

AKS - Azure Kubernetes

Cloud Computing

1/20/2020

Data Communications Labs

Tony Shen

AKS - Azure Kubernetes

Contents

Introduction	2
Prerequisites	2
Ubuntu Workstation	2
Setting up your Ubuntu workstation	2
Readying your Ubuntu workstation	2
Windows Workstation	4
Checking and installing required software	5
Creating K8 Cluster in Azure	5
Using K8 cluster in Azure	7
Conclusion.....	8
Appendices.....	8
Appendix A – Readying Ubuntu Workstation	8
Appendix B - Creating K8 cluster.....	17
Appendix D – Using K8 cluster	31
References	41

Revision History
01/20/20 Created
01/21/20 Revised
02/03/21 Revised as v4
02/19/23 Revised as v5

AKS - Azure Kubernetes

Introduction

Kubernetes (K8) is a popular Docker containerized app deployment and management orchestration system. K8 was originated in Google. Google Cloud Platform (GCP) offers K8 as a cloud service. Microsoft Azure does the same, and so does AWS. In Azure, K8 service is called AKS, standing for Azure Kubernetes Service.

In this write-up, we walk you through about how to set up a K8 cluster and how to run a container app in a K8 cluster in Azure cloud. In this walk-through we show you how to use Azure CLI to interact with AKS in getting the work done.

Prerequisites

For you to follow this walk-through hands-on, you need to have the following

1. Azure account and active subscription. If you don't already have one, you may sign up for a free account that comes with a free-tier subscription.
2. Docker Hub account. If you don't already have one, go to Docker Hub website to sign up for a free account.

After you have met these two prerequisites, keep your Azure account sign-in credentials and your Docker Hub sign-in credentials handy for further use.

Ubuntu Workstation

You need to have a workstation to work with AKS. OS of that workstation can be Windows, Mac, or Linux. In this walk-through, we choose Ubuntu Linux for our workstation.

Setting up your Ubuntu workstation

If you use a Windows 10 laptop as your daily work compute device, you may create your Ubuntu workstation using VMware Workstation that is installed and running in your laptop. If you don't have VMware workstation in your laptop, you can download and install a free trial version of that product for limited use. Once VMware Workstation is up and running in your laptop, you can create your Ubuntu workstation as a virtual machine and start using it. See Appendix A for details of how to do that.

Readying your Ubuntu workstation

After your Ubuntu workstation is up and running, you need to install a few software packages on it. To get this task done, follow and run the following commands in sequence.

Installing git

```
tshen@ubuntu:~$ sudo apt install git
```

Installing curl

```
tshen@ubuntu:~$ sudo apt install curl
```

AKS - Azure Kubernetes

Installing Azure CLI

```
tshen@ubuntu:~$ sudo apt install azure-cli
```

Installing kubectl for Azure AKS

```
tshen@ubuntu:~$ sudo az aks install-cli
```

Downloading client to "/usr/local/bin/kubectl" from "https://storage.googleapis.com/kubernetes-release/release/v1.20.0/bin/linux/amd64/kubectl"

Please ensure that /usr/local/bin is in your search PATH, so the `kubectl` command can be found.

```
tshen@ubuntu:~$ which kubectl
```

```
/usr/local/bin/kubectl
```

```
tshen@ubuntu:~$
```

Updating apt-get

```
tshen@ubuntu:~$ sudo apt-get update
```

Installing apt-get security related software

```
tshen@ubuntu:~$ sudo apt-get install \
```

```
> apt-transport-https \
```

```
> ca-certificates \
```

```
> curl \
```

```
> gnupg-agent \
```

```
> software-properties-common
```

```
tshen@ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
tshen@ubuntu:~$ sudo apt-key fingerprint 0EBFCD88
```

```
tshen@ubuntu:~$ sudo add-apt-repository \
```

```
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
```

```
> $(lsb_release -cs) \
```

```
> stable
```

```
tshen@ubuntu:~$ sudo apt-get update
```

Installing Docker CE engine and CLI

```
tshen@ubuntu:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Verifying Docker is working on our workstation

```
tshen@ubuntu:~$ sudo docker run hello-world
```

```
tshen@ubuntu:~$ sudo docker images
```

Installing Docker Compose

```
tshen@ubuntu:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
tshen@ubuntu:~$ sudo chmod +x /usr/local/bin/docker-compose
```

Getting a Docker container app to test Docker Compose with

```
tshen@ubuntu:~$ mkdir containers
```

```
tshen@ubuntu:~$ cd containers
```

```
tshen@ubuntu:~/containers$ git clone https://github.com/Azure-Samples/azure-voting-app-redis.git
```

```
tshen@ubuntu:~/containers$ ls -l
```

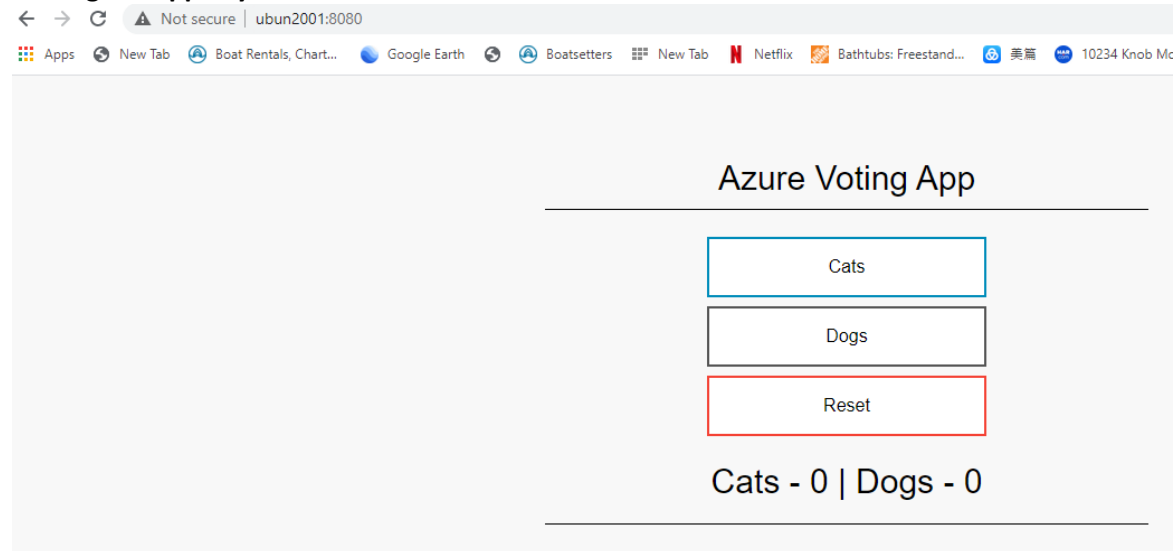
AKS - Azure Kubernetes

```
total 4
drwxrwxr-x 5 tshen tshen 4096 Dec 17 07:54 azure-voting-app-redis
tshen@ubuntu:~/containers$ cd azure-voting-app-redis
```

Starting up Docker Compose and run the app

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ docker-compose up -d
tshen@ubuntu:~/containers/azure-voting-app-redis$ curl localhost:8080
```

Testing the app in your browser



Shutting down Docker Composer

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ sudo docker-compose down
```

If you see the web page as shown above, congratulations, you come thus far to the point where your Ubuntu workstation is fully prepared and ready for you to use for AKS.

Appendix A – Readyng Ubuntu Workstation provides a detailed example with screen outputs, where you see what steps are involved for what needs to be done and how it is done as well as certain dependencies among those steps.

Windows Workstation

You can make your Windows 10 laptop as your workstation. This approach works if you have not installed any software that involves Docker and Kubernetes and that was customized for other clouds, such as AWS and GCP. If you have, your workstation will not work properly with AKS.

To prepare your Windows laptop or PC as your workstation, do the following
Open a CMD window as Administrator. At the prompt, check if you have required software installed. If not, find the online documentation for relevant software, install its Windows version and the latest stable release, configure it when required, and verify it again, by following pertinent instructions.

AKS - Azure Kubernetes

Checking and installing required software

```
C:> git --version
C:> az --version
C:> docker --version
C:> docker-compose --version
C:> curl --version
C:> wget --version
C:> kubectl version
```

Creating K8 Cluster in Azure

To create K8 cluster, you will need to create a number of resources. It is recommended that you have a naming standard (NS) in place to guide you in naming those resources before you start.

Table below lists names of resources that follow an example NS.

Item	Name	Description
Location	ussc	southcentralus (US South Central)
Resource Group	dcl-ussc-rg01	Resource group in South Central
K8 Cluster	dcl-ussc-rg01-k801	K8 cluster in South Central resource group
Azure Container Registry	dclusscrg01k801acr	Azure Container Registry in South Central resource group

By the example NS, a few elements constitute a resource name. Those elements indicate the owner, location, resource type, and resource name

1. Business name/Owner/Stakeholder name - dclp
2. Azure Location/Region - ussc (US South Central)
3. Resource Group - dcl-ussc-rg01
4. K8 cluster name - dcl-ussc-rg01-k801
5. Azure Container Registry name - dcl-ussc-rg01-k801-acr

To create a K8 cluster, follow these steps. Command(s) are provided for each step for your easy reference

1. Login from Azure CLI
\$ az login
2. List existing K8 clusters (Optional)
\$ az aks list
3. List cluster monitoring status (Optional)
\$ az provider show -n Microsoft.OperationsManagement -o table
\$ az provider show -n Microsoft.Operationallnsights -o table
4. Create a resource group for K8 cluster (Optional if you already have one)
\$ az group create -n dcl-ussc-rg01 -l southcentralus
5. Create K8 cluster in the resource group
\$ az aks create -g dcl-ussc-rg01 -n dcl-ussc-rg01-k8-01 --node-count 3 --enable-addons monitoring --generate-ssh-keys
6. Create ACR (Optional if you already have one)
\$ az acr create -g dcl-ussc-rg01 -n dclusscrg01k801acr --sku Basic

AKS - Azure Kubernetes

7. Update ACR with admin-enabled
\$ az acr update -n dclusscrg01k801acr --admin-enabled
8. Obtain ACR credentials
\$ az acr credential show -n dclusscrg01k801acr
9. Login ACR
\$ sudo az acr login -n dclusscrg01k801acr
10. Login K8 cluster
\$ az aks get-credentials -g dcl-ussc-rg01 -n dcl-ussc-rg01-k8-01
11. Check K8 cluster running status
\$ kubectl get nodes
\$ kubectl get services
\$ kubectl get namespaces
\$ kubectl get pods
12. Attach ACR to K8 cluster
\$ az aks update -n dcl-ussc-rg01-k8-01 -g dcl-ussc-rg01 --attach-acr dclusscrg01k801acr
13. Login Docker Hub (Assume that your container image is in Docker Hub)
\$ sudo docker login
14. List your images in Docker Hub
\$ sudo docker images
15. Obtain ACR login server name
dclusscrg01k801acr.azurecr.io (append ".azurecr.io" to registry name "dclusscrg01k801acr")
16. Tag your image in Docker Hub
\$ sudo docker tag mcr.microsoft.com/azuredocs/azure-vote-front:v1
dclusscrg01k801acr.azurecr.io/azure-vote-front:v1 (precede app image name with ACR login
server name)
\$ sudo docker images
17. List ACR images
\$ sudo az acr repository list -n dclusscrg01k801acr --output table
(It should be an empty return)
18. Push (Register) your image from Docker Hub to ACR
\$ sudo docker push dclusscrg01k801acr.azurecr.io/azure-vote-front:v1
19. Query ACR
\$ az acr list --query "[].{name:name}" --output tsv (list registry)
\$ az acr repository list -n dclusscrg01k801acr --output tsv (list repository in the registry)
\$ az acr repository show-tags --repository azure-vote-front -n dclusscrg01k801acr --output tsv
\$ az acr repository show -n dclusscrg01k801acr --image azure-vote-front:v1
20. Update your container app YAML by pointing to ACR registry, repository and tag
21. Deploy the app
\$ kubectl apply -f azure-vote-all-in-one-redis.yaml
22. Query services
\$ kubectl get service azure-vote-back
\$ kubectl get service azure-vote-front
23. Verify that the app is running
\$ curl <http://13.66.84.45> (external IP from service azure-vote-front)

Appendix B – Creating K8 Cluster provides a detailed example with screen outputs, where you see what steps are involved for what needs to be done and how it is done as well as certain dependencies among those steps.

Using K8 cluster in Azure

Commonly used commands are listed below

1. **Connecting to K8 cluster**
`$ az aks get-credentials --resource-group <rg name> --name <k8 cluster name>`
2. **Connecting to ACR**
`$ sudo az acr login --name <acr name>`
3. **Displaying repositories/images in ACR**
`$ sudo az acr repository list --name <acr name> -output table`
`$ az acr repository show -n <acr name> --image <repo name>:1`
`$ az acr repository show -n <acr name> --image <repo name>:2`
4. **Displaying cluster nodes**
`$ kubectl get nodes`
`$ kubectl describe nodes`
5. **Displaying pods**
`$ kubectl describe pods`
6. **Displaying all namespaces**
`$ kubectl get namespaces`
7. **Displaying all deployments**
`$ kubectl get deployments --all-namespaces=true`
8. **Displaying all pods running in all namespaces**
`$ kubectl get pods -l app=sampleapp --all-namespaces=true`
9. **Describing nodes**
`$ kubectl get nodes`
`$ kubectl describe nodes`
`$ kubectl describe <node name>`
10. **Listing node pool**
`$ az aks nodepool list -g <rg name> --cluster-name <k8 cluster name>`
11. **Listing data volumes**
`$ kubectl get sc`
12. **Executing shell commands in pod**
`$ kubectl get pods`
`$ kubectl describe pods`
`$ kubectl exec <pod name> -- ls -l`

AKS - Azure Kubernetes

```
$ kubectl exec <pod name> -- df
$ kubectl exec <pod name> -- cat /etc/hosts
```

13. Listing az extensions

```
$ az extension list
```

Appendix C – Using K8 Cluster provides a detailed example with screen outputs, where you see a number of commonly used commands for querying the cluster and outputs those commands produce. For a complete listing of available commands, please consult with Azure Kubernetes documentation and Kubernetes documentation.

Conclusion

In this write-up, we showed how to use Azure CLI to create and use Kubernetes cluster in Azure. Azure portal is equally capable of creating and managing K8 clusters. Since Azure portal is GUI, it is easier to use. The main drawback of using Azure portal is lack of means for automation. That's the reason that this write-up focuses on how to create and use Kubernetes in Azure using Azure CLI. Azure CLI commands can be easily scripted for automation.

Appendices

Appendix A – Readyng Ubuntu Workstation

How to create Ubuntu virtual machine using VMware Workstation

Installing Ubuntu 20 workstation (ubun2021)

```
tshen@ubuntu:/proc$ uname -a
Linux ubuntu 5.4.0-58-generic #64-Ubuntu SMP Wed Dec 9 08:16:25 UTC 2020 x86_64 x86_64 x86_64
GNU/Linux
tshen@ubuntu:/proc$ cat /etc/*release*
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=20.04
DISTRIB_CODENAME=focal
DISTRIB_DESCRIPTION="Ubuntu 20.04.1 LTS"
NAME="Ubuntu"
VERSION="20.04.1 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.1 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
```

AKS - Azure Kubernetes

```
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
tshen@ubuntu:/proc$
```

```
tshen@ubuntu:/proc$ free
      total    used    free   shared  buff/cache   available
Mem:   4002284 1165236  120688    2276  2716360  2545444
Swap:   945416    3596   941820
tshen@ubuntu:/proc$ cat /proc/cpuinfo | grep processor
processor    : 0
processor    : 1
processor    : 2
processor    : 3
tshen@ubuntu:/proc$
```

Installing SSHD

```
# apt install openssh-server
```

Configuring sudo

```
/etc/sudoers
# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) NOPASSWD:ALL
# See sudoers(5) for more information on "#include" directives:
#includedir /etc/sudoers.d
root@ubuntu:/home/tshen# id tshen
uid=1000(tshen) gid=1000(tshen)
groups=1000(tshen),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),131(lxd),132(sambas
hare)
root@ubuntu:/home/tshen#
```

Configuring static IP

```
root@ubuntu:/etc/netplan# cat /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
# renderer: NetworkManager
renderer: networkd
ethernets:
  ens33:
    dhcp4: no
    addresses:
      - 192.168.192.153/24
    gateway4: 192.168.192.2
    nameservers:
      addresses: [75.75.75.75, 75.75.76.76]
root@ubuntu:/etc/netplan#
```

AKS - Azure Kubernetes

```
# systemctl stop NetworkManager
# systemctl disable NetworkManager
# netplan apply
# ip addr
tshen@ubuntu:~$ ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
    qlen 1000
    link/ether 00:0c:29:38:e7:14 brd ff:ff:ff:ff:ff:ff
    inet 192.168.192.153/24 brd 192.168.192.255 scope global ens33
        valid_lft forever preferred_lft forever
    inet6 fe80::59cf:da04:87af:43b4/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
tshen@ubuntu:~$
```

Installing Docker Engine server for Linux using docker registry

Reference: <https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository>

Note:

1. Docker Engine (DE) is distributed via client and server version
2. DE client is Docker Desktop (DD)
3. DD is available for Mac and Windows only, not for Linux yet
4. To install DE for Linux, install DE for Server for your chosen Linux platform
5. DE Server for Linux, however, does not include docker-compose. Docker Compose, therefore, needs to be installed separately after installing DE server for Linux

```
tshen@ubuntu:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [109 kB]
Hit:2 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [101 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [24.3 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [56.5 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [699 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [263 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal-updates/main DEP-11 48x48 Icons [55.6 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-updates/universe i386 Packages [523 kB]
Get:11 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [705 kB]
Get:12 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [205 kB]
Get:13 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [2,468 B]
```

AKS - Azure Kubernetes

```
Get:14 http://us.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [1,768 B]
Fetched 2,859 kB in 3s (1,008 kB/s)
Reading package lists... Done
tshen@ubuntu:~$ sudo apt-get install \
> apt-transport-https \
> ca-certificates \
> curl \
> gnupg-agent \
> software-properties-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20201027ubuntu0.20.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.68.0-1ubuntu2.4).
The following NEW packages will be installed:
  apt-transport-https gnupg-agent
The following packages will be upgraded:
  python3-software-properties software-properties-common software-properties-gtk
3 upgraded, 2 newly installed, 0 to remove and 176 not upgraded.
Need to get 107 kB of archives.
After this operation, 207 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 apt-transport-https all 2.0.2ubuntu0.2 [1,708 B]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 gnupg-agent all 2.2.19-3ubuntu2 [5,236 B]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 software-properties-common all 0.98.9.3 [10.6 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 software-properties-gtk all 0.98.9.3 [64.0 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 python3-software-properties all 0.98.9.3 [25.3 kB]
Fetched 107 kB in 0s (287 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 199922 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.0.2ubuntu0.2_all.deb ...
Unpacking apt-transport-https (2.0.2ubuntu0.2) ...
Selecting previously unselected package gnupg-agent.
Preparing to unpack .../gnupg-agent_2.2.19-3ubuntu2_all.deb ...
Unpacking gnupg-agent (2.2.19-3ubuntu2) ...
Preparing to unpack .../software-properties-common_0.98.9.3_all.deb ...
Unpacking software-properties-common (0.98.9.3) over (0.98.9.2) ...
Preparing to unpack .../software-properties-gtk_0.98.9.3_all.deb ...
Unpacking software-properties-gtk (0.98.9.3) over (0.98.9.2) ...
Preparing to unpack .../python3-software-properties_0.98.9.3_all.deb ...
Unpacking python3-software-properties (0.98.9.3) over (0.98.9.2) ...
Setting up apt-transport-https (2.0.2ubuntu0.2) ...
```

AKS - Azure Kubernetes

```
Setting up python3-software-properties (0.98.9.3) ...
Setting up gnupg-agent (2.2.19-3ubuntu2) ...
Setting up software-properties-common (0.98.9.3) ...
Setting up software-properties-gtk (0.98.9.3) ...
Processing triggers for dbus (1.12.16-2ubuntu2.1) ...
Processing triggers for shared-mime-info (1.15-1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for libglib2.0-0:amd64 (2.64.3-1~ubuntu20.04.1) ...
Processing triggers for man-db (2.9.1-1) ...
tshen@ubuntu:~$
tshen@ubuntu:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
OK
tshen@ubuntu:~$
tshen@ubuntu:~$ sudo apt-key fingerprint 0EBFCD88
pub  rsa4096 2017-02-22 [SCEA]
     9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
uid      [ unknown] Docker Release (CE deb) <docker@docker.com>
sub  rsa4096 2017-02-22 [S]
tshen@ubuntu:~$

tshen@ubuntu:~$ sudo add-apt-repository \
> "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
> $(lsb_release -cs) \
> stable"
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:2 https://download.docker.com/linux/ubuntu focal InRelease [36.2 kB]
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:6 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [5,637 B]
Fetched 41.8 kB in 1s (49.1 kB/s)
Reading package lists... Done
tshen@ubuntu:~$

tshen@ubuntu:~$ sudo apt-get update
Hit:1 http://us.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 https://download.docker.com/linux/ubuntu focal InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:5 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
tshen@ubuntu:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

AKS - Azure Kubernetes

The following additional packages will be installed:

docker-ce-rootless-extras pigz slirp4netns

Suggested packages:

aufs-tools cgroupfs-mount | cgroup-lite

The following NEW packages will be installed:

containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras pigz slirp4netns

0 upgraded, 6 newly installed, 0 to remove and 176 not upgraded.

Need to get 103 MB of archives.

After this operation, 450 MB of additional disk space will be used.

Do you want to continue? [Y/n] Y

Get:1 <https://download.docker.com/linux/ubuntu/focal/stable/amd64/containerd.io> amd64 1.4.3-1 [28.1 MB]

Get:2 <http://us.archive.ubuntu.com/ubuntu/focal/universe/amd64/pigz> amd64 2.4-1 [57.4 kB]

Get:3 <http://us.archive.ubuntu.com/ubuntu/focal/universe/amd64/slirp4netns> amd64 0.4.3-1 [74.3 kB]

Get:4 <https://download.docker.com/linux/ubuntu/focal/stable/amd64/docker-ce-cli> amd64 5:20.10.1~3-0~ubuntu-focal [41.4 MB]

Get:5 <https://download.docker.com/linux/ubuntu/focal/stable/amd64/docker-ce> amd64 5:20.10.1~3-0~ubuntu-focal [24.8 MB]

Get:6 <https://download.docker.com/linux/ubuntu/focal/stable/amd64/docker-ce-rootless-extras> amd64 5:20.10.1~3-0~ubuntu-focal [8,908 kB]

Fetch 103 MB in 4s (24.4 MB/s)

Selecting previously unselected package pigz.

(Reading database ... 199930 files and directories currently installed.)

Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...

Unpacking pigz (2.4-1) ...

Selecting previously unselected package containerd.io.

Preparing to unpack .../1-containerd.io_1.4.3-1_amd64.deb ...

Unpacking containerd.io (1.4.3-1) ...

Selecting previously unselected package docker-ce-cli.

Preparing to unpack .../2-docker-ce-cli_5%3a20.10.1~3-0~ubuntu-focal_amd64.deb ...

Unpacking docker-ce-cli (5:20.10.1~3-0~ubuntu-focal) ...

Selecting previously unselected package docker-ce.

Preparing to unpack .../3-docker-ce_5%3a20.10.1~3-0~ubuntu-focal_amd64.deb ...

Unpacking docker-ce (5:20.10.1~3-0~ubuntu-focal) ...

Selecting previously unselected package docker-ce-rootless-extras.

Preparing to unpack .../4-docker-ce-rootless-extras_5%3a20.10.1~3-0~ubuntu-focal_amd64.deb ...

Unpacking docker-ce-rootless-extras (5:20.10.1~3-0~ubuntu-focal) ...

Selecting previously unselected package slirp4netns.

Preparing to unpack .../5-slirp4netns_0.4.3-1_amd64.deb ...

Unpacking slirp4netns (0.4.3-1) ...

Setting up slirp4netns (0.4.3-1) ...

Setting up containerd.io (1.4.3-1) ...

Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service →
/lib/systemd/system/containerd.service.

Setting up docker-ce-cli (5:20.10.1~3-0~ubuntu-focal) ...

Setting up pigz (2.4-1) ...

Setting up docker-ce (5:20.10.1~3-0~ubuntu-focal) ...

AKS - Azure Kubernetes

```
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service →  
/lib/systemd/system/docker.service.
```

```
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket →  
/lib/systemd/system/docker.socket.
```

```
Setting up docker-ce-rootless-extras (5:20.10.1~3-0~ubuntu-focal) ...
```

```
Processing triggers for man-db (2.9.1-1) ...
```

```
Processing triggers for systemd (245.4-4ubuntu3.2) ...
```

```
tshen@ubuntu:~$
```

```
tshen@ubuntu:~$ sudo docker run hello-world
```

```
Unable to find image 'hello-world:latest' locally
```

```
latest: Pulling from library/hello-world
```

```
0e03bdcc26d7: Pull complete
```

```
Digest: sha256:1a523af650137b8acddaed439c17d684df61ee4d74feac151b5b337bd29e7eec
```

```
Status: Downloaded newer image for hello-world:latest
```

```
Hello from Docker!
```

```
This message shows that your installation appears to be working correctly.
```

```
To generate this message, Docker took the following steps:
```

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(amd64)
3. The Docker daemon created a new container from that image which runs the
executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
to your terminal.

```
To try something more ambitious, you can run an Ubuntu container with:
```

```
$ docker run -it ubuntu bash
```

```
Share images, automate workflows, and more with a free Docker ID:
```

```
https://hub.docker.com/
```

```
For more examples and ideas, visit:
```

```
https://docs.docker.com/get-started/
```

```
tshen@ubuntu:~$ sudo docker images
```

```
REPOSITORY TAG IMAGE ID CREATED SIZE  
hello-world latest bf756fb1ae65 11 months ago 13.3kB  
tshen@ubuntu:~$
```

Installing Docker Compose

```
tshen@ubuntu:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
% Total % Received % Xferd Average Speed Time Time Time Current  
Dload Upload Total Spent Left Speed  
100 651 100 651 0 0 2465 0 --:--:-- --:--:-- --:--:-- 2465
```

AKS - Azure Kubernetes

```
100 11.6M 100 11.6M 0 0 7374k 0 0:00:01 0:00:01 --:--: 13.0M
tshen@ubuntu:~$ which docker-compose
tshen@ubuntu:~$ ls -l /usr/local/bin/docker-compose
-rw-r--r-- 1 root root 12218968 Dec 17 07:43 /usr/local/bin/docker-compose
tshen@ubuntu:~$ sudo chmod +x /usr/local/bin/docker-compose
tshen@ubuntu:~$ which docker-compose
/usr/local/bin/docker-compose
tshen@ubuntu:~$
```

Getting an app to run with Docker Composer

```
tshen@ubuntu:~$ mkdir containers
tshen@ubuntu:~$ cd containers
tshen@ubuntu:~/containers$ git clone https://github.com/Azure-Samples/azure-voting-app-redis.git
tshen@ubuntu:~/containers$ ls -l
total 4
drwxrwxr-x 5 tshen tshen 4096 Dec 17 07:54 azure-voting-app-redis
tshen@ubuntu:~/containers$ cd azure-voting-app-redis
tshen@ubuntu:~/containers/azure-voting-app-redis$ ls -l
total 24
drwxrwxr-x 3 tshen tshen 4096 Dec 17 07:54 azure-vote
-rw-rw-r-- 1 tshen tshen 1532 Dec 17 07:54 azure-vote-all-in-one-redis.yaml
-rw-rw-r-- 1 tshen tshen 433 Dec 17 07:54 docker-compose.yaml
drwxrwxr-x 2 tshen tshen 4096 Dec 17 07:54 jenkins-tutorial
-rw-rw-r-- 1 tshen tshen 1162 Dec 17 07:54 LICENSE
-rw-rw-r-- 1 tshen tshen 1806 Dec 17 07:54 README.md
tshen@ubuntu:~/containers/azure-voting-app-redis$ cat docker-compose.yaml
version: '3'
services:
  azure-vote-back:
    image: mcr.microsoft.com/oss/bitnami/redis:6.0.8
    container_name: azure-vote-back
    environment:
      ALLOW_EMPTY_PASSWORD: "yes"
    ports:
      - "6379:6379"

  azure-vote-front:
    build: ./azure-vote
    image: mcr.microsoft.com/azuredocs/azure-vote-front:v1
    container_name: azure-vote-front
    environment:
      REDIS: azure-vote-back
    ports:
      - "8080:80"
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Running the app locally with Docker Compose

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ docker-compose up -d
```


AKS - Azure Kubernetes

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mcr.microsoft.com/azuredocs/azure-vote-front	v1	4ea526af5d15	14 minutes ago	944MB
mcr.microsoft.com/oss/bitnami/redis	6.0.8	3a54a920bb6c	2 months ago	103MB
tiangolo/uwsgi-nginx-flask	python3.6	a16ce562e863	4 months ago	944MB
hello-world	latest	bf756fb1ae65	11 months ago	13.3kB

```
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

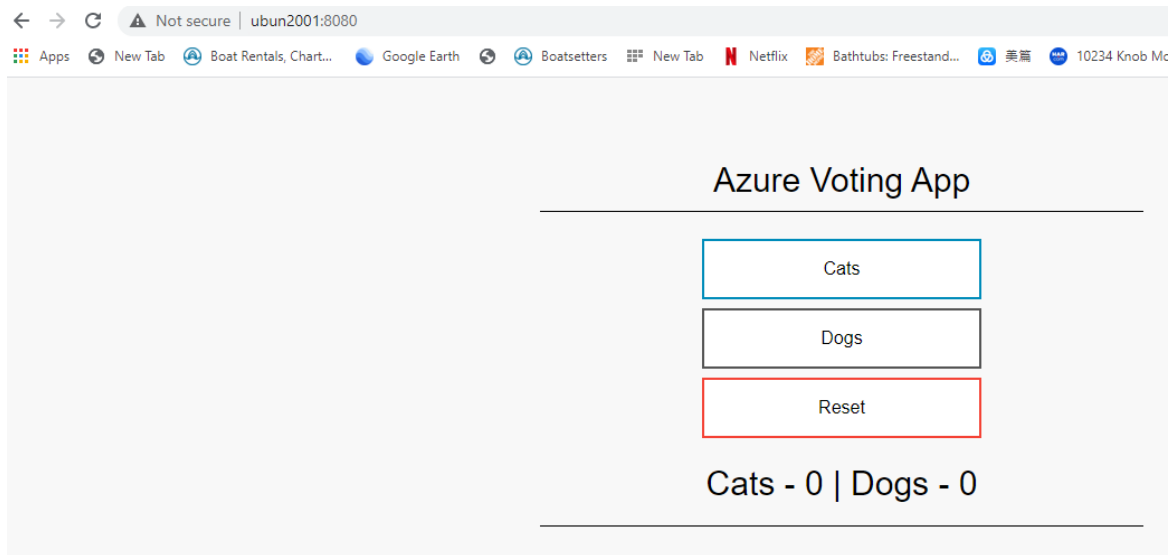
```
tshen@ubuntu:~/containers/azure-voting-app-redis$ curl localhost:8080
```

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <link rel="stylesheet" type="text/css" href="/static/default.css">
  <title>Azure Voting App</title>

  <script language="JavaScript">
    function send(form){
    }
  </script>

</head>
<body>
  <div id="container">
    <form id="form" name="form" action="/" method="post"><center>
      <div id="logo">Azure Voting App</div>
      <div id="space"></div>
      <div id="form">
        <button name="vote" value="Cats" onclick="send()" class="button button1">Cats</button>
        <button name="vote" value="Dogs" onclick="send()" class="button button2">Dogs</button>
        <button name="vote" value="reset" onclick="send()" class="button button3">Reset</button>
        <div id="space"></div>
        <div id="space"></div>
        <div id="results"> Cats - 0 | Dogs - 0 </div>
      </form>
    </div>
  </div>
</body>
</html>tshen@ubuntu:~/containers/azure-voting-app-redis$
```

AKS - Azure Kubernetes



Shutting down Docker Compose

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ sudo docker-compose down
Stopping azure-vote-back ... done
Stopping azure-vote-front ... done
Removing azure-vote-back ... done
Removing azure-vote-front ... done
Removing network azure-voting-app-redis_default
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Appendix B - Creating K8 cluster

AZ Login

```
tshen@ubuntu:~$ az login
```

To sign in, use a web browser to open the page <https://microsoft.com/devicelogin> and enter the code F8TL6DUAJ to authenticate.

```
[
  {
    "cloudName": "AzureCloud",
    "id": "<omitted>",
    "isDefault": false,
    "name": "Azure subscription 1",
    "state": "Enabled",
    "tenantId": "<omitted>",
    "user": {
      "name": "<omitted>",
      "type": "user"
    }
  },
  {
    "cloudName": "AzureCloud",
```

AKS - Azure Kubernetes

```
"id": "<omitted>",
"isDefault": true,
"name": "<omitted>",
"state": "Enabled",
"tenantId": "<omitted>",
"user": {
  "name": "<omitted>",
  "type": "user"
}
}
]
tshen@ubuntu:~$
```

Checking AKS cluster status

```
tshen@ubuntu:~$ az aks list
[]
```

Checking Cluster monitor status

```
tshen@ubuntu:~$ az provider show -n Microsoft.OperationsManagement -o table
Namespace      RegistrationPolicy  RegistrationState
-----
Microsoft.OperationsManagement  RegistrationRequired  Registered
tshen@ubuntu:~$ az provider show -n Microsoft.Operationallnsights -o table
Namespace      RegistrationPolicy  RegistrationState
-----
Microsoft.Operationallnsights  RegistrationRequired  Registered
tshen@ubuntu:~$
```

Creating a resource group for K8 cluster

```
tshen@ubuntu:~$ az group create --name dcl-ussc-rg01 --location southcentralus
{
  "id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourceGroups/dcl-ussc-rg01",
  "location": "southcentralus",
  "managedBy": null,
  "name": "dcl-ussc-rg01",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
tshen@ubuntu:~$
```

Creating K8 cluster in the RG

```
tshen@ubuntu:~$ az aks create -g dcl-ussc-rg01 -n dcl-ussc-rg01-k8-01 --node-count 3 --enable-addons
monitoring --generate-ssh-keys
{
```

AKS - Azure Kubernetes

```
"aadProfile": null,
"addonProfiles": {
  "KubeDashboard": {
    "config": null,
    "enabled": false,
    "identity": null
  },
  "omsagent": {
    "config": {
      "logAnalyticsWorkspaceResourceID": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourcegroups/defaultresourcegroup-scus/providers/microsoft.operationalinsights/workspaces/defaultworkspace-6d6c2372-faed-4d8f-86d1-3019ed20ea44-scus"
    },
    "enabled": true,
    "identity": null
  }
},
"agentPoolProfiles": [
  {
    "availabilityZones": null,
    "count": 3,
    "enableAutoScaling": null,
    "enableNodePublicIp": null,
    "maxCount": null,
    "maxPods": 110,
    "minCount": null,
    "name": "nodepool1",
    "nodeLabels": null,
    "nodeTaints": null,
    "orchestratorVersion": "1.18.14",
    "osDiskSizeGb": 128,
    "osType": "Linux",
    "provisioningState": "Succeeded",
    "scaleSetEvictionPolicy": null,
    "scaleSetPriority": null,
    "tags": null,
    "type": "VirtualMachineScaleSets",
    "vmSize": "Standard_DS2_v2",
    "vnetSubnetId": null
  }
],
"apiServerAccessProfile": null,
"diskEncryptionSetId": null,
"dnsPrefix": "dcl-ussc-r-dcl-ussc-rg01-6d6c23",
"enablePodSecurityPolicy": null,
"enableRbac": true,
"fqdn": "dcl-ussc-r-dcl-ussc-rg01-6d6c23-6ac570e4.hcp.southcentralus.azmk8s.io",
```

AKS - Azure Kubernetes

```
"id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourcegroups/dcl-ussc-rg01/providers/Microsoft.ContainerService/managedClusters/dcl-ussc-rg01-k8-01",
"identity": null,
"identityProfile": null,
"kubernetesVersion": "1.18.14",
"linuxProfile": {
  "adminUsername": "azureuser",
  "ssh": {
    "publicKeys": [
      {
        "keyData": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACruAPKjc0WqrXqM3/zmF2aal7BrNeemBs0PGC0HSSYL/SwCM1
3qivwBV7XtcKYwMk8Qis77i8qiH9qL4UiTMgxJ+CD1RCK7+aDTblEXIBph+2ambSmSEVw1KOLetF/U+a9YIP
5PxRoS/IOgkYsDpLV80RAffNpk6ka/DQAlqbx8UHospSPK1OAnfCEJ0QmoxGu0X5OeOJKxFEBPqEUJyPB6Fu
eUew3OyOMf1tJpuAYHG1mS0qYwCshbRbFnsPKW/f12jd73fDTTU3y1lzbw9QZp3IZYOMk0MX+vUWby9q
1JeDp6qgQSL7w/vjtbENMkAl0JC8NGQiWVlWYlXqRRLn"
      }
    ]
  },
},
"location": "southcentralus",
"maxAgentPools": 10,
"name": "dcl-ussc-rg01-k8-01",
"networkProfile": {
  "dnsServiceIp": "10.0.0.10",
  "dockerBridgeCidr": "172.17.0.1/16",
  "loadBalancerProfile": {
    "allocatedOutboundPorts": null,
    "effectiveOutboundIps": [
      {
        "id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourceGroups/MC_dcl-ussc-rg01_dcl-ussc-rg01-k8-01_southcentralus/providers/Microsoft.Network/publicIPAddresses/211c0079-27e9-471b-8c3b-b8a0985d7f05",
        "resourceGroup": "MC_dcl-ussc-rg01_dcl-ussc-rg01-k8-01_southcentralus"
      }
    ]
  },
},
"idleTimeoutInMinutes": null,
"managedOutboundIps": {
  "count": 1
},
"outboundIpPrefixes": null,
"outboundIps": null
},
"loadBalancerSku": "Standard",
"networkPlugin": "kubenet",
"networkPolicy": null,
"outboundType": "loadBalancer",
"podCidr": "10.244.0.0/16",
```

AKS - Azure Kubernetes

```
"serviceCidr": "10.0.0.0/16"
},
"nodeResourceGroup": "MC_dcl-ussc-rg01_dcl-ussc-rg01-k8-01_southcentralus",
"privateFqdn": null,
"provisioningState": "Succeeded",
"resourceGroup": "dcl-ussc-rg01",
"servicePrincipalProfile": {
  "clientId": "0818abf8-2b47-43eb-a7db-6d9ed4b890d9",
  "secret": null
},
"tags": null,
"type": "Microsoft.ContainerService/ManagedClusters",
"windowsProfile": null
}
tshen@ubuntu:~$ az aks list --query "[].{name:name}" --output table
Name
-----
dcl-ussc-rg01-k8-01
tshen@ubuntu:~$
```

Creating ACR

```
tshen@ubuntu:~$ az acr create -g dcl-ussc-rg01 -n dclusscrg01k801acr --sku Basic
{
  "adminUserEnabled": false,
  "creationDate": "2021-01-26T17:01:21.892134+00:00",
  "id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourceGroups/dcl-ussc-rg01/providers/Microsoft.ContainerRegistry/registries/dclusscrg01k801acr",
  "location": "southcentralus",
  "loginServer": "dclusscrg01k801acr.azurecr.io",
  "name": "dclusscrg01k801acr",
  "networkRuleSet": null,
  "policies": {
    "quarantinePolicy": {
      "status": "disabled"
    },
    "retentionPolicy": {
      "days": 7,
      "lastUpdatedTime": "2021-01-26T17:01:23.233585+00:00",
      "status": "disabled"
    },
    "trustPolicy": {
      "status": "disabled",
      "type": "Notary"
    }
  },
  "provisioningState": "Succeeded",
  "resourceGroup": "dcl-ussc-rg01",
}
```

AKS - Azure Kubernetes

```
"sku": {
  "name": "Basic",
  "tier": "Basic"
},
"status": null,
"storageAccount": null,
"tags": {},
"type": "Microsoft.ContainerRegistry/registries"
}
tshen@ubuntu:~$
```

Updating ACR

```
tshen@ubuntu:~$ az acr update -n dclusscrg01k801acr --admin-enabled
{
  "adminUserEnabled": true,
  "creationDate": "2021-01-26T17:01:21.892134+00:00",
  "id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourceGroups/dcl-ussc-rg01/providers/Microsoft.ContainerRegistry/registries/dclusscrg01k801acr",
  "location": "southcentralus",
  "loginServer": "dclusscrg01k801acr.azurecr.io",
  "name": "dclusscrg01k801acr",
  "networkRuleSet": null,
  "policies": {
    "quarantinePolicy": {
      "status": "disabled"
    },
    "retentionPolicy": {
      "days": 7,
      "lastUpdatedTime": "2021-01-26T17:01:23.233585+00:00",
      "status": "disabled"
    },
    "trustPolicy": {
      "status": "disabled",
      "type": "Notary"
    }
  },
  "provisioningState": "Succeeded",
  "resourceGroup": "dcl-ussc-rg01",
  "sku": {
    "name": "Basic",
    "tier": "Basic"
  },
  "status": null,
  "storageAccount": null,
  "tags": {},
  "type": "Microsoft.ContainerRegistry/registries"
}
tshen@ubuntu:~$
```

AKS - Azure Kubernetes

Obtaining ACR credentials

```
tshen@ubuntu:~$ az acr credential show -n dclusscrg01k801acr
{
  "passwords": [
    {
      "name": "password",
      "value": "<omitted>"
    },
    {
      "name": "password2",
      "value": "<omitted>"
    }
  ],
  "username": "<omitted>"
}
tshen@ubuntu:~$
```

Login ACR

```
tshen@ubuntu:~$ sudo az acr login -n <omitted>
Unable to get AAD authorization tokens with message: Please run 'az login' to setup account.
Unable to get admin user credentials with message: Please run 'az login' to setup account.
Username: dclusscrg01k801acr
Password:
Login Succeeded
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

tshen@ubuntu:~$
```

Login K8 cluster

```
tshen@ubuntu:~$ az aks get-credentials -g dcl-ussc-rg01 -n dcl-ussc-rg01-k8-01
Merged "dcl-ussc-rg01-k8-01" as current context in /home/tshen/.kube/config
tshen@ubuntu:~$
```

Checking K8 cluster running status

```
tshen@ubuntu:~$ kubectl get nodes
NAME                                STATUS  ROLES  AGE  VERSION
aks-nodepool1-64776267-vmss000000  Ready  agent  26m  v1.18.14
aks-nodepool1-64776267-vmss000001  Ready  agent  26m  v1.18.14
aks-nodepool1-64776267-vmss000002  Ready  agent  26m  v1.18.14
tshen@ubuntu:~$ kubectl get services
NAME      TYPE        CLUSTER-IP  EXTERNAL-IP  PORT(S)    AGE
kubernetes  ClusterIP  10.0.0.1    <none>       443/TCP    27m
```


AKS - Azure Kubernetes

```
tshen@ubuntu:~$ kubectl get namespaces
NAME          STATUS AGE
default       Active 27m
kube-node-lease Active 28m
kube-public   Active 28m
kube-system   Active 28m
tshen@ubuntu:~$ kubectl get pods
No resources found in default namespace.
tshen@ubuntu:~$
```

Attaching ACR to K8 cluster

```
tshen@ubuntu:~$ az aks update -n dcl-ussc-rg01-k8-01 -g dcl-ussc-rg01 --attach-acr dclusscrg01k801acr
{
  "aadProfile": null,
  "addonProfiles": {
    "KubeDashboard": {
      "config": null,
      "enabled": false,
      "identity": null
    },
    "omsagent": {
      "config": {
        "logAnalyticsWorkspaceResourceID": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourcegroups/defaultresourcegroup-scus/providers/microsoft.operationalinsights/workspaces/defaultworkspace-6d6c2372-faed-4d8f-86d1-3019ed20ea44-scus"
      },
      "enabled": true,
      "identity": null
    }
  },
  "agentPoolProfiles": [
    {
      "availabilityZones": null,
      "count": 3,
      "enableAutoScaling": null,
      "enableNodePublicIp": null,
      "maxCount": null,
      "maxPods": 110,
      "minCount": null,
      "name": "nodepool1",
      "nodeLabels": null,
      "nodeTaints": null,
      "orchestratorVersion": "1.18.14",
      "osDiskSizeGb": 128,
      "osType": "Linux",
      "provisioningState": "Succeeded",
      "scaleSetEvictionPolicy": null,

```

AKS - Azure Kubernetes

```
"scaleSetPriority": null,
"tags": null,
"type": "VirtualMachineScaleSets",
"vmSize": "Standard_DS2_v2",
"vnetSubnetId": null
}
],
"apiServerAccessProfile": null,
"diskEncryptionSetId": null,
"dnsPrefix": "dcl-ussc-r-dcl-ussc-rg01-6d6c23",
"enablePodSecurityPolicy": null,
"enableRbac": true,
"fqdn": "dcl-ussc-r-dcl-ussc-rg01-6d6c23-6ac570e4.hcp.southcentralus.azmk8s.io",
"id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourcegroups/dcl-ussc-rg01/providers/Microsoft.ContainerService/managedClusters/dcl-ussc-rg01-k8-01",
"identity": null,
"identityProfile": null,
"kubernetesVersion": "1.18.14",
"linuxProfile": {
  "adminUsername": "azureuser",
  "ssh": {
    "publicKeys": [
      {
        "keyData": "ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQACruAPKjc0WqrXqM3/zmF2aal7BrNeemBs0PGC0HSSYL/SwCM1
3qivwBV7XtcKYwMk8Qis77i8qiH9qL4UiTMgxJ+CD1RCK7+aDTblEXIBph+2ambSmSEVw1KOLetF/U+a9YIP
5PxRoS/IOgkYsDpLV80RAffNpk6ka/DQAqbx8UHospSPK1OAnfCEJ0QmoxGu0X5OeOJKxFEBPqEUJyPB6Fu
eUew3OyOMf1tJpuAYHG1mS0qYwCshbRbFnsPKW/f12jd73fDTTU3y1lzbw9QZp3IZYOMk0MX+vUWby9q
1JeDp6qgQSL7w/vjtbENMkAI0JC8NGQiWVLFwYixQrRLn"
      }
    ]
  }
},
"location": "southcentralus",
"maxAgentPools": 10,
"name": "dcl-ussc-rg01-k8-01",
"networkProfile": {
  "dnsServiceIp": "10.0.0.10",
  "dockerBridgeCidr": "172.17.0.1/16",
  "loadBalancerProfile": {
    "allocatedOutboundPorts": null,
    "effectiveOutboundIps": [
      {
        "id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourceGroups/MC_dcl-ussc-rg01_dcl-ussc-rg01-k8-01_southcentralus/providers/Microsoft.Network/publicIPAddresses/211c0079-27e9-471b-8c3b-b8a0985d7f05",
        "resourceGroup": "MC_dcl-ussc-rg01_dcl-ussc-rg01-k8-01_southcentralus"
      }
    ]
  }
}
```

AKS - Azure Kubernetes

```
    },
    "idleTimeoutInMinutes": null,
    "managedOutboundIps": {
      "count": 1
    },
    "outboundIpPrefixes": null,
    "outboundIps": null
  },
  "loadBalancerSku": "Standard",
  "networkPlugin": "kubenet",
  "networkPolicy": null,
  "outboundType": "loadBalancer",
  "podCidr": "10.244.0.0/16",
  "serviceCidr": "10.0.0.0/16"
},
"nodeResourceGroup": "MC_dcl-ussc-rg01_dcl-ussc-rg01-k8-01_southcentralus",
"privateFqdn": null,
"provisioningState": "Succeeded",
"resourceGroup": "dcl-ussc-rg01",
"servicePrincipalProfile": {
  "clientId": "0818abf8-2b47-43eb-a7db-6d9ed4b890d9",
  "secret": null
},
"tags": null,
"type": "Microsoft.ContainerService/ManagedClusters",
"windowsProfile": null
}
tshen@ubuntu:~$
```

Login Docker Hub

```
tshen@ubuntu:~$ sudo docker login
```

Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to <https://hub.docker.com> to create one.

Username: tonydshen

Password:

WARNING! Your password will be stored unencrypted in /root/.docker/config.json.

Configure a credential helper to remove this warning. See

<https://docs.docker.com/engine/reference/commandline/login/#credentials-store>

Login Succeeded

```
tshen@ubuntu:~$
```

Listing Docker Hub images

```
tshen@ubuntu:~$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dclpussc01acr.azurecr.io/azure-vote-front	v1	4ea526af5d15	5 weeks ago	944MB
mcr.microsoft.com/azuredocs/azure-vote-front	v1	4ea526af5d15	5 weeks ago	944MB

AKS - Azure Kubernetes

mcr.microsoft.com/oss/bitnami/redis	6.0.8	3a54a920bb6c	4 months ago	103MB
tiangolo/uwsgi-nginx-flask	python3.6	a16ce562e863	5 months ago	944MB
hello-world	latest	bf756fb1ae65	12 months ago	13.3kB

Listing ACR images

```
tshen@ubuntu:~$ sudo az acr repository list -n dclusscrg01k801acr --output table
Unable to get AAD authorization tokens with message: Please run 'az login' to setup account.
Unable to get admin user credentials with message: Please run 'az login' to setup account.
Username: dclusscrg01k801acr
Password:
(Empty)
tshen@ubuntu:~$
```

Tagging Docker Hub image

```
tshen@ubuntu:~$ sudo docker tag mcr.microsoft.com/azuredocs/azure-vote-front:v1
dclusscrg01k801acr.azurecr.io/azure-vote-front:v1
tshen@ubuntu:~/containers/azure-voting-app-redis$ sudo docker images
REPOSITORY                                TAG      IMAGE ID      CREATED        SIZE
dclpussc01acr.azurecr.io/azure-vote-front  v1       4ea526af5d15  5 weeks ago   944MB
dclpussc01acr.azurecr.io/azure-vote-front  v2       4ea526af5d15  5 weeks ago   944MB
dclusscrg01k801acr.azurecr.io/azure-vote-front v1       4ea526af5d15  5 weeks ago   944MB
mcr.microsoft.com/azuredocs/azure-vote-front v1       4ea526af5d15  5 weeks ago   944MB
mcr.microsoft.com/oss/bitnami/redis        6.0.8    3a54a920bb6c  4 months ago  103MB
tiangolo/uwsgi-nginx-flask                 python3.6 a16ce562e863  5 months ago  944MB
hello-world                                latest   bf756fb1ae65  12 months ago 13.3kB
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Pushing Docker Hub image to ACR

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ sudo docker push
dclusscrg01k801acr.azurecr.io/azure-vote-front:v1
The push refers to repository [dclusscrg01k801acr.azurecr.io/azure-vote-front]
680e61d1f560: Pushed
7a03a307b6d4: Pushed
1971b933c82a: Pushed
596464cb926b: Pushed
07289a4a4d58: Pushed
4249ef06ae69: Pushed
e775b278161a: Pushed
27d9a6169c9f: Pushed
b9557a3d31d9: Pushed
c8f52965417f: Pushed
9cde96ca61c2: Pushed
f3fd11fd6d6c: Pushed
e553743c668b: Pushed
88af765ea71f: Pushed
b47c358499df: Pushed
cbd550635f44: Pushed
2efef025159b: Pushed
```

AKS - Azure Kubernetes

```
3acbf6947195: Pushed
f90fe4978ca2: Pushed
88383b8e3cb5: Pushed
e93627dfc607: Pushed
8f23b00cc77f: Pushed
cf691a2ea3f9: Pushed
3d3e92e98337: Pushed
8967306e673e: Pushed
9794a3b3ed45: Pushed
5f77a51ade6a: Pushed
e40d297cf5f8: Pushed
v1: digest: sha256:9e545b3621ecbbfbcf1034beafae91aecffaf45541e52ed9a5ed7fcfaed15a6c size: 6175
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Listing ACR again

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ sudo az acr repository list -n dclusscrg01k801acr --
output table
Unable to get AAD authorization tokens with message: Please run 'az login' to setup account.
Unable to get admin user credentials with message: Please run 'az login' to setup account.
Username: dclusscrg01k801acr
Password:
Result
-----
azure-vote-front
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Querying ACR (Optional)

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ az acr list --query "[].{name:name}" --output tsv
dclusscrg01k801acr
tshen@ubuntu:~/containers/azure-voting-app-redis$ az acr repository list -n dclusscrg01k801acr --
output tsv
azure-vote-front
tshen@ubuntu:~/containers/azure-voting-app-redis$ az acr repository show-tags --repository azure-
vote-front -n dclusscrg01k801acr --output tsv
v1
tshen@ubuntu:~/containers/azure-voting-app-redis$ az acr repository show -n dclusscrg01k801acr --
image azure-vote-front:v1
{
  "changeableAttributes": {
    "deleteEnabled": true,
    "listEnabled": true,
    "readEnabled": true,
    "writeEnabled": true
  },
  "createdTime": "2021-01-26T19:16:24.2048529Z",
  "digest": "sha256:9e545b3621ecbbfbcf1034beafae91aecffaf45541e52ed9a5ed7fcfaed15a6c",
  "lastUpdateTime": "2021-01-26T19:16:24.2048529Z",
  "name": "v1",
```

AKS - Azure Kubernetes

```
"signed": false
}
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Updating YAML

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ cat azure-vote-all-in-one-redis.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: azure-vote-back
spec:
  replicas: 1
  selector:
    matchLabels:
      app: azure-vote-back
  template:
    metadata:
      labels:
        app: azure-vote-back
    spec:
      nodeSelector:
        "beta.kubernetes.io/os": linux
      containers:
        - name: azure-vote-back
          image: mcr.microsoft.com/oss/bitnami/redis:6.0.8
          env:
            - name: ALLOW_EMPTY_PASSWORD
              value: "yes"
          ports:
            - containerPort: 6379
              name: redis
---
apiVersion: v1
kind: Service
metadata:
  name: azure-vote-back
spec:
  ports:
    - port: 6379
  selector:
    app: azure-vote-back
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: azure-vote-front
spec:
```

AKS - Azure Kubernetes

```
replicas: 3
selector:
  matchLabels:
    app: azure-vote-front
strategy:
  rollingUpdate:
    maxSurge: 1
    maxUnavailable: 1
minReadySeconds: 5
template:
  metadata:
    labels:
      app: azure-vote-front
  spec:
    nodeSelector:
      "beta.kubernetes.io/os": linux
    containers:
      - name: azure-vote-front
        image: dclusscrg01k801acr.azurecr.io/azure-vote-front:v1
        ports:
          - containerPort: 80
        resources:
          requests:
            cpu: 250m
          limits:
            cpu: 500m
        env:
          - name: REDIS
            value: "azure-vote-back"
---
apiVersion: v1
kind: Service
metadata:
  name: azure-vote-front
spec:
  type: LoadBalancer
  ports:
    - port: 80
  selector:
    app: azure-vote-front
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Deploying app

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ kubectl apply -f azure-vote-all-in-one-redis.yaml
deployment.apps/azure-vote-back created
service/azure-vote-back created
deployment.apps/azure-vote-front created
service/azure-vote-front created
```

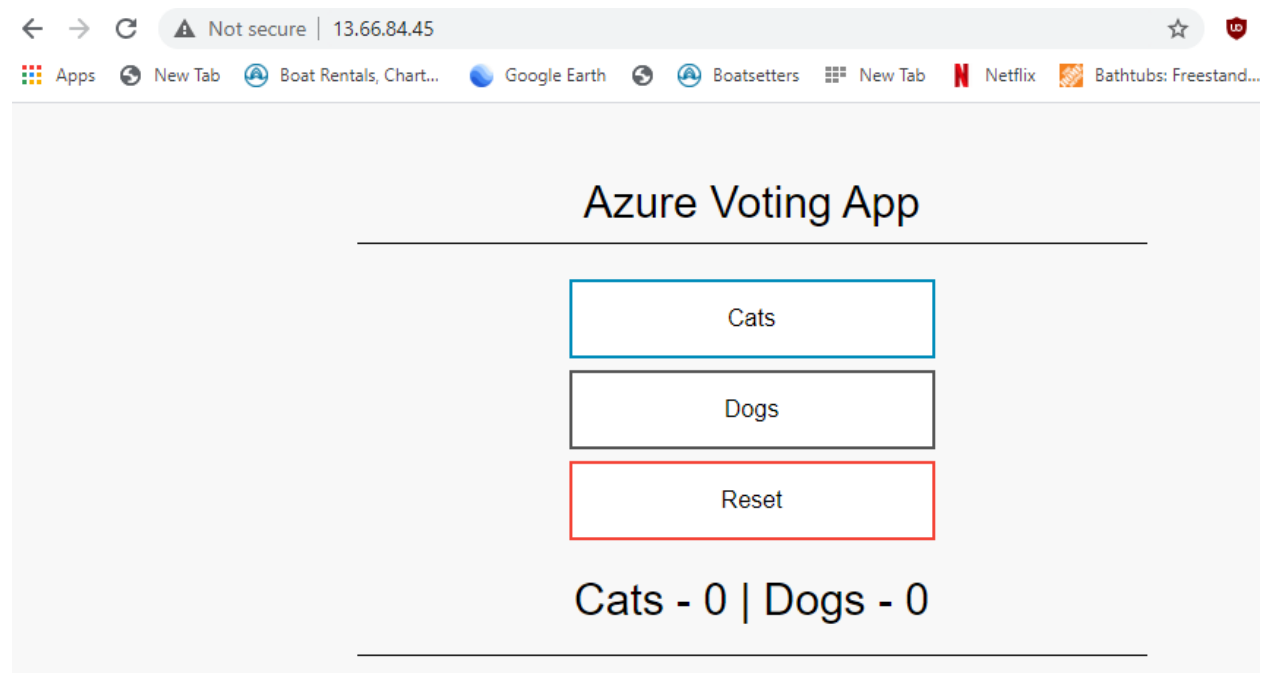
AKS - Azure Kubernetes

```
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Querying services

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ kubectl get service azure-vote-back
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE
azure-vote-back ClusterIP   10.0.219.22  <none>        6379/TCP  2m31s
tshen@ubuntu:~/containers/azure-voting-app-redis$ kubectl get service azure-vote-front
NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
azure-vote-front LoadBalancer 10.0.154.209  13.66.84.45   80:32706/TCP 2m37s
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Verifying the app running



Appendix D – Using K8 cluster

```
$ kubectl describe pods
$ kubectl describe nodes
```

Connecting to K8 cluster

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ az aks get-credentials --resource-group dclp-ussc-01 --name dclp-ussc-k8-01
```

Connecting to ACR

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ sudo az acr login --name dclpussc01acr
```


AKS - Azure Kubernetes

Note: required login password can be obtained from Azure Portal ACR Access Keys page

Displaying repositories/images in ACR

```
tshen@ubuntu:~/yaml$ sudo az acr repository list --name dclpusscdevopsk801acr --output table
```

Unable to get AAD authorization tokens with message: Please run 'az login' to setup account.

Unable to get admin user credentials with message: Please run 'az login' to setup account.

Username: dclpusscdevopsk801acr

Password:

Result

dclpusscdevopsk801106b (Repo in ACR dclpusscdevopsk801acr)

```
tshen@ubuntu:~/yaml$
```

Note: Each repository holds one image with the same name. Image is tagged dignifies the image version.
Tag is a number

```
tshen@ubuntu:~/yaml$ az acr repository show -n dclpusscdevopsk801acr --image
```

```
dclpusscdevopsk801106b:1
```

```
{
  "changeableAttributes": {
    "deleteEnabled": true,
    "listEnabled": true,
    "readEnabled": true,
    "writeEnabled": true
  },
  "createdTime": "2020-12-18T12:46:58.8576705Z",
  "digest": "sha256:ed8ceea0bb4b06eed510a9f0aded2f949442f2689a6a1624597782385c87f76d",
  "lastUpdateTime": "2020-12-18T12:46:58.8576705Z",
  "name": "1",
  "signed": false
}
```

```
tshen@ubuntu:~/yaml$ az acr repository show -n dclpusscdevopsk801acr --image
```

```
dclpusscdevopsk801106b:2
```

```
{
  "changeableAttributes": {
    "deleteEnabled": true,
    "listEnabled": true,
    "readEnabled": true,
    "writeEnabled": true
  },
  "createdTime": "2020-12-18T14:15:14.1670855Z",
  "digest":
sha256:65d89dadf7c485d59538d0f164008449e49dcce8234750d32a6bb5115c8bc73a",
  "lastUpdateTime": "2020-12-18T14:15:14.1670855Z",
  "name": "2",
}
```

AKS - Azure Kubernetes

```
"signed": false
}
tshen@ubuntu:~/yaml$
```

Displaying all namespaces

```
tshen@ubuntu:~/yaml$ kubectl get namespaces
NAME          STATUS AGE
default       Active 3h36m
dev5fb8       Active 3h33m
kube-node-lease Active 3h36m
kube-public    Active 3h36m
kube-system    Active 3h36m
tshen@ubuntu:~/yaml$
```

```
tshen@ubuntu:~/yaml$ kubectl get deployments --all-namespaces=true
NAMESPACE NAME          READY UP-TO-DATE AVAILABLE AGE
dev5fb8 sampleapp      2/2 2 2 3h20m
kube-system coredns        2/2 2 2 3h22m
kube-system coredns-autoscaler 1/1 1 1 3h22m
kube-system dashboard-metrics-scraper 1/1 1 1 3h22m
kube-system kubernetes-dashboard 1/1 1 1 3h22m
kube-system metrics-server 1/1 1 1 3h22m
kube-system omsagent-rs 1/1 1 1 3h22m
kube-system tunnelfront 1/1 1 1 3h22m
```

```
tshen@ubuntu:~/yaml$ kubectl get pods -l app=sampleapp --all-namespaces=true
NAMESPACE NAME          READY STATUS RESTARTS AGE
dev5fb8 sampleapp-5dd9c8d9bb-88z5g 1/1 Running 0 118m
dev5fb8 sampleapp-5dd9c8d9bb-ftqlk 1/1 Running 0 118m
tshen@ubuntu:~/yaml$ kubectl describe pod --namespace dev5fb8
Name: sampleapp-5dd9c8d9bb-88z5g
Namespace: dev5fb8
Priority: 0
Node: aks-agentpool-25742782-1/10.240.0.6
Start Time: Fri, 18 Dec 2020 06:16:57 -0800
Labels: app=sampleapp
        pod-template-hash=5dd9c8d9bb
        release=azuredevops
Annotations: <none>
Status: Running
IP: 10.244.2.4
IPs:
  IP: 10.244.2.4
Controlled By: ReplicaSet/sampleapp-5dd9c8d9bb
```

AKS - Azure Kubernetes

Containers:

sampleapp:

Container ID:

docker://5c0bda9397dd1ddb76b06b66dbb817c57dbfc5386b0e9aa0dbc461b91a31f814

Image: dclpusscdevopsk801acr.azurecr.io/dclpusscdevopsk801106b:2

Image ID: docker-

pullable://dclpusscdevopsk801acr.azurecr.io/dclpusscdevopsk801106b@sha256:65d89dadf7c485d59538d0f164008449e49dcce8234750d32a6bb5115c8bc73a

Port: 80/TCP

Host Port: 0/TCP

State: Running

Started: Fri, 18 Dec 2020 06:17:01 -0800

Ready: True

Restart Count: 0

Liveness: http-get http://:http/ delay=30s timeout=1s period=10s #success=1 #failure=3

Readiness: http-get http://:http/ delay=30s timeout=1s period=10s #success=1 #failure=3

Environment:

APPINSIGHTS_INSTRUMENTATIONKEY: ce85697a-3563-428a-ae3f-5bff5bbfb34f

Mounts:

/var/run/secrets/kubernetes.io/serviceaccount from default-token-7p4gn (ro)

Conditions:

Type Status

Initialized True

Ready True

ContainersReady True

PodScheduled True

Volumes:

default-token-7p4gn:

Type: Secret (a volume populated by a Secret)

SecretName: default-token-7p4gn

Optional: false

QoS Class: BestEffort

Node-Selectors: <none>

Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s

node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events: <none>

Name: sampleapp-5dd9c8d9bb-ftqlk

Namespace: dev5fb8

Priority: 0

Node: aks-agentpool-25742782-2/10.240.0.5

Start Time: Fri, 18 Dec 2020 06:17:36 -0800

Labels: app=sampleapp

AKS - Azure Kubernetes

```
pod-template-hash=5dd9c8d9bb
release=azuredevops
Annotations: <none>
Status:      Running
IP:          10.244.1.5
IPs:
  IP:        10.244.1.5
Controlled By: ReplicaSet/sampleapp-5dd9c8d9bb
Containers:
  sampleapp:
    Container ID:
docker://70eb799f77e754735c54bf00e5ce1c0f7b3e131f0859010db7838ad26ee7bf91
    Image:      dclpusscdevopsk801acr.azurecr.io/dclpusscdevopsk801106b:2
    Image ID:    docker-
pullable://dclpusscdevopsk801acr.azurecr.io/dclpusscdevopsk801106b@sha256:65d89dadf7c48
5d59538d0f164008449e49dcce8234750d32a6bb5115c8bc73a
    Port:       80/TCP
    Host Port:   0/TCP
    State:       Running
      Started:    Fri, 18 Dec 2020 06:17:40 -0800
    Ready:       True
    Restart Count: 0
    Liveness:     http-get http://:http/ delay=30s timeout=1s period=10s #success=1 #failure=3
    Readiness:    http-get http://:http/ delay=30s timeout=1s period=10s #success=1 #failure=3
    Environment:
      APPINSIGHTS_INSTRUMENTATIONKEY: ce85697a-3563-428a-ae3f-5bff5bbfb34f
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-7p4gn (ro)
Conditions:
  Type            Status
  Initialized      True
  Ready            True
  ContainersReady  True
  PodScheduled     True
Volumes:
  default-token-7p4gn:
    Type:          Secret (a volume populated by a Secret)
    SecretName:    default-token-7p4gn
    Optional:      false
QoS Class:        BestEffort
Node-Selectors:    <none>
Tolerations:      node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                  node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:           <none>
```

AKS - Azure Kubernetes

tshen@ubuntu:~/yaml\$

Describing node

tshen@ubuntu:~/containers/azure-voting-app-redis\$ **kubectl get nodes**

```
NAME                                STATUS  ROLES  AGE  VERSION
aks-nodepool1-23085697-vmss000000  Ready  agent  11h  v1.18.10
```

tshen@ubuntu:~/containers/azure-voting-app-redis\$ **kubectl describe node aks-nodepool1-23085697-vmss000000**

```
Name:          aks-nodepool1-23085697-vmss000000
Roles:         agent
Labels:        agentpool=nodepool1
               beta.kubernetes.io/arch=amd64
               beta.kubernetes.io/instance-type=Standard_DS2_v2
               beta.kubernetes.io/os=linux
               failure-domain.beta.kubernetes.io/region=southcentralus
               failure-domain.beta.kubernetes.io/zone=0
               kubernetes.azure.com/cluster=MC_dclp-ussc-01_dclp-ussc-k8-01_southcentralus
               kubernetes.azure.com/node-image-version=AKSUBuntu-1804-2020.12.01
               kubernetes.azure.com/role=agent
               kubernetes.io/arch=amd64
               kubernetes.io/hostname=aks-nodepool1-23085697-vmss000000
               kubernetes.io/os=linux
               kubernetes.io/role=agent
               node-role.kubernetes.io/agent=
               node.kubernetes.io/instance-type=Standard_DS2_v2
               storageprofile=managed
               storagetier=Premium_LRS
               topology.kubernetes.io/region=southcentralus
               topology.kubernetes.io/zone=0
```

```
Annotations:   node.alpha.kubernetes.io/ttl: 0
               volumes.kubernetes.io/controller-managed-attach-detach: true
```

CreationTimestamp: Thu, 17 Dec 2020 06:15:41 -0800

Taints: <none>

Unschedulable: false

Lease:

HolderIdentity: aks-nodepool1-23085697-vmss000000

AcquireTime: <unset>

RenewTime: Thu, 17 Dec 2020 17:54:35 -0800

Conditions:

Type	Status	LastHeartbeatTime	LastTransitionTime	Reason
NetworkUnavailable	False	Thu, 17 Dec 2020 06:16:19 -0800	Thu, 17 Dec 2020 06:16:19 -0800	
RouteCreated		RouteController created a route		
MemoryPressure	False	Thu, 17 Dec 2020 17:50:50 -0800	Thu, 17 Dec 2020 06:15:41 -0800	
KubeletHasSufficientMemory		kubelet has sufficient memory available		

AKS - Azure Kubernetes

DiskPressure False Thu, 17 Dec 2020 17:50:50 -0800 Thu, 17 Dec 2020 06:15:41 -0800
KubeletHasNoDiskPressure kubelet has no disk pressure
PIDPressure False Thu, 17 Dec 2020 17:50:50 -0800 Thu, 17 Dec 2020 06:15:41 -0800
KubeletHasSufficientPID kubelet has sufficient PID available
Ready True Thu, 17 Dec 2020 17:50:50 -0800 Thu, 17 Dec 2020 06:15:51 -0800
KubeletReady kubelet is posting ready status. AppArmor enabled

Addresses:

Hostname: aks-nodepool1-23085697-vmss000000
InternalIP: 10.240.0.4

Capacity:

attachable-volumes-azure-disk: 8
cpu: 2
ephemeral-storage: 129900528Ki
hugepages-1Gi: 0
hugepages-2Mi: 0
memory: 7121296Ki
pods: 110

Allocatable:

attachable-volumes-azure-disk: 8
cpu: 1900m
ephemeral-storage: 119716326407
hugepages-1Gi: 0
hugepages-2Mi: 0
memory: 4675984Ki
pods: 110

System Info:

Machine ID: 55f4cb5d4b6f4e7689784a0645b11a57
System UUID: d5bd6e30-faaf-474d-8bdd-68a8dc3d9450
Boot ID: 16000650-7a3a-49f8-a13e-faf2b4d1be15
Kernel Version: 5.4.0-1032-azure
OS Image: Ubuntu 18.04.5 LTS
Operating System: linux
Architecture: amd64
Container Runtime Version: docker://19.3.12
Kubelet Version: v1.18.10
Kube-Proxy Version: v1.18.10

PodCIDR: 10.244.0.0/24

PodCIDRs: 10.244.0.0/24

ProviderID: azure:///subscriptions/6d6c2372-faed-4d8f-86d1-

3019ed20ea44/resourceGroups/mc_dclp-ussc-01_dclp-ussc-k8-

01_southcentralus/providers/Microsoft.Compute/virtualMachineScaleSets/aks-nodepool1-23085697-vmss/virtualMachines/0

Non-terminated Pods: (12 in total)

Namespace	Name	CPU Requests	CPU Limits	Memory Requests	Memory Limits
-----------	------	--------------	------------	-----------------	---------------

default	azure-vote-back-859c8848cb-zxrn2	0 (0%)	0 (0%)	0 (0%)	0 (0%)
---------	----------------------------------	--------	--------	--------	--------

5h58m

AKS - Azure Kubernetes

```

default      azure-vote-front-6fb87b8d57-prgd2    250m (13%)  500m (26%)  0 (0%)      0
(0%)  5h58m
default      nginx0-deployment-646d9cbf66-69hmz    0 (0%)      0 (0%)      0 (0%)      0 (0%)
5h10m
default      nginx0-deployment-646d9cbf66-vc8b6    0 (0%)      0 (0%)      0 (0%)      0 (0%)
5h10m
kube-system  coredns-748cdb7bf4-r5kfm             100m (5%)   0 (0%)      70Mi (1%)   170Mi
(3%)  11h
kube-system  coredns-748cdb7bf4-zqrvk             100m (5%)   0 (0%)      70Mi (1%)   170Mi
(3%)  11h
kube-system  coredns-autoscaler-868b684fd4-b77wx   20m (1%)    0 (0%)      10Mi (0%)    0
(0%)  11h
kube-system  kube-proxy-slbrh                     100m (5%)   0 (0%)      0 (0%)      0 (0%)      11h
kube-system  metrics-server-58fdc875d5-dx4q2       44m (2%)    0 (0%)      55Mi (1%)    0
(0%)  11h
kube-system  omsagent-j9plx                        75m (3%)    250m (13%)  225Mi (4%)   600Mi
(13%) 11h
kube-system  omsagent-rs-7f7df7b7f7-vh6s5         150m (7%)   1 (52%)     250Mi (5%)   1Gi
(22%) 11h
kube-system  tunnelfront-6768c587bf-2nxnk          10m (0%)    0 (0%)      64Mi (1%)    0 (0%)
11h

```

Allocated resources:

(Total limits may be over 100 percent, i.e., overcommitted.)

Resource	Requests	Limits
cpu	849m (44%)	1750m (92%)
memory	744Mi (16%)	1964Mi (43%)
ephemeral-storage	0 (0%)	0 (0%)
hugepages-1Gi	0 (0%)	0 (0%)
hugepages-2Mi	0 (0%)	0 (0%)
attachable-volumes-azure-disk	0	0

Events: <none>

tshen@ubuntu:~/containers/azure-voting-app-redis\$

Listing node pool

tshen@ubuntu:~/containers/azure-voting-app-redis\$ az aks nodepool list -g dclp-ussc-01 --cluster-name dclp-ussc-k8-01

```

[
  {
    "agentPoolType": "VirtualMachineScaleSets",
    "availabilityZones": null,
    "count": 1,
    "enableAutoScaling": null,
    "enableNodePublicIp": null,
    "id": "/subscriptions/6d6c2372-faed-4d8f-86d1-3019ed20ea44/resourcegroups/dclp-ussc-01/providers/Microsoft.ContainerService/managedClusters/dclp-ussc-k8-01/agentPools/nodepool1",
    "maxCount": null,

```

AKS - Azure Kubernetes

```
"maxPods": 110,
"minCount": null,
"name": "nodepool1",
"nodeLabels": null,
"nodeTaints": null,
"orchestratorVersion": "1.18.10",
"osDiskSizeGb": 128,
"osType": "Linux",
"provisioningState": "Succeeded",
"resourceGroup": "dclp-ussc-01",
"scaleSetEvictionPolicy": null,
"scaleSetPriority": null,
"tags": null,
"type": "Microsoft.ContainerService/managedClusters/agentPools",
"vmSize": "Standard_DS2_v2",
"vnetSubnetId": null
}
]
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Listing az extensions

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ az extension list
[
  {
    "extensionType": "whl",
    "name": "azure-devops",
    "version": "0.17.0"
  }
]
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Listing data volumes

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ kubectl get sc
NAME          PROVISIONER          RECLAIMPOLICY  VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION  AGE
azurefile      kubernetes.io/azure-file Delete         Immediate      true           13h
azurefile-premium kubernetes.io/azure-file Delete         Immediate      true           13h
default (default) kubernetes.io/azure-disk Delete         Immediate      true           13h
managed-premium kubernetes.io/azure-disk Delete         Immediate      true           13h
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

Executing shell commands in pod

```
tshen@ubuntu:~/containers/azure-voting-app-redis$ kubectl get pods
NAME          READY  STATUS  RESTARTS  AGE
azure-vote-back-859c8848cb-zxrn2  1/1    Running  0         7h53m
azure-vote-front-6fb87b8d57-prgd2  1/1    Running  0         7h53m
nginx0-deployment-646d9cbf66-69hmz  1/1    Running  0         7h5m
```

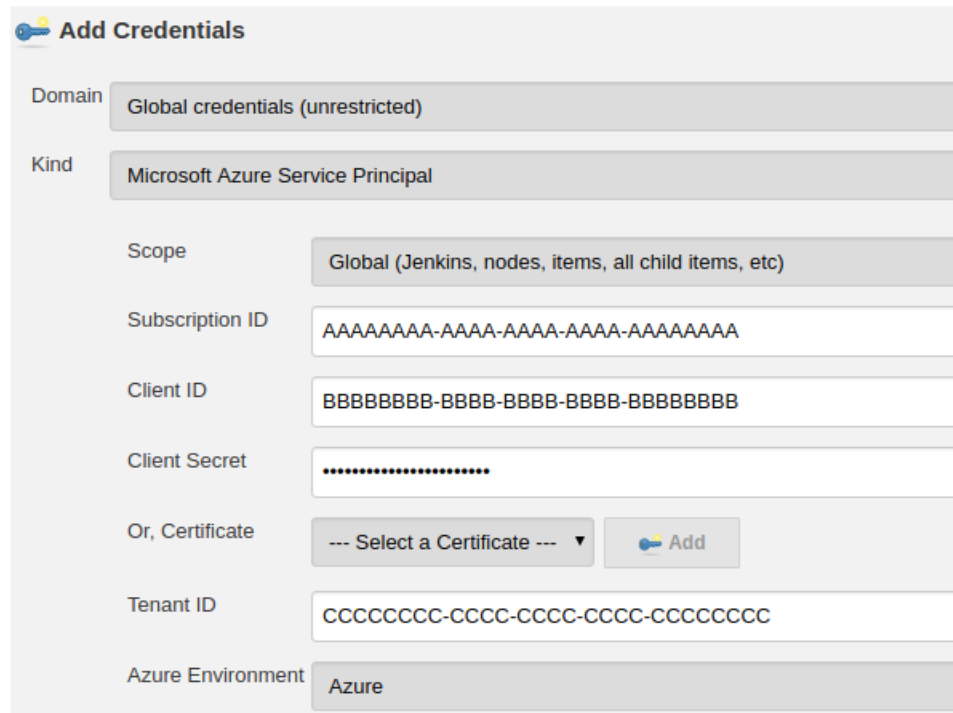

AKS - Azure Kubernetes

```
nginx0-deployment-646d9cbf66-vc8b6 1/1 Running 0 7h5m
tshen@ubuntu:~/containers/azure-voting-app-redis$ kubectl exec nginx0-deployment-646d9cbf66-69hmz -- ls -l
total 80
drwxr-xr-x 2 root root 4096 Dec 9 23:22 bin
drwxr-xr-x 2 root root 4096 Nov 22 12:37 boot
drwxr-xr-x 5 root root 360 Dec 17 20:44 dev
drwxr-xr-x 1 root root 4096 Dec 15 20:20 docker-entrypoint.d
-rwxrwxr-x 1 root root 1202 Dec 15 20:20 docker-entrypoint.sh
drwxr-xr-x 1 root root 4096 Dec 17 20:44 etc
drwxr-xr-x 2 root root 4096 Nov 22 12:37 home
drwxr-xr-x 1 root root 4096 Dec 15 20:20 lib
drwxr-xr-x 2 root root 4096 Dec 9 23:22 lib64
drwxr-xr-x 2 root root 4096 Dec 9 23:22 media
drwxr-xr-x 2 root root 4096 Dec 9 23:22 mnt
drwxr-xr-x 2 root root 4096 Dec 9 23:22 opt
dr-xr-xr-x 208 root root 0 Dec 17 20:44 proc
drwx----- 2 root root 4096 Dec 9 23:22 root
drwxr-xr-x 1 root root 4096 Dec 17 20:44 run
drwxr-xr-x 2 root root 4096 Dec 9 23:22 sbin
drwxr-xr-x 2 root root 4096 Dec 9 23:22 srv
dr-xr-xr-x 12 root root 0 Dec 17 20:44 sys
drwxrwxrwt 1 root root 4096 Dec 15 20:20 tmp
drwxr-xr-x 1 root root 4096 Dec 9 23:22 usr
drwxr-xr-x 1 root root 4096 Dec 9 23:22 var
tshen@ubuntu:~/containers/azure-voting-app-redis$ kubectl exec nginx0-deployment-646d9cbf66-69hmz -- df
Filesystem 1K-blocks Used Available Use% Mounted on
overlay 129900528 23400868 106483276 19% /
tmpfs 65536 0 65536 0% /dev
tmpfs 3560648 0 3560648 0% /sys/fs/cgroup
/dev/sda1 129900528 23400868 106483276 19% /etc/hosts
shm 65536 0 65536 0% /dev/shm
tmpfs 3560648 12 3560636 1% /run/secrets/kubernetes.io/serviceaccount
tmpfs 3560648 0 3560648 0% /proc/acpi
tmpfs 3560648 0 3560648 0% /proc/scsi
tmpfs 3560648 0 3560648 0% /sys/firmware
tshen@ubuntu:~/containers/azure-voting-app-redis$ kubectl exec nginx0-deployment-646d9cbf66-69hmz -- cat /etc/hosts
# Kubernetes-managed hosts file.
127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
fe00::0 ip6-mcastprefix
fe00::1 ip6-allnodes
fe00::2 ip6-allrouters
10.244.0.13 nginx0-deployment-646d9cbf66-69hmz
tshen@ubuntu:~/containers/azure-voting-app-redis$
```

AKS - Azure Kubernetes

Azure Service Principal Example

The completed service principal should use the `id` field for Subscription ID, the `appId` value for Client ID, `password` for Client Secret, and `tenant` for Tenant ID. Select **Add** to add the service principal and then configure the plugin to use the newly created credential.



The screenshot shows the 'Add Credentials' form in Jenkins. The form is titled 'Add Credentials' with a key icon. It has several fields: 'Domain' set to 'Global credentials (unrestricted)', 'Kind' set to 'Microsoft Azure Service Principal', 'Scope' set to 'Global (Jenkins, nodes, items, all child items, etc)', 'Subscription ID' with a placeholder 'AAAAAAAA-AAAA-AAAA-AAAA-AAAAAAAA', 'Client ID' with a placeholder 'BBBBBBBBB-BBBB-BBBB-BBBB-BBBBBBBBBB', 'Client Secret' with a masked password '.....', 'Or, Certificate' with a dropdown '--- Select a Certificate ---' and an 'Add' button, 'Tenant ID' with a placeholder 'CCCCCCCC-CCCC-CCCC-CCCC-CCCCCCCC', and 'Azure Environment' set to 'Azure'.

References

Azure Kubernetes documentation

<https://docs.microsoft.com/en-us/azure/aks/>

Azure Kubernetes REST API

<https://docs.microsoft.com/en-us/rest/api/aks/>

Azure Creating Service Principal

<https://docs.microsoft.com/en-us/azure/developer/jenkins/scale-deployments-using-vm-agents>

Linux Installing Docker CE

<https://linuxconfig.org/how-to-install-and-configure-docker-ce-moby-engine-on-fedora-32>

<https://learn.microsoft.com/en-us/azure/aks/intro-kubernetes>

Appendix

- AKS RBAC integration with AAD
- AKS Monitoring with Azure Monitor and Azure Log Analytics
- AKS Autoscaling
- GPU-enabled nodes
- Confidential computing nodes (Preview)
- Mariner nodes
- Storage volume support
 - Azure disks for single pod access
 - Azure Files for multiple, concurrent pod access
- Virtual networks and ingress
 - An AKS cluster can be deployed into an existing virtual network. In this configuration, every pod in the cluster is assigned an IP address in the virtual network and can directly communicate with other pods in the cluster and other nodes in the virtual network.
 - Pods can also connect to other services in a peered virtual network and on-premises networks over ExpressRoute or site-to-site (S2S) VPN connections.
- Ingress with HTTP application routing
- Development tooling integration
 - Kubernetes has a rich ecosystem of development and management tools that work seamlessly with AKS. These tools include [Helm](#) and the [Kubernetes extension for Visual Studio Code](#).
 - Azure provides several tools that help streamline Kubernetes, such as DevOps Starter.
- DevOps Starter
 - Creates Azure resources (such as AKS).
 - Configures a release pipeline in Azure DevOps Services that includes a build pipeline for CI.
 - Sets up a release pipeline for CD.
 - Generates an Azure Application Insights resource for monitoring.
- AKS Node VM size
 - Minimum 2 vCPUs
- Azure support request