**DevOPS Automation:**

1. **Login to Cloudshell and select Bash**
2. **Create a resource group**
    a. az group create --name myapp-rg --location eastus

3. **Create a container registry**
    a. az acr create --resource-group myapp-rg --name <YourRegistryUniqueName> --sku Basic
4. **Create a Kubernetes cluster**
    a. az aks create --resource-group myapp-rg  --name myapp --node-count 1 --enable-addons monitoring --generate-ssh-keys

5. **Get repository:**

    a. Login to Github.com on Web portal

    b. **Clone** below repository to your own GIT account:

    [https://github.com/sanjshah2001/sentimentanalysis.git](https://github.com/sanjshah2001/sentimentanalysis.git)

6. **Update ts.yaml file:**

    a. Click on ts.yaml

    b. Click Edit (Pencil Button)

    c. Change line 11 and 15 with your own registry name:

    Image: <YourRegistryUniqueName>.azure.io/timeseries

7. **Update azure-pipelines.yml**

    a. Go to line 15 and change registryname to your own unique name

    `azureContainerRegistry: <YourRegistryUniqueName>.azurecr.io`

8. **Create new DevOPS project**

    a. Login to dev.azure.com

    b. Click on Create New Project with project name finservdevops

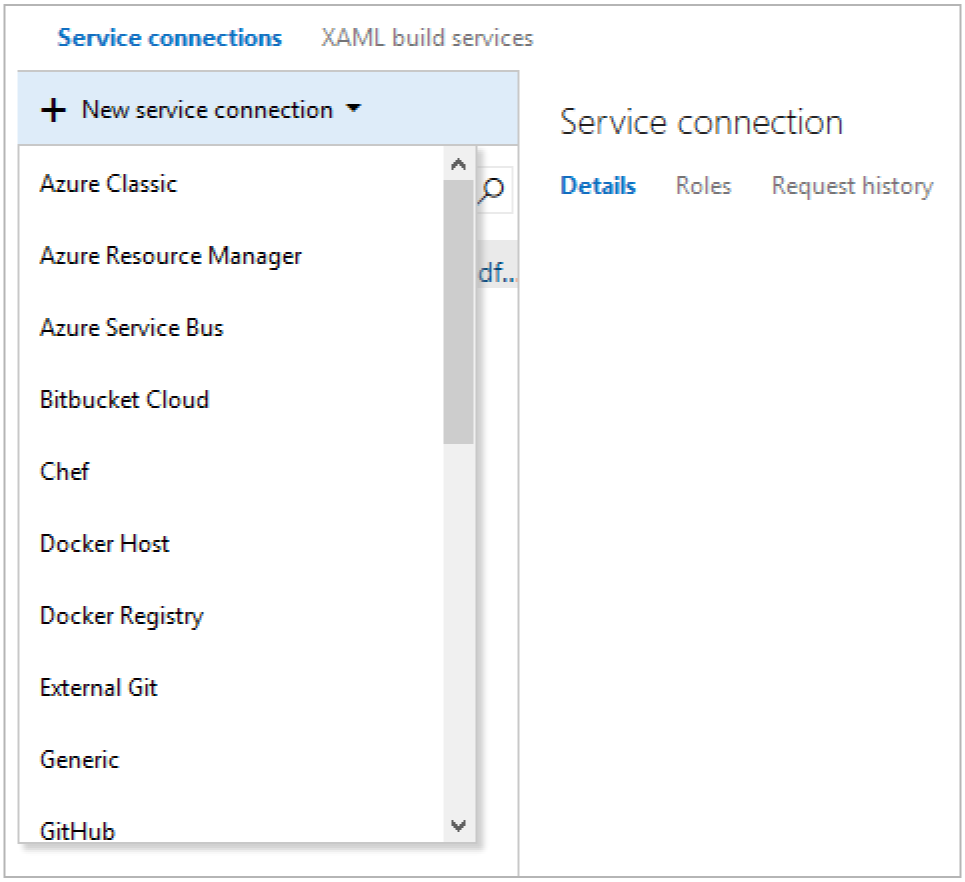
    c. Select visibility as public

    d. Click on advance

    e. Make sure version control is Git and Work item process is Basic

9. **Create Service End Point**

   a. Go to Project settings page at the bottom left
   b. Choose **+ New service connection** and select **Azure Resource Manager**.



Connection Name: finservrg

Scope Level: Subscription

Subscription: Choose from drop down that you used in last exercise

Resource Group: myResourceGroup (Or name you gave in last exercise)

10. **Create Build Pipeline:**

    a. Click on pipeline -> New pipeline

    b. Select GitHub (YAML) (Not enterprise server option)

    c. Select repository that you cloned earlier (Your own repository)

    d. Screen will show azure-pipeline.yml file (DO NOT RUN IT YET)

    e. Go to pipeline on left menu and select “Queue” on top right corner

f. Select branch as your local branch and press Queue

g. Click on “Update ts.yaml” to view progress of the build job

h. As shown, this job will:

   i. Build customized image

   ii. Tag image

   iii. Push image

   iv. Run image

   v. Copy source files on to containers

   vi. Push artifacts to the release cycle

**11. Create release pipeline:**

a. Click on release option on left menu

b. Select “New Pipeline”

c. Select template “Deploy to a Kubernetes cluster” and Apply

d. Select “QA” as stage name and then click save. Choose default options on pop-up window

e. Next to Artifact, click “Add”

f. Click “Build” as source type

g. Choose project name form drop down that you created during build cycle

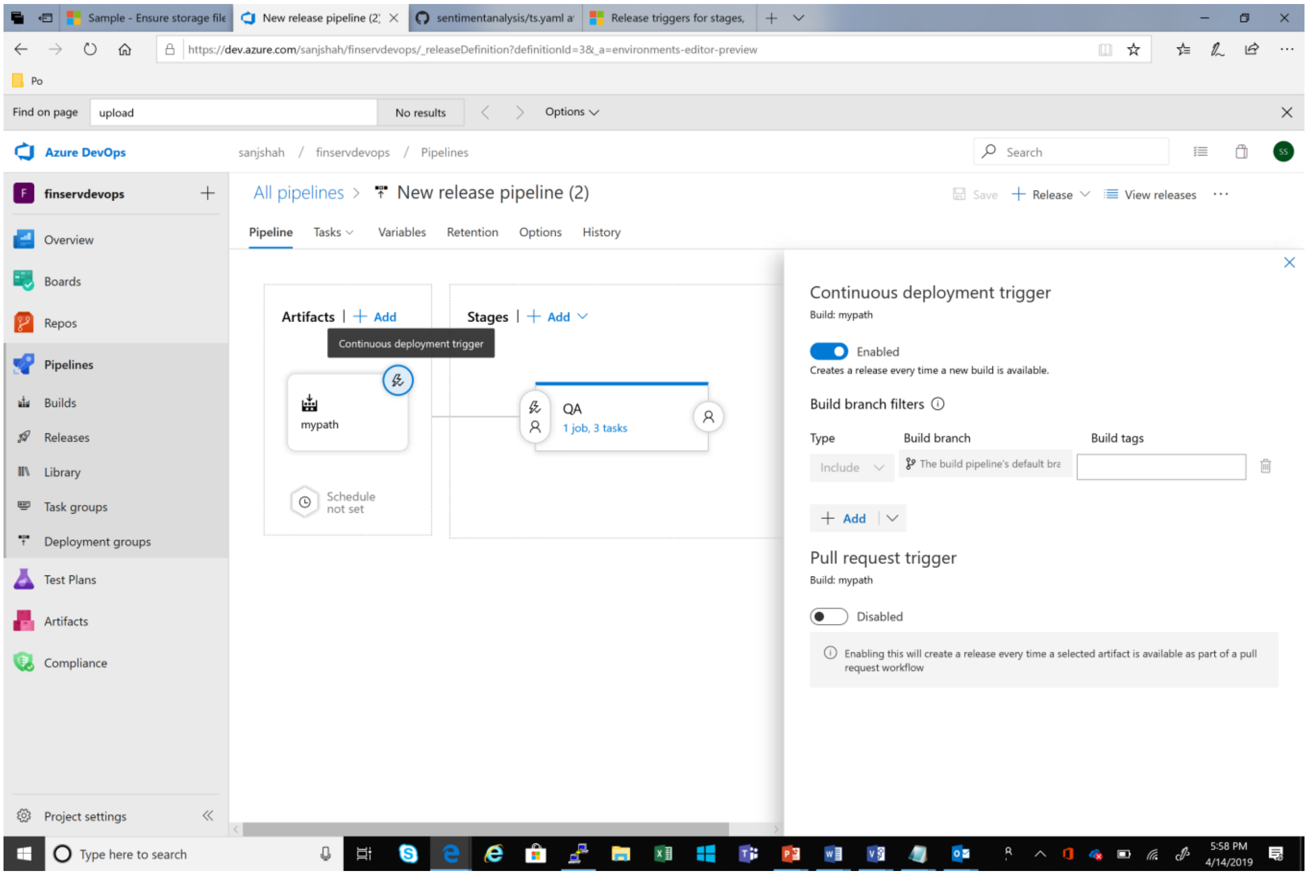
h. Select source repository from drop down as build.pipeline

i. Keep “Default” as latest version

j. Under Source Alias enter “finpath”

k. Click save

l. Click small icon on top of Artifacts box which will open up continuous deployment trigger:



Continuous deployment trigger

Build: mypath

Enabled

Creates a release every time a new build is available.

Build branch filters

| Type | Build branch | Build tags |
|---|---|---|
| Include | The build pipeline's default bra | |

+ Add

Pull request trigger

Build: mypath

Disabled

Enabling this will create a release every time a selected artifact is available as part of a pull request workflow

m. Enable option where is says “Create a release every time new build is available”

n. Save

o. Click on “Task” from the menu on top

p. Click on Kubectl and window will open on right

q. Under Kubernetes Service connection , select “Manage”
(select “Kubernetes” from the dropdown for New Service Connection)

r. This will open another tab to create new service connection

s. Click on Add new service connection

t. Select Azure subscription

u. Give Connection Name “KubeConnection”

v. Select your subscription

w. Drop down menu will automatically find your Kubernetes cluster that you created in previous exercise

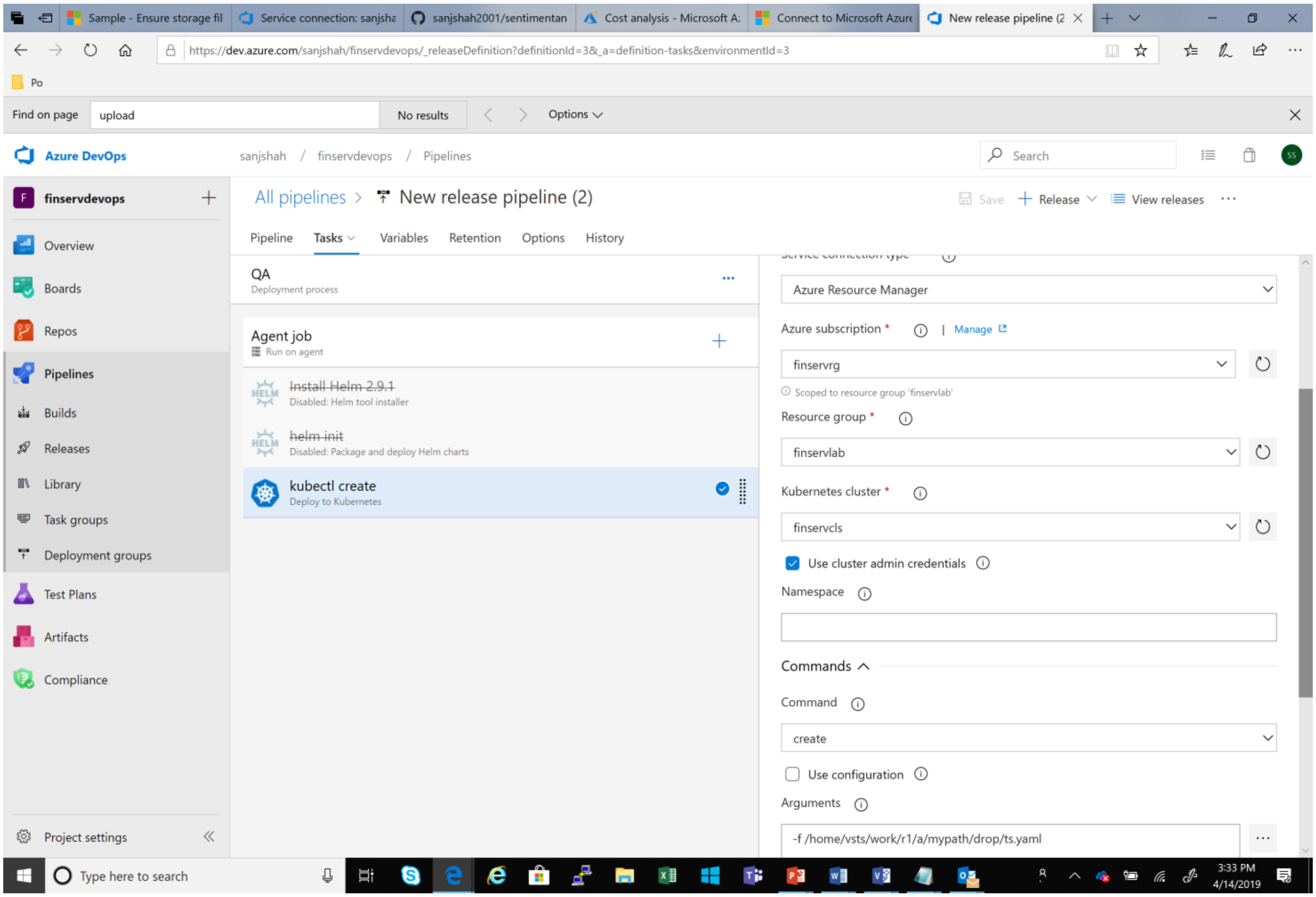
x. Select “default” as namespace

y. Click OK

z. Click on Variable task on top and add following variable:

   i. Name: imageRepoName Value: <YourRegistryUniqueName>.azurecr.io/timeseries

aa. Go back to previous window and refresh Kubernetes service connection option. Drop down will show new connection you created. Select that one



https://dev.azure.com/sanjshah/finservdevops/_releaseDefinition?definitionId=3&_a=definition-tasks&environmentId=3

Find on page: upload

New release pipeline (2)

QA
Deployment process

Agent job
Run on agent

~~Install Helm 2.9.1~~
Disabled: Helm tool installer

~~helm init~~
Disabled: Package and deploy Helm charts

kubectl create
Deploy to Kubernetes

Service connection type: Azure Resource Manager

Azure subscription *: finservrg
Scoped to resource group 'finservlab'

Resource group *: finservlab

Kubernetes cluster *: finservcls

☑ Use cluster admin credentials

Namespace:

Command: create

☐ Use configuration

Arguments: -f /home/vsts/work/r1/a/mypath/drop/ts.yaml

bb. Under command enter “create”

cc. Under argument enter “-f D:\a\r1\a\finpath\drop\ts.yaml”

dd. Click save

## 12. Test new release

a. Go to your GIT repository

b. Click on ts.yaml

c. Click Edit (Pencil Button)

d. Edit file ts.yaml and change POD name from finservpod to finservpodv2 under metadata

e. Save and commit changes

f. Go back to Azure DevOPS build pipeline, a new build would have already triggered

g. Once finished, check release pipeline, which is automatically triggered upon successful finish of build job

h. Go to Cloudshell (Bash) again

   i. az aks get-credentials --resource-group  --name myapp-rg

   ii. kubectl get nodes

   iii. kubectl get pods (To ensure new POD is created)

i. If successful release, a new POD called "finservpodv2" is created, access using:
kubectl exec -it finservpodv2 --container gettsdata -- /bin/bash

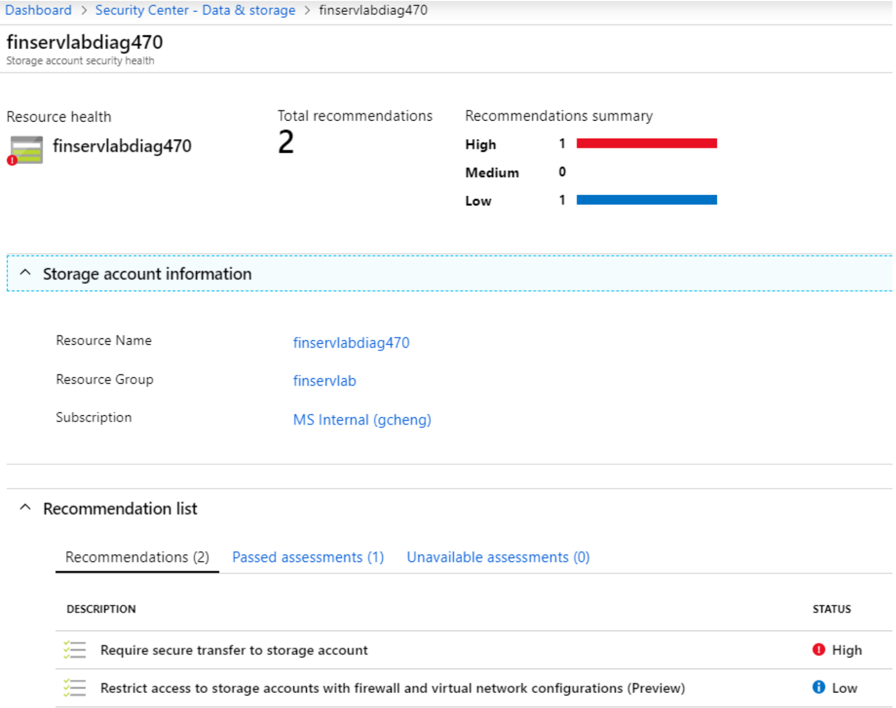
13. **Security and Compliance Testing (Bonus Lab)**

a. Challenge: Modify the python script to export the data files (price_data and senti_data) directly to storage accounts and ensure that file and storage account encryption is enabled for secure data storage. For secure data storage enforcement, you should use Azure Policy to enforce this on your resourcegroup. Below are some helpful links to do so:

   i. https://docs.microsoft.com/en-us/azure/governance/policy/samples/ensure-https-storage-account
   ii. https://docs.microsoft.com/en-us/azure/governance/policy/samples/ensure-storage-file-encryption
   iii. https://docs.microsoft.com/en-us/azure/governance/policy/samples/require-storage-account-encryption
   iv. https://docs.microsoft.com/en-us/azure/governance/policy/concepts/definition-structure

b. If you have an existing storage account within Azure for this lab, once you apply the policies to the resourcegroup, you can take action on them within Azure Security Center:

   i. Example – below shows a storage account that does not meet the secure data transfer policy:



Dashboard > Security Center - Data & storage > finservlabdiag470

finservlabdiag470
Storage account security health

Resource health: finservlabdiag470

Total recommendations: 2

Recommendations summary

| | |
|---|---|
| High | 1 |
| Medium | 0 |
| Low | 1 |

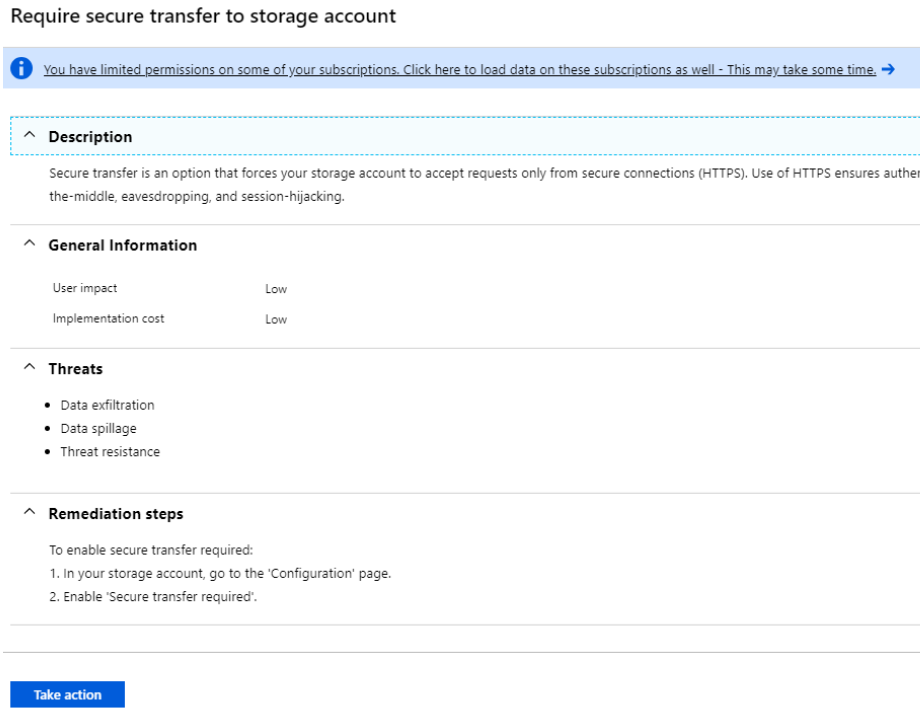
Storage account information

| | |
|---|---|
| Resource Name | finservlabdiag470 |
| Resource Group | finservlab |
| Subscription | MS Internal (gcheng) |

Recommendation list

| DESCRIPTION | STATUS |
|---|---|
| Require secure transfer to storage account | High |
| Restrict access to storage accounts with firewall and virtual network configurations (Preview) | Low |

ii. If you drill down to non-compliant items, it provides remedy actions:



Require secure transfer to storage account

You have limited permissions on some of your subscriptions. Click here to load data on these subscriptions as well - This may take some time.

Description

Secure transfer is an option that forces your storage account to accept requests only from secure connections (HTTPS). Use of HTTPS ensures authen
the-middle, eavesdropping, and session-hijacking.

General Information

| | |
|---|---|
| User impact | Low |
| Implementation cost | Low |

Threats

- Data exfiltration
- Data spillage
- Threat resistance

Remediation steps

To enable secure transfer required:
1. In your storage account, go to the 'Configuration' page.
2. Enable 'Secure transfer required'.

c. If you are interesting in exploring Azure Policy further, there is an additional hands on lab here: https://handsonlabs.microsoft.com/handsonlabs/SelfPacedLabs?storyId=story://content-private/content/sp-azuregovernance/1-azpolicy/a-policy