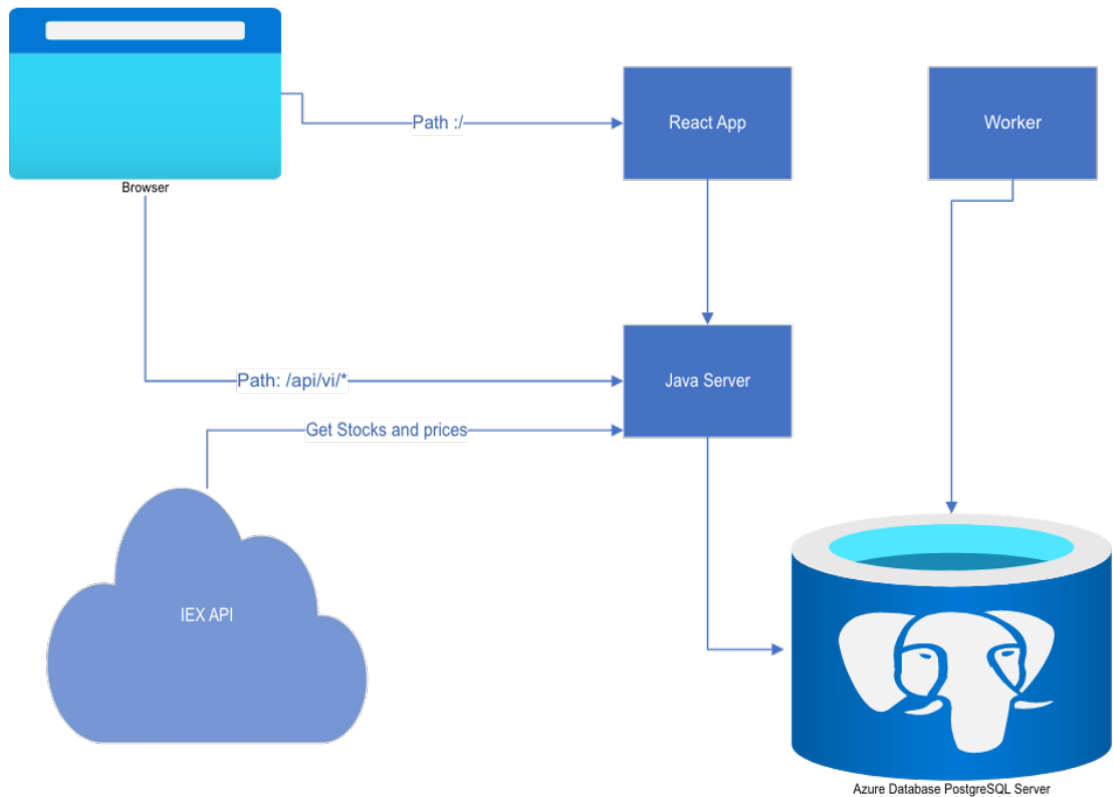


# Innovation Series Steps to Setup

## Demo Application and User Stories

Block Diagram



UI

The screenshot shows a web application titled "STOCK DASHBOARD". It displays a list of stocks with their symbols, current prices, and percentage changes. The interface includes a navigation bar with "Ctrl-Alt-Del", "Preferences", and "Disconnect" buttons. The stock list is as follows:

Stock Symbol	Current Price	Change (%)
DUK-A	\$28.5	0
NESR	\$13.795	-0.9%
WMS	\$118.31	2.04%
CENHU	\$10.22	0%
TDACU	\$14.64	0%
SEAS	\$49.43	-0.22%
FDMT	\$24.46	-0.24%
BIOX	\$13.79	4.08%
IAUF	\$55.68	0.37%
DALT	\$10.6273	0

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/>

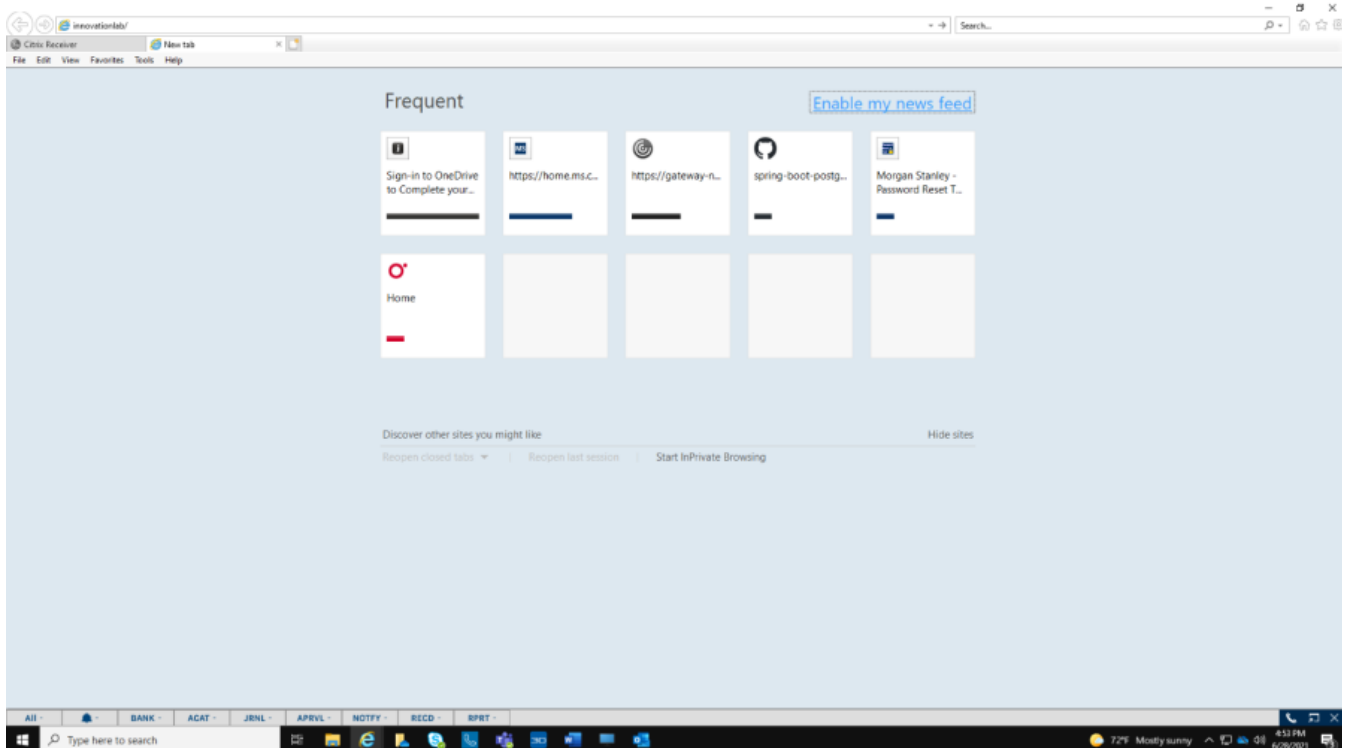
When the time interval of 1 min has elapsed

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

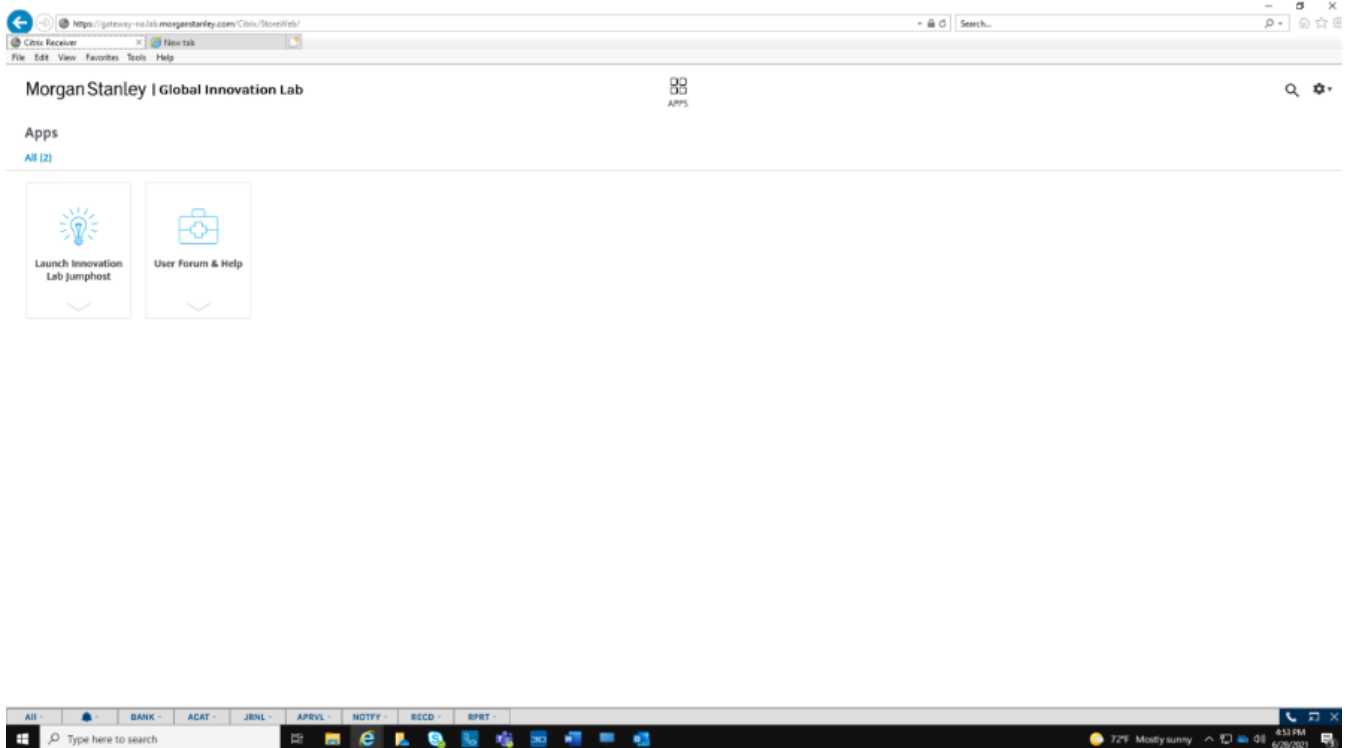
- **As a developer** I should create a React App which displays a static message along with Stocks and their variations.
- **As a developer** I should create a Spring boot application which fetches all stocks by calling "[https://cloud.iexapis.com/stable/tops?token=pk\\_53f96e249be3442d803886bb59504119](https://cloud.iexapis.com/stable/tops?token=pk_53f96e249be3442d803886bb59504119)" 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/>+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

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.

The screenshot shows the Morgan Stanley Global Innovation Lab portal. The top navigation bar includes the logo and user information (Anjna Khanna). The main content area is divided into three sections:

- Explorer Pack:** Contains two "Create a new instance" buttons, each with a plus icon and a description: "Choose from one of the multiple images selected for you by the Innovation Lab team."
- Test Data Library:** A table with columns for Morgan Stanley, Refinitiv Labs, and Public Datasets. It lists "MiFid II RTS 27 Best Execution" and "Morgan Stanley Research Reports".
- Project Pack MSL-5581:** Shows a budget available of \$10,000 out of \$10,000. Below this is a table with columns: Type, IP, Name, Region, Status, and Actions. The table contains one row: "D2s\_v3", "10.1.204.4", "monicaeagleVM", "eastus", "Stopped", and actions (power, refresh, delete, etc.).

On the right side, there is a "Talk to us" section with a message: "We can't provide a way to talk to us directly from the Portal. For support, go to <http://labhelp> on your internal desktop's browser."

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

The screenshot shows the "Create a new Explorer Pack instance" page. The title is "Create a new Explorer Pack instance". Below it is "Step 1: Choose an instance image". The instructions say: "Select the operating system and version for your new instance. Some instance images have extra features included. Click on an instance image for specific details about that image."

There are three instance images available:

- Enterprise Linux 8.3:** Selected (indicated by a blue checkmark). Below it is a dropdown menu showing "Enterprise Linux 7.9".
- Ubuntu:** Below it is a dropdown menu showing "Server 20.04 LTS".
- Windows:** Below it is a dropdown menu showing "Server 2019 Devel...".

On the right side, there is a detailed description of "Red Hat Enterprise Linux 8.3":

Red Hat Enterprise Linux is a Linux distribution developed by Red Hat and targeted toward the commercial market. It is the underlying operating system behind Aurora and Aquilon. Red Hat Enterprise Linux 8 (Ootpa) is based on Fedora 28 and was officially released on May 7, 2019.

The Red Hat Enterprise Linux image comes with a base non-graphical Red Hat Enterprise Linux server environment.

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 command

```
git clone https://github.com/anjakS/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](#) command. The login to be used is your <Morgan Stanley login>@lab.morganstanley.com. ie. 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

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 from localhost to IP address on Red Hat Instance in code.

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 postgresql name >.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 and run the applications

1. React Application
  - a. Build- npm install
  - b. Run – npm start
2. Server and Worker Applications
  - a. Build – mvn install
  - b. 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>

2. 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>

2. React App ClusterIP file

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

3. Server deployment file

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

4. Server ClusterIP file

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

5. Worker deployment file

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

6. 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](https://anjnaacr.azurecr.io/stocksserverfrontend:v3)

anjnadockerid1/stocksserverbackend:v3 [anjnaacr.azurecr.io/stocksserverbackend:v3](https://anjnaacr.azurecr.io/stocksserverbackend:v3)

anjnadockerid1/stocksserverworker:v3 [anjnaacr.azurecr.io/stocksserverworker:v3](https://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](https://anjnaacr.azurecr.io/stocksserverfrontend:v3)

sudo docker push [anjnaacr.azurecr.io/stocksserverbackend:v3](https://anjnaacr.azurecr.io/stocksserverbackend:v3)

sudo docker push [anjnaacr.azurecr.io/stocksserverworker:v3](https://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**

```

https://aka.ms/cli_ref
Read more about the command in reference docs
[ec2-user@ip-172-16-21-75 Step3]$ sudo az acr login -n anjnaacr
Login Succeeded
[ec2-user@ip-172-16-21-75 Step3]$ sudo docker images
REPOSITORY              TAG               IMAGE ID           CREATED            SIZE
anjnaacr.azurecr.io/stocksserverbackend  v2               c3426916bb93      2 weeks ago       156MB
anjnadockerid1/stocksserverbackend       v2               c3426916bb93      2 weeks ago       156MB
anjnaacr.azurecr.io/stocksserverworker   v2               4f792d0c0c97      2 weeks ago       125MB
anjnadockerid1/stocksserverworker        v2               4f792d0c0c97      2 weeks ago       125MB
anjnadockerid1/stocksserverfrontend      v2               4faca6da6e14      2 weeks ago       507MB
anjnaacr.azurecr.io/stocksserverfrontend v2               4faca6da6e14      2 weeks ago       507MB
<none>                                   <none>           723e951d276c      2 weeks ago       156MB
node                                     13.12.0-alpine   483343d6c5f5      16 months ago     114MB
openjdk                                  8-jdk-alpine     a3562aa0b991      2 years ago       105MB
[ec2-user@ip-172-16-21-75 Step3]$ sudo docker push anjnaacr.azurecr.io/stocksserverfrontend:v2
The push refers to repository [anjnaacr.azurecr.io/stocksserverfrontend]
ff28bd584a42: Pushed
096b7fd53180: Pushed
94fa1e847112: Pushed
6e6178e8d138: Pushed
2cc9f2ec2fcb: Pushed
78315c55df93: Pushed
55d358b7de11: Pushed
f97384e8cabc: Pushed
d56e5e720148: Pushed
ee9f30bc1f: Pushed
v2: digest: sha256:d5c245d23d674e078a7be38be84e2f46e0a34f57bf09b9a8d62166cd69d1e5d4 size: 2417
[ec2-user@ip-172-16-21-75 Step3]$ sudo docker push anjnaacr.azurecr.io/stocksserverbackend:v2
The push refers to repository [anjnaacr.azurecr.io/stocksserverbackend]
f5ffd38ea6cc: Pushed
ceaf9e1ebef5: Pushed
9b9b7f3d56a0: Pushed
f1b5933fe4b5: Pushed
v2: digest: sha256:f95a41f4499e1d6d096c7b76a209d81172e5377940bf9dc04e1b105a4f010650 size: 1159
[ec2-user@ip-172-16-21-75 Step3]$ sudo docker push anjnaacr.azurecr.io/stocksserverworker:v2
The push refers to repository [anjnaacr.azurecr.io/stocksserverworker]
bb75cb03275a: Pushed
ceaf9e1ebef5: Mounted from stocksserverbackend
9b9b7f3d56a0: Mounted from stocksserverbackend
f1b5933fe4b5: Mounted from stocksserverbackend
v2: digest: sha256:a3846c984cd362f4000e3777ff0d3153e658039d35cff3a60710756f08218a19 size: 1159

```

## 24. Create Azure Kubernetes cluster

**Step 1:** Fill in these variables with the correct variables for your AKS deployment. SUBSCRIPTION="0e814ead-65cd-4e24-a8d9-f86329958b25"  
RESOURCE\_GROUP="futch"  
AKS\_CLUSTER\_NAME="aks-futch-test"

AKS\_VNET\_SUBNET="default-4" This value will depend on team (default-1, default-2, default-3 or default-4)

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

```
sudo az login
```

**Step 3:** Create route table for the default 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 to 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 and the vnet restriction 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:

```
<ac:structured-macro ac:name="unmigrated-wiki-markup" ac:schema-version="1" ac:macro-id="a232ac1f-67b6-4936-b0ec-2c5543a4560a"><ac:plain-text-body><![CDATA[SP_ID=$(az resource list --subscription $SUBSCRIPTION --resource-group $RESOURCE_GROUP --name $AKS_CLUSTER_NAME --query [*].identity.principalId -o tsv)]]></ac:plain-text-body></ac:structured-macro>
```

```
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  
kubectl get nodes
```

25. Attach ACR to AKS

```
sudo az aks update -n myAKSCluster -g myResourceGroup --attach-acr <acr-name>
```

26. Install Kubernetes CLI

```
sudo az aks install-cli
```

```
PATH="/usr/local/bin:$PATH"
```

27. Connect to Cluster using Kubectl

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

28. Run command to see 3 nodes are running

```
kubectl get nodes
```

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

```
cd <PATH>/Innovation2021/Step3
```

```
kubectl apply -f k8s
```

30. Test application



**kubecttl 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

**kubecttl get pods --This shows 3 pods**

**kubecttl scale --replicas=5 deployment/stocksserverfrontend**

**kubecttl 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>