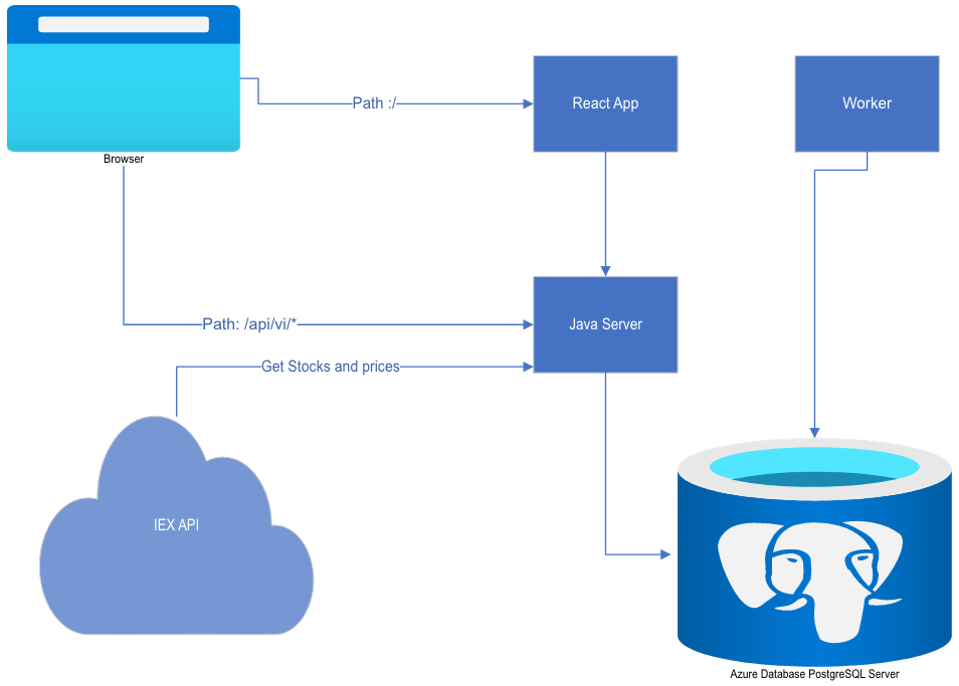
# Innovation Series Steps to Setup

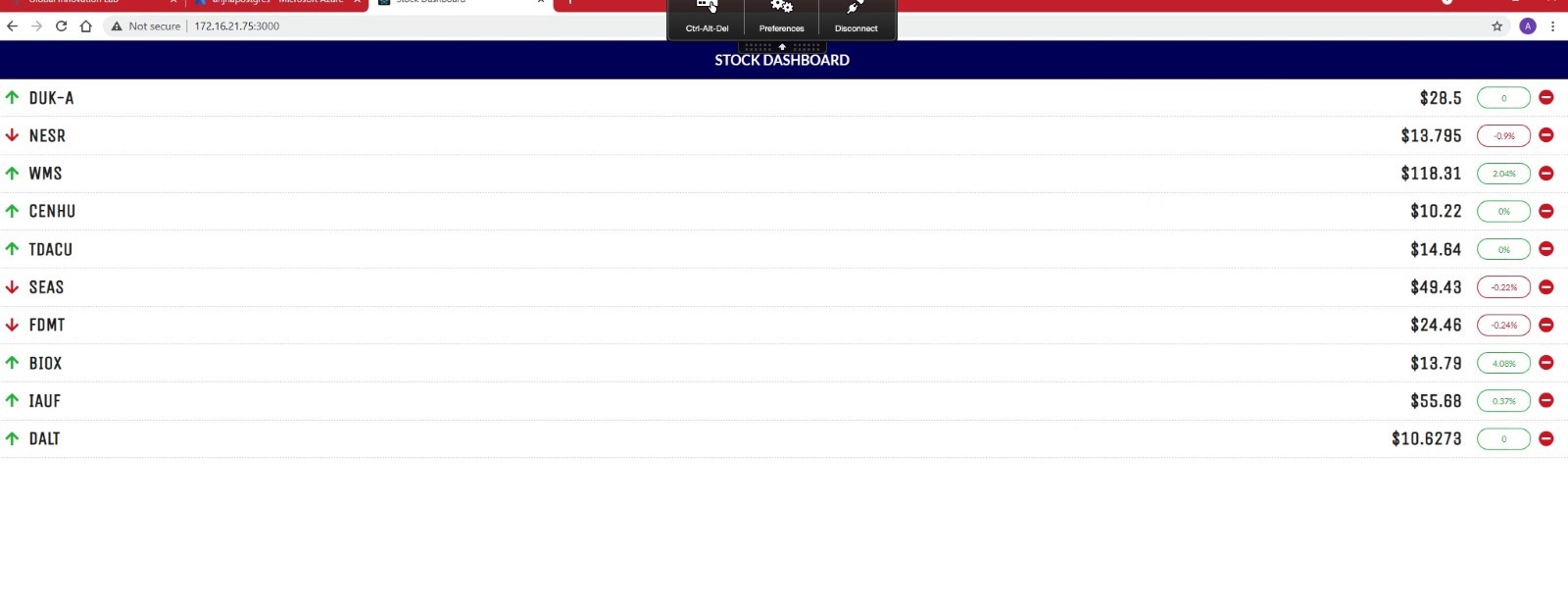
# Demo Application and User Stories

Click here to expand...

Block Diagram



UI



User Story

\_\_\_\_\_\_\_\_\_\_\_

* **As a stocks analyst** I should be able to see all the stocks and their variations in a web application from <https://www.iexcloud.io/docs/api/>

**When** the time interval of 1 min has elapsed

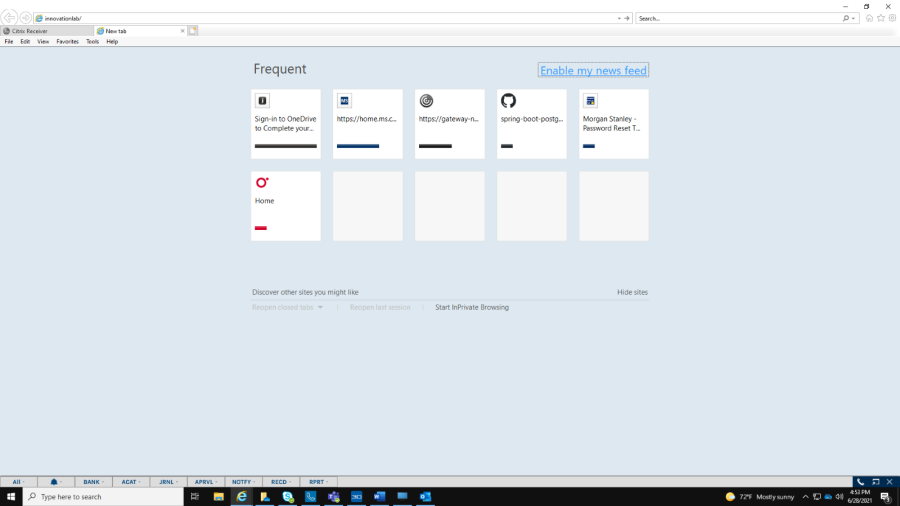
**So that** I that I can see the variations in the UI for each stock.

* **As a developer I** should create a React App which displays a static message along with Stocks and their variations.
* **As a developer I** should create a Spring boot application which fetches all stocks by calling **"**[**https://cloud.iexapis.com/stable/**](https://cloud.iexapis.com/stable/)**tops?token=pk\_53f96e249be3442d803886bb59504119"**  RestFul API  **So that** UI can display all stocks.
* **As a developer I** should create a Spring boot server application which fetches data for stock symbols by calling [**https://cloud.iexapis.com/stable/stock/**](https://cloud.iexapis.com/stable/stock/)**"**+stock.getSymbol()+**"/quote?token=pk\_53f96e249be3442d803886bb59504119** RESTful API **So that** UI can display all stocks data variation and store symbol, Company Name, primary exchange, latest price and latest time in PostgreSQL
* **As a developer I** should create a Spring boot worker application which fetches data for stock symbols from PostgreSQL and display MAX value for each symbol to console. **So that**for each symbol we can get the max value till date.

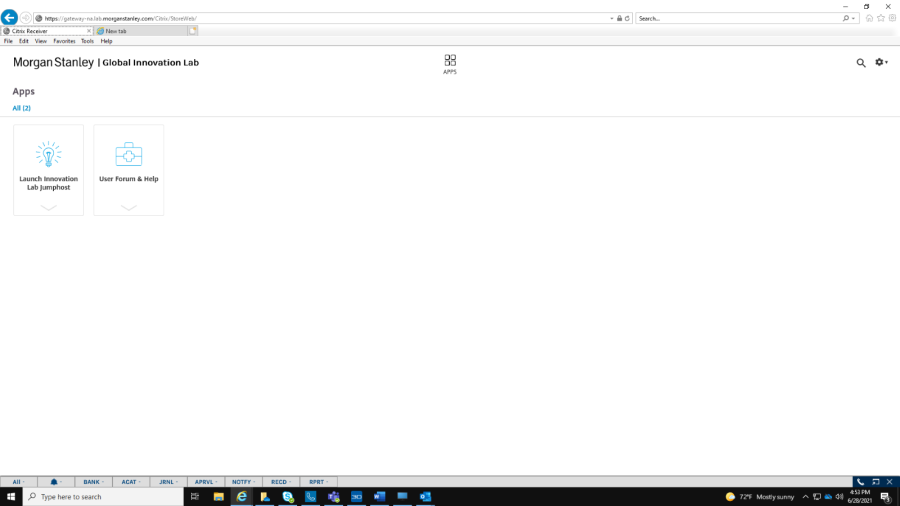
# Day 1 Lab Instructions for Machine setup steps

Click here to expand...

1. Login to Morgan Stanley Desktop
2. Launch a browser and go to innovationlab/



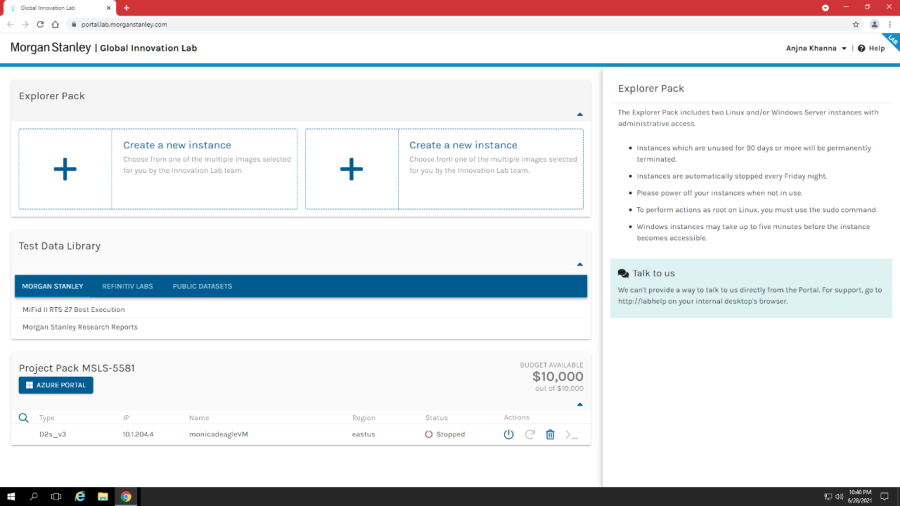
3.  will take you to Global innovation lab page as shown.



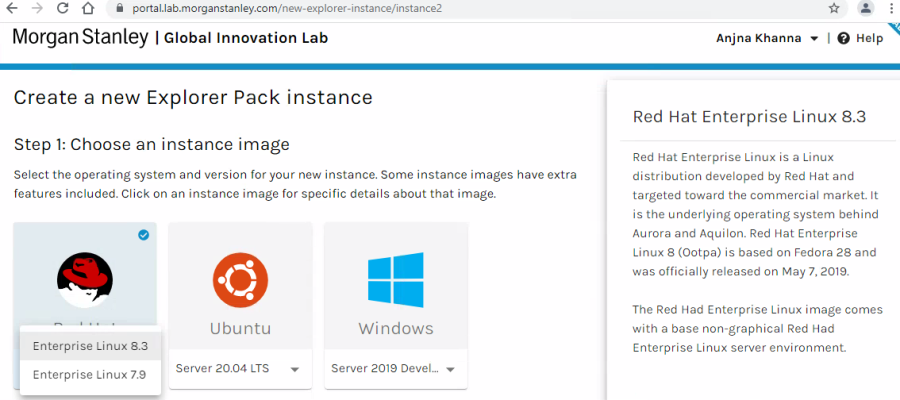
4. Click on 'Launch Innovation Lab Jumphost' it will take you to following page. Then click on "Global Innovation" icon



5. You can see following page with Create new instance boxes. Click on of them.



6. Choose Red Hat Linux machine and from drop down. And click create.



It takes some time to create and start.

Connect to the newly created Red Hat instance.

## Install JAVA,Maven, NodeJS, Docker, Docker compose, Azure CLI

Copy the contents of the following file on Red Hat Instance into .bash\_profile file created in Global Innovation Lab:

On command prompt run following commands:

* git clone <https://github.com/anjnakS/Innovation2021.git>
* cd Innovation2021

and then run following command:

* **source innovationLab.sh**

## Check if installation is complete

Run following commands to check installation:

* **mvn -v**
* **node -v**
* **sudo dnf repolist -v**
* **systemctl is-active docker**

Run the Azure CLI with the az command. To sign in, use [az login](https://docs.microsoft.com/en-us/cli/azure/reference-index#az_login) command. The login to be used is your <Morgan Stanley [<login>@lab.morganstanley.com](mailto:login%3E@lab.morganstanley.com).  ie. [johndoe@lab.morganstanley.com](mailto:johndoe@lab.morganstanley.com)

* **az login**
* (If the CLI can open your default browser, it will do so and load an Azure sign-in page)

# Creating docker containers and Kubernetes cluster

Click here to expand...

7. We need docker ID for creating docker images.

8. Create docker ID from <https://hub.docker.com/signup/>

9. Create PostgreSQL server on Azure :

* Resource group - default (Only default resource group is to be used)
* Name – anjnapostgres (Suggested)

Use this link to create a PostgreSql instance: <https://docs.microsoft.com/en-us/azure/postgresql/tutorial-design-database-using-azure-portal>

10 .Create 3 projects

* React App
* Spring boot –Server (Stores data to PostgreSQL on Azure)
* Java console application –Worker (Reads data from PostgreSQL on Azure)

Or   
      git clone <https://github.com/anjnakS/Innovation2021.git>

11. Change directory to Step3 and change the database connection string in application.properties file for Server and Worker. Change the IP address to address defined in  kubernetes yaml file as mentioned in following step.

12 .For 4 teams the IP address in all the files will be replaced by allowed IP address in the subnet assigned to them. E.g. if you are using default-3 subnet the IP address in files (10.1.207.97) should be replaced by (10.1.206.97) and (10.1.207.98) by (10.1.206.98) and similarly for default-2 by 10.1.205.97 and 10.1.205.98 and finally for subnet default-1 by 10.1.204.97 and 10.1.204.98.

Files to be changed

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/stocks-master/src/main/java/com/example/StocksServer/controller/StocksController.java>

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/stocks-react-master/src/services/StockService.js>

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/k8s/client-cluster-ip-service.yaml>

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/k8s/server-cluster-ip-service.yaml>

Replace

       jdbc:<postgresql://anjnapostgres.postgres.database.azure.com:5432/postgres> with jdbc:postgresql://<your postgressql name >.[postgres.database.azure.com](http://postgres.database.azure.com):5432/postgres.  
       anjnak@anjnapostgres with your username and finally replace Postgres1 with your password.

Files to be changed

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/stocks-master/src/main/resources/application.properties>

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/stocks-worker/src/main/resources/application-context.xml>

13 . Build the applications

1. React Application
   1. Build- npm install
2. Server and Worker Applications
   1. Build – mvn install
   2. Run- mvn springboot:run

14.  Add Docker files for each application

1. For React Application

|  |
| --- |
| <https://github.com/anjnakS/Innovation2021/blob/main/Step3/stocks-react-master/Dockerfile> |

1. For Spring boot application Server and Worker

|  |
| --- |
| <https://github.com/anjnakS/Innovation2021/blob/main/Step3/stocks-master/Dockerfile>  <https://github.com/anjnakS/Innovation2021/blob/main/Step3/stocks-worker/Dockerfile> |

15. Add docker compose yml file

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/docker-compose.yml>

16. Docker should be running. Create Docker images and run containers)

**sudo docker-compose up –build**

--If error about rate limit is encountered run docker login and sign in to personal docker account → **sudo docker login**

17 . Stop running containers **sudo docker-compose down**

18. Create Azure Container Registry <https://docs.microsoft.com/en-us/azure/container-registry/container-registry-get-started-portal#:~:text=%20Quickstart%3A%20Create%20an%20Azure%20container%20registry%20using,an%20image.%20If%20you%20don%27t%20yet...%20More%20>

19. Create Kubernetes YML file(s) in K8s folder

1. React App deployment file

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/k8s/client-deployment.yaml>

1. React App ClusterIP file

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/k8s/client-cluster-ip-service.yaml>

1. Server deployment file

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/k8s/server-deployment.yaml>

1. Server ClusterIP file

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/k8s/server-cluster-ip-service.yaml>

1. Worker deployment file

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/k8s/worker-deployment.yaml>

1. Ingress Service file

<https://github.com/anjnakS/Innovation2021/blob/main/Step3/k8s/ingress-service.yaml>

20. Run command to see docker images

**sudo** **docker images**

21. Using following command to tag images for React App, Server and worker for AKS

**sudo** **docker tag <image name> <acrLoginServer>/<image name>:v1**

E.g.   
sudo docker tag anjnadockerid1/stocksserverfrontend:v3 [anjnaacr.azurecr.io/stocksserverfrontend:v3](http://anjnaacr.azurecr.io/stocksserverfrontend:v3)  
anjnadockerid1/stocksserverbackend:v3 [anjnaacr.azurecr.io/stocksserverbackend:v3](http://anjnaacr.azurecr.io/stocksserverbackend:v3)   
anjnadockerid1/stocksserverworker:v3 [anjnaacr.azurecr.io/stocksserverworker:v3](http://anjnaacr.azurecr.io/stocksserverworker:v3)

22. Push newly tagged images to ACR created in step 17 by running following command

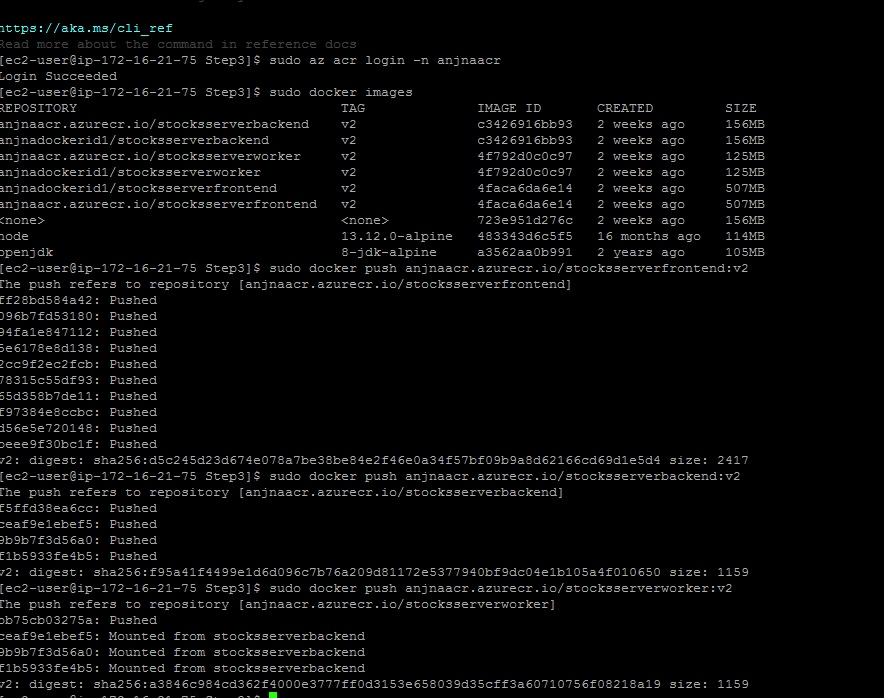
**sudo** **az** **acr login --name <ACR Name>**

**sudo** **docker push <acrLoginServer>/<image name>**

E.g.   
sudo docker push [anjnaacr.azurecr.io/stocksserverfrontend:v3](http://anjnaacr.azurecr.io/stocksserverfrontend:v3)  
sudo docker push [anjnaacr.azurecr.io/stocksserverbackend:v3](http://anjnaacr.azurecr.io/stocksserverbackend:v3)  
sudo docker push [anjnaacr.azurecr.io/stocksserverworker:v3](http://anjnaacr.azurecr.io/stocksserverworker:v3)

23. List images in ACR by running following command you should see 3 images created and pushed for 2 applications.

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



24. Create Azure Kubernetes cluster

Click here to expand...

**Step 1: Fill in these variables with the correct variables for your AKS deployment.**

SUBSCRIPTION="e247041b-0729-4095-9488-564fbc84a3b7"

RESOURCE\_GROUP="default"

AKS\_CLUSTER\_NAME="anjnaaks"

AKS\_VNET\_SUBNET="default-4"

This value will depend on team (default-1, default-2,deafult-3 or default-4)

**Step 2: Log in to Azure from AZ CLI.**

sudo az login

**Step 3: Create route table for the UDR cluster and attach it to the subnet.** When creating a UDR cluster for the first time, a route table needs to be created and attached the the subnet where the cluster will live. This only needs to be done once if all clusters are deployed to the same subnet. Please note:

This set requires the Labs team to remove the read-only lock policy from your subscription.

sudo az network route-table create --subscription $SUBSCRIPTION --name aks-route-table --resource-group infrastructure sudo az network route-table route create --subscription $SUBSCRIPTION --route-table-name aks-route-table --resource-group infrastructure --name default-route --address-prefix 0.0.0.0/0 --next-hop-type VirtualNetworkGateway

sudo az network vnet subnet update --subscription $SUBSCRIPTION --resource-group infrastructure --vnet-name default --name $AKS\_VNET\_SUBNET --route-table aks-route-table

**Step 4: Create the AKS cluster with azure networking and UDR enabled.**

You may wish to customize the cluster further by adding additional switches.

Check the "az aks create" command with no options provided to see a list of options.

Please note: This step might fail with an error related to the service principal credentials. This is due to a known Azure AD propagation delay problem. If you experience it, just try again until it is successful.

sudo az aks create --subscription $SUBSCRIPTION --resource-group $RESOURCE\_GROUP --name $AKS\_CLUSTER\_NAME --outbound-type userDefinedRouting --network-plugin azure --generate-ssh-keys --vnet-subnet-id /subscriptions/$SUBSCRIPTION/resourceGroups/infrastructure/providers/Microsoft.Network/virtualNetworks/default/subnets/$AKS\_VNET\_SUBNET

**Step 5: Granting the AKS cluster permission to use the existing Vnet**

Please note: This step might require assistance from the Labs team, if you do not already have the Owner or User Access Management role in your subscription.

Managed Service Identity is now enabled by default for newly created AKS clusters.

For newer clusters and clusters using Managed Service Identity,

if you're experiencing problems creating load balancers where the error message contains "does not have authorization to perform action 'Microsoft.Network/virtualNetworks/subnets/read' over scope", please use the following commands to grant AKS access to the Vnet:

SP\_ID=$(az resource list --subscription $SUBSCRIPTION --resource-group $RESOURCE\_GROUP --name $AKS\_CLUSTER\_NAME --query [\*].identity.principalId -o tsv)

sudo az role assignment create --assignee $SP\_ID --role "Contributor" --scope /subscriptions/$SUBSCRIPTION/resourceGroups/infrastructure

**Step 6: Setting up and using kubectl.**

sudo az aks get-credentials --subscription $SUBSCRIPTION --resource-group $RESOURCE\_GROUP --name $AKS\_CLUSTER\_NAME

25. Attach ACR to AKS

**sudo** **az** **aks update -n $AKS\_CLUSTER\_NAME -g $RESOURCE\_GROUP --attach-acr <acr-name>**

26. Install Kubernetes CLI

cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes]

name=Kubernetes

baseurl=[https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64](https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64enabled=1gpgcheck=1repo_gpgcheck=1gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg)

[enabled=1](https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64enabled=1gpgcheck=1repo_gpgcheck=1gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg)

[gpgcheck=1](https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64enabled=1gpgcheck=1repo_gpgcheck=1gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg)

[repo\_gpgcheck=1](https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64enabled=1gpgcheck=1repo_gpgcheck=1gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg)

[gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg](https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64enabled=1gpgcheck=1repo_gpgcheck=1gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg) [https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg](https://packages.cloud.google.com/yum/doc/rpm-package-key.gpgEOF)

[EOF](https://packages.cloud.google.com/yum/doc/rpm-package-key.gpgEOF)

yum install -y kubectl

27. Connect to Cluster using Kubetctl

**sudo az aks get-credentials --resource-group $RESOURCE\_GROUP --name $AKS\_CLUSTER\_NAME**

28. Run command to see 3 nodes are running

**sudo kubectl get nodes**

29. Deploy applications in AKS. After the command runs successfully all deployments and services are created.

**cd <PATH>/Innovation2021/Step3**

**sudo kubectl apply -f k8s**

30 . Test application

**sudo kubectl get service client-cluster-ip-service –w** you will see output with external IP. To see the application in action, open a web browser to the external IP address of your service. E.g. <https://10.1.207.97:3000>

31. Manually scale pods

**sudo kubectl get pods --This shows 3 pods**

**sudo kubectl scale --replicas=5 deployment/stocksserverfrontend**

**sudo kubectl get pods –** This is show 5 pods

References  
<https://github.com/webmakaka/Docker-and-Kubernetes-The-Complete-Guide>  
<https://app.pluralsight.com/library/courses/azure-container-service-big-picture/table-of-contents>  
<https://docs.microsoft.com/en-us/azure/container-registry/container-registry-get-started-portal>  
<https://docs.microsoft.com/en-us/azure/aks/tutorial-kubernetes-prepare-acr?tabs=azure-cli>   
GIT   
<https://github.com/anjnakS/Innovation2021.git>