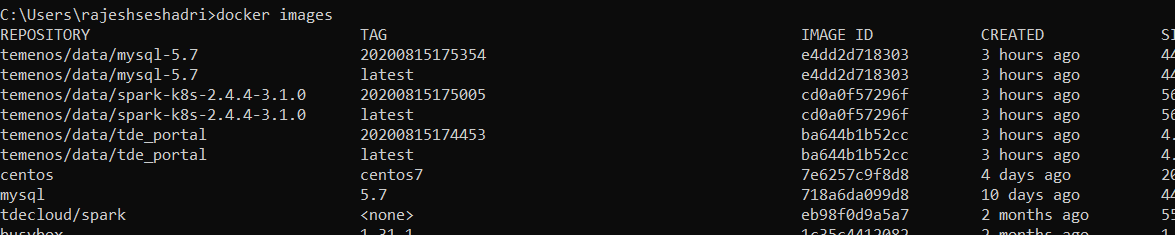
After overriding the properties, launch the docker tdl by running the batch file launch-tdl.bat

**Note**: Please open the launch-tdl.sh to verify the following highlighted line is specified in case of override defined

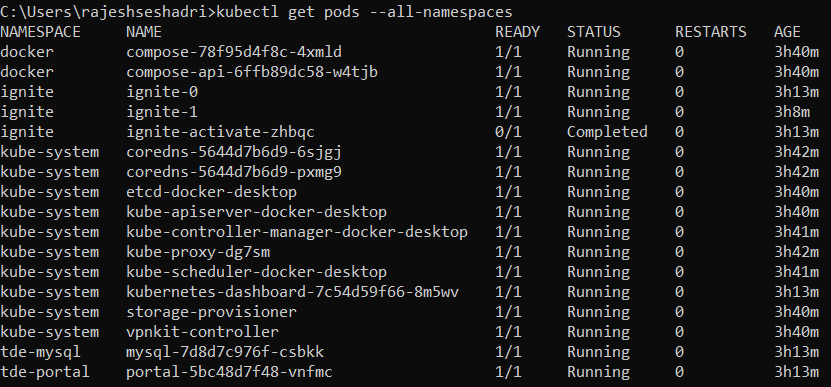
helm install tde . --set masternode=$hostname --set storage.azure.storageLocation="somelocation" -f values.yaml -f values.admin-bootstrap.sample.yaml -f values.local.overrides.test.yaml -f values.addon.linux-on-prem.yaml --set clusterSizing.atlas=0

Run the script launch-tdl.bat/launch-tdl.sh. This may take some time.

1. To check the Docker images are started fine. Run this command “docker images”



1. To check Pods are started fine. Run this command “kubectl get pods --all-namespaces”



**Accessing the TDE Portal**

Use the following URL to access the TDE Portal:

<https://localhost:30443/>

You can log on as admin/admin or developer/560045

**Delete helm chart and Docker images**

Run the following command to delete the helm and docker images. **Use with caution, it will delete all the docker images,**

stop-tdl.sh