# Install docker and Kubernetes in the machine.

## Docker for windows, [URL:-](file:///C:\Users\ritesh.siotiya\git-project\material\aks\-) https://docs.docker.com/desktop/windows/install/

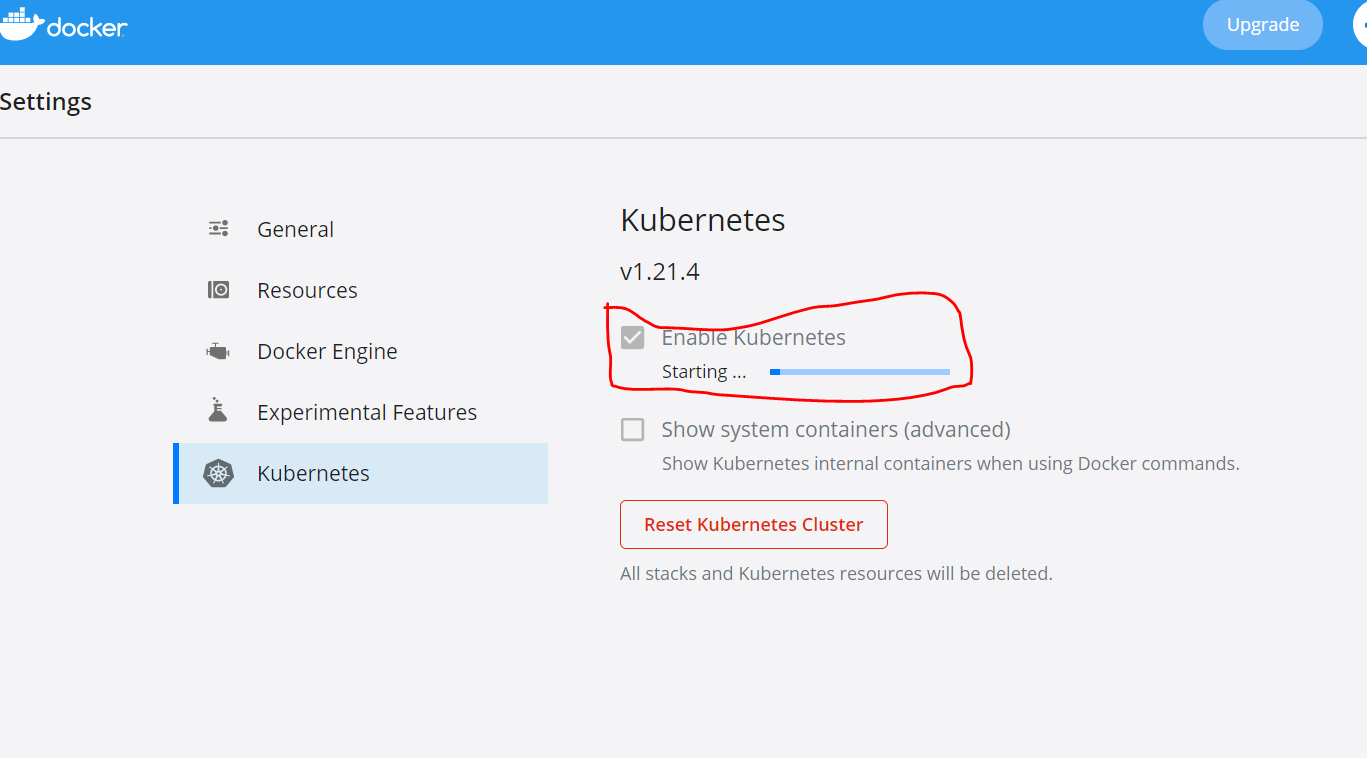
## Enable server service

### Go to Services.msc 🡪 then enable the **service**

## Install the Linux kernel update package

<https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi\>

## Enable Kubernetes.



# Create Kubernetes cluster in AKS.

## Using Azure GUI

## Using script.

### Script is present in script folder.

### Once the AKS Cluster is created then need to Connect to the cluster Configure kubectl to connect to your Kubernetes cluster using the az aks get-credentials command.

### Verify the connection to your cluster using the [kubectl get](https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#get) command. This command returns a list of the cluster nodes

CMD:- kubectl get nodes

# Create azure container registry (ACR).

## Using Azure GUI

## Using script

### Script is present in script folder.

## Create it though command line or from portal.

## To creating the ACR using script kindly follow the script files.

# Provide access to the AKS to pull the images from ACR.

## Using Azure GUI

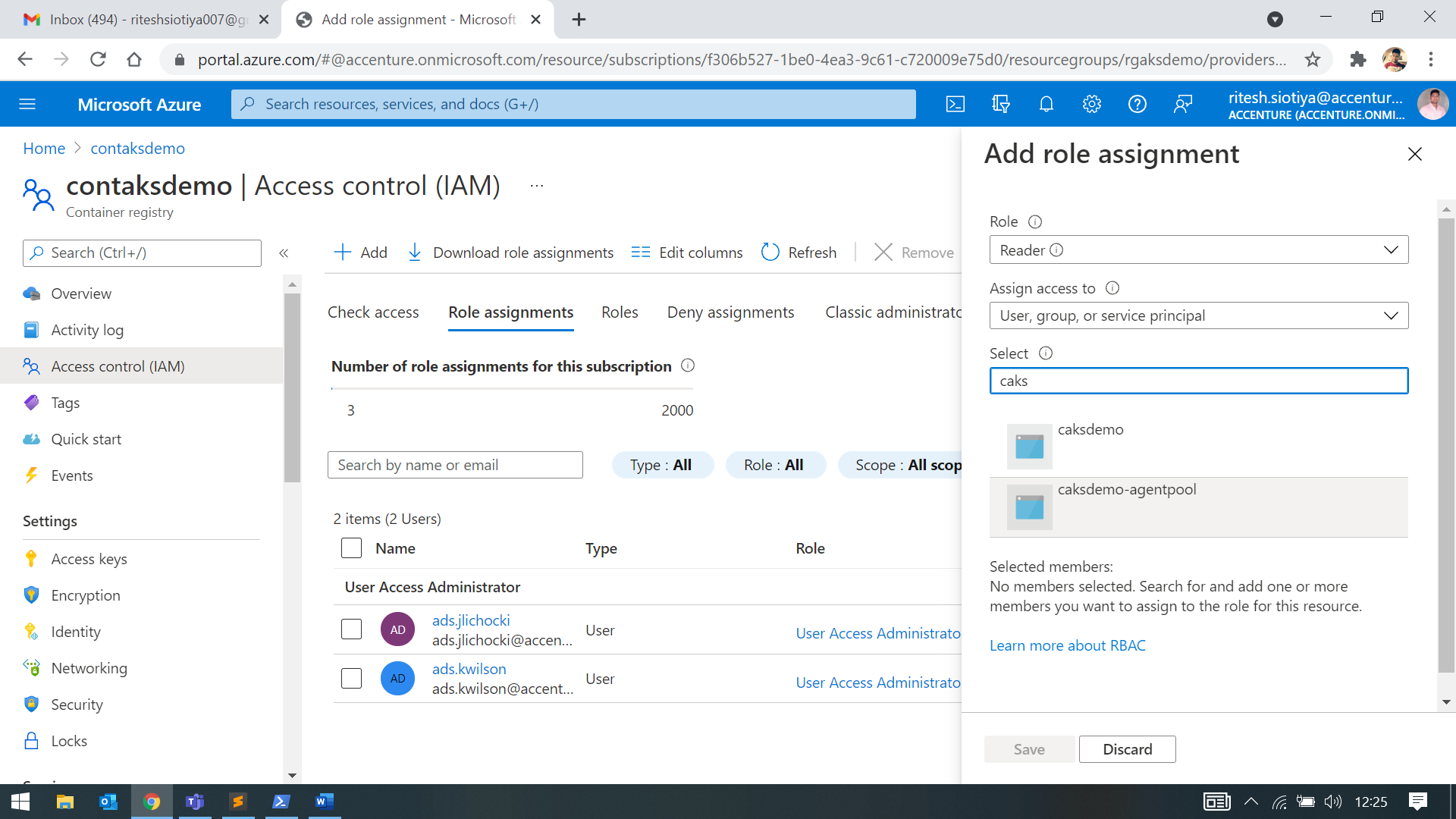
### Goto ACR instance 🡪 Access control IAM 🡪 Add roleassignment

### Role – Reader

### Select – AKS instance. (agent pool identity)

### Click save

### To perform the configuration using script kindly check the script file.



## Using script

### Command is present inside script - az aks update

# Create docker application to be deployed in docker container.

## Create sample app using visual studio. Choose docker app.

## Create an image to push in ACR

## Build the docker app to create an image.

CMD: docker build -f .

CMD: docker build -t aksapp:v1 . || Build with tag

### Once the build is successful it will create image you can check by using

CMD: docker image

CMD: docker image rm -f ----🡪 f- force || To Remove image

## Run image using detached mode

CMD: docker run --name <container name> -d -p <hosting port>:<app port> <image name>:<version>

### Once run command succeeded, container will be create, you can see the container,

CMD: docker ps -a 🡪 -a : all including exited as well

# Push image to ACR

## Login to ACR before push an image.

CMD: az acr login -n <acr name>

## Tag local image

CMD: docker tag <local image name : version > <acr image name: version>

## Push image to ACR

CMD: docker push <acr image name: version>

# Create Deployment in AKS.

## Confirm the user account

CMD : az account show

## To set the subscription follow the command,

CMD: Set-AzContext -SubscriptionId <subid>

## Check the running node

CMD : kubectl get nodes

## Create deployment

CMD : kubectl create deployment <deployment name> --image=<acr image name> --replicas=1

## Check the Pods description

CMD: kubectl describe pods <podname>

CMD: kubectl describe node <nodename>

# Expose service

CMD: kubectl expose deployment < deployment name > --type=LoadBalancer --target-port=80 --port=80

## To watch the service

CMD: kubectl get svc –watch