# AKS SETUP

Table of Contents

[1.Pre – Requisites 1](#_Toc136339978)

[2. Secret Key for Docker Hub 1](#_Toc136339979)

[3. Build and push image to Docker Hub 1](#_Toc136339980)

[4. Resource Group connection 1](#_Toc136339981)

[5. Azure Portal Login 2](#_Toc136339982)

[6. Namespace Creation 2](#_Toc136339983)

[7. Create configmap for particular namespace 2](#_Toc136339984)

[8. Create Service for nflows using service Yaml file 3](#_Toc136339985)

[9. Create nflows application Pod using Deployment Yaml file 3](#_Toc136339986)

[10. Create Service and deployment for neo4j DB using Yaml file 3](#_Toc136339987)

[11. Kubectl commands 3](#_Toc136339988)

# 

# 1.Pre – Requisites

* User should have access to the azure Portal to access the resources.
* Resource groups (AKS) should be created based on the requirement.

# 2. Secret Key for Docker Hub

* Create the secret key for stradegi docker hub private repo

# kubectl create secret docker-registry stradegi-docker-hub-secret --docker-server=https://index.docker.io/v1/ --docker-username=stradegi001 --docker-password=\*\*\*\*\*\* --docker-email=thangachan@stradegi.com -n <namespace name>

* Create the docker image based on the requirement using docker file

# 3. Build and push image to Docker Hub

* Replace <version name> with nflows version name in the below command

#docker build --no-cache -t stradegi001/tomcat:<version name> -f <docker file name> .

#docker push stradegi001/tomcat:<version name>

# 4. Resource Group connection

* Connect a resource group using below commands

#az account set --subscription <subscription id>

#az aks get-credentials --resource-group <resource group name> --name <AKS name>

# 5. Azure Portal Login

* Login to the Azure via power shell using the below command

#az login



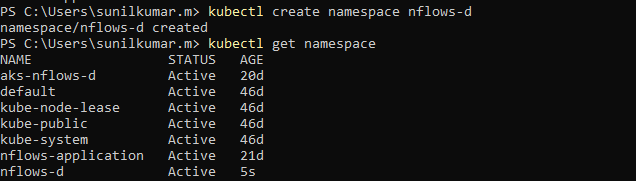
# 6. Namespace Creation

* Create the namespace using the below command

# kubectl create namespace <namespace name>

* To list the available namespace

# Kubectl get namespace



# 7. Create configmap for particular namespace

* Configmap should be created to handle the environment variables
* Replace <namespace name> with namespace name and <config name> with configmap name in below command

#Kubectl -n <namespace name> create configmap <config name> --from-env-file=.\configMapper.conf

* To list the configmap for particular namespace

#kubectl -n <namespace name> get configmap



# 8. Create Service for nflows using service Yaml file

* Replace the <namespace name> with namespace name and <service.yaml> with service yaml file name

#Kubectl -n <namespace name> apply -f .\<service.yaml>

* To check the service is created in particular namespace

# kubectl -n <namespace name> get svc

# 9. Create nflows application Pod using Deployment Yaml file

* Replace the <namespace name> with namespace name and <service.yaml> with deployment yaml file name

#Kubectl -n <namespace name> apply -f .\<deployment.yaml>

* To check the pod is created in particular namespace

# kubectl -n <namespace name> get pods

# 10. Create Service and deployment for neo4j DB using Yaml file

* Replace the <namespace name> with namespace name and yaml file name.

#kubectl -n <namespace name> apply -f .\<neo4jservice.yaml>

#kubectl -n <namespace name> apply -f .\<neo4jdeployment.yaml>

# 11. Kubectl commands

* To check the logs of a particular pod

# kubectl -n <namespace name> logs <pod name> -f

* To check the events of a particular namespace

# kubectl -n <namespace name> get events --sort-by='.metadata.managedFields[0].time'

* To delete the particular deployment in a namespace

#kubectl -n <namespace name> delete deployment.apps/<pod name>

* To scale down the deployment pod

#kubectl -n <namespace name> scale deployment <pod name> --replicas=0

* To scale up the deployment pod

#kubectl -n <namespace name> scale deployment <pod name> --replicas=1