**Step1:** Login into Ubuntu using Putty (IP address, Username, Password)

**Step2:** type command – **sudo su (**To run single command and elevated privilege)

**Step3:** Install Azure CLI

**sudo apt-get update**

**sudo apt-get install curl apt-transport-https lsb-release gpg**

**curl -sL https://packages.microsoft.com/keys/microsoft.asc | \**

**gpg --dearmor | \**

**sudo tee /etc/apt/trusted.gpg.d/microsoft.asc.gpg > /dev/null**

**AZ\_REPO=$(lsb\_release -cs)**

**echo "deb [arch=amd64] https://packages.microsoft.com/repos/azure-cli/ $AZ\_REPO main" | \**

**sudo tee /etc/apt/sources.list.d/azure-cli.list**

**sudo apt-get update**

**sudo apt-get install azure-cli**

**Step4:** Login into azure portal

**az login** (it will open default browser)

else open the URL: <https://microsoft.com/devicelogin> and paste the Key

**Step5:** Docker Install

**sudo apt-get update**

**sudo apt-get install apt-transport-https ca-certificates curl software-properties-common**

**curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -**

**sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"**

**sudo apt-get update**

**sudo apt-get install docker-ce**

**Step6:** Verify the docker installation type below command

**docker version**

(make sure docker will be small letter)

**Step7**: Install Kubectl

**sudo apt-get update && sudo apt-get install -y apt-transport-https**

**curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -**

**echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a /etc/apt/sources.list.d/kubernetes.list**

**sudo apt-get update**

**sudo apt-get install -y kubectl**

**Step8:** Verify the Kubectl installation type below command

**kubectl version**

(make sure everything will be small letter)

**Step9:** Prepare for application

Ref: <https://docs.microsoft.com/en-us/azure/aks/tutorial-kubernetes-prepare-app>

Get the application from GIT using following command

**git clone https://github.com/Azure-Samples/azure-voting-app-redis.git**

Change into the cloned directory

**cd azure-voting-app-redis**

**Step10:** Create Container Images

Following command to convert into images

**docker-compose**

download the Redis image

**docker-compose up –d**

**Step11:** See the created Images use following command

**$ docker images**

**See the running containers**

$ docker ps

**Step12:** Deploy into azure container registry

Create ACR

**az group create --name myResourceGroup --location eastus**

**az acr create --resource-group myResourceGroup --name <acrName> --sku Basic**

Login into ACR

**az acr login --name <acrName>**

To see all images in container

**$ docker images**

**Step13:** To get the login server address

**az acr list --resource-group myResourceGroup --query "[].{acrLoginServer:loginServer}" --output table**

tag your local azure-vote-front image with the acrloginServer address of the container registry. To indicate the image version, add :v1 to the end of the image name

**docker tag azure-vote-front <acrLoginServer>/azure-vote-front:v1**

To verify the tags are applied

**$ docker images**

**Step14:** Push Images to registry

**docker push <acrLoginServer>/azure-vote-front:v1**

**Step15:** List out the images in the registry

**az acr repository list --name <acrName> --output table**

To see the tags for a specific image

**az acr repository show-tags --name <acrName> --repository azure-vote-front --output table**

**Step16**: Deploy into AKS Cluster

To access images stored in ACR, you must grant the AKS service principal the correct rights to pull images from ACR

**az acr show --resource-group myResourceGroup --name <acrName> --query "applicationid" --output tsv**

**Step17:** create Cluster

**Create az aks create \**

**--resource-group myResourceGroup \**

**--name myAKSCluster \**

**--node-count 2 \**

**--service-principal <appId> \**

**--client-secret <password> \**

**--generate-ssh-keys**

**Step18:** Connect to cluster

**az aks get-credentials --resource-group myResourceGroup --name myAKSCluster**

To verify your cluster

**$ kubectl get nodes**

**Step19:** Run Application in AKS

**az acr list --resource-group myResourceGroup --query "[].{acrLoginServer:loginServer}" --output table**

Make sure that you're in the cloned azure-voting-app-redis directory

**vi azure-vote-all-in-one-redis.yaml**

Replace microsoft with your ACR login server name.

**containers:**

**- name: azure-vote-front**

**image: microsoft/azure-vote-front:v1**

Provide your own ACR login server name

**containers:**

**- name: azure-vote-front**

**image: <acrName>.azurecr.io/azure-vote-front:v1**

**kubectl apply -f azure-vote-all-in-one-redis.yaml**

**$ kubectl apply -f azure-vote-all-in-one-redis.yaml**