Custom Training

Day 21

**Docker and Kubernetes**

Docker vs Virtual Machines

* To provide some S/W and H/W functionalities virtually, VMs are needed
* VMs run on Hypervisor which are very costly
* To run applications that do not require all the features of the VM, special containers can be used
* This is where Docker comes into the picture
* Docker uses libraries and runs images on the Docker Engine instead of a hypervisor without the use of complicated infrastructure

| **Docker** | **Virtualization** |
| --- | --- |
| It boots in a few seconds. | It takes a few minutes for VMs to boot. |
| Pre-built docker containers are readily available. | Ready-made VMs are challenging to find. |
| Docker has a complex usage mechanism consisting of both third-party and docker-managed tools. | Tools are easy to use and more straightforward to work with. third-party. |
| Limited to Linux. | Can run a variety of guest OS. |
| Dockers make use of the execution engine. | VMs make use of the hypervisor. |
| It is lightweight. | It is heavyweight. |
| Host OS can be different from container OS. | Host OS can be different from guest OS. |
| Can run many docker containers on a laptop. | Cannot run more than a couple of VMS  on an average laptop. |
| Docker can get a virtual network adapter. It can have separate IPs ad Ports. | Each VMS gets its virtual network adapter. |
| Sharing of files is possible. | Sharing library and files are not possible. |
| Lacks security measures. | Security depends on the hypervisor. |
| A container is portable. | VMS is dependent on a hypervisor. |
| It allows running an application in an isolated environment known as a container | It provides easiness in managing applications, recovery mechanisms, and isolation from the host operating system |

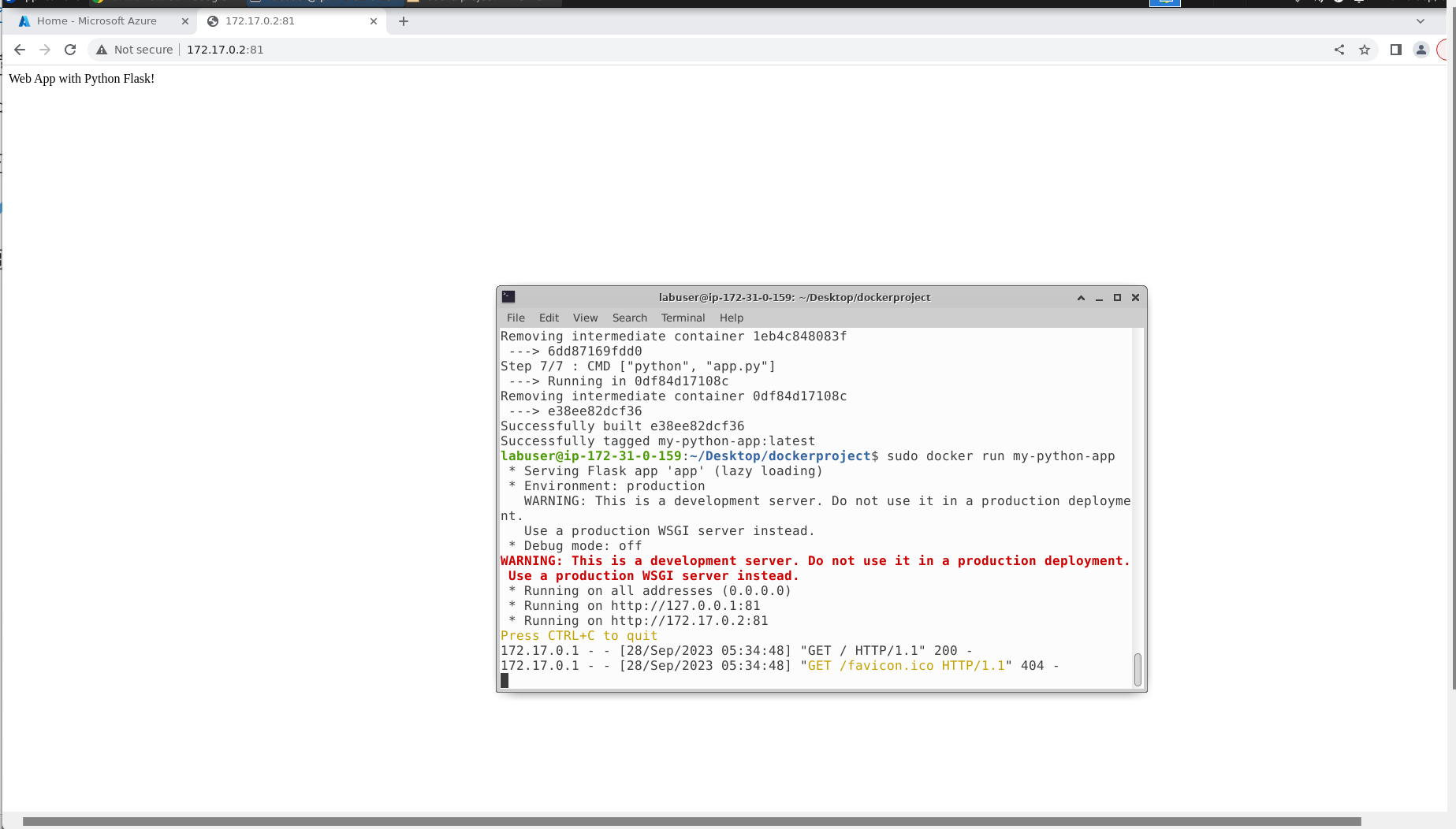
Go to terminal: sudo systemctl start docker

Sudo docker ps

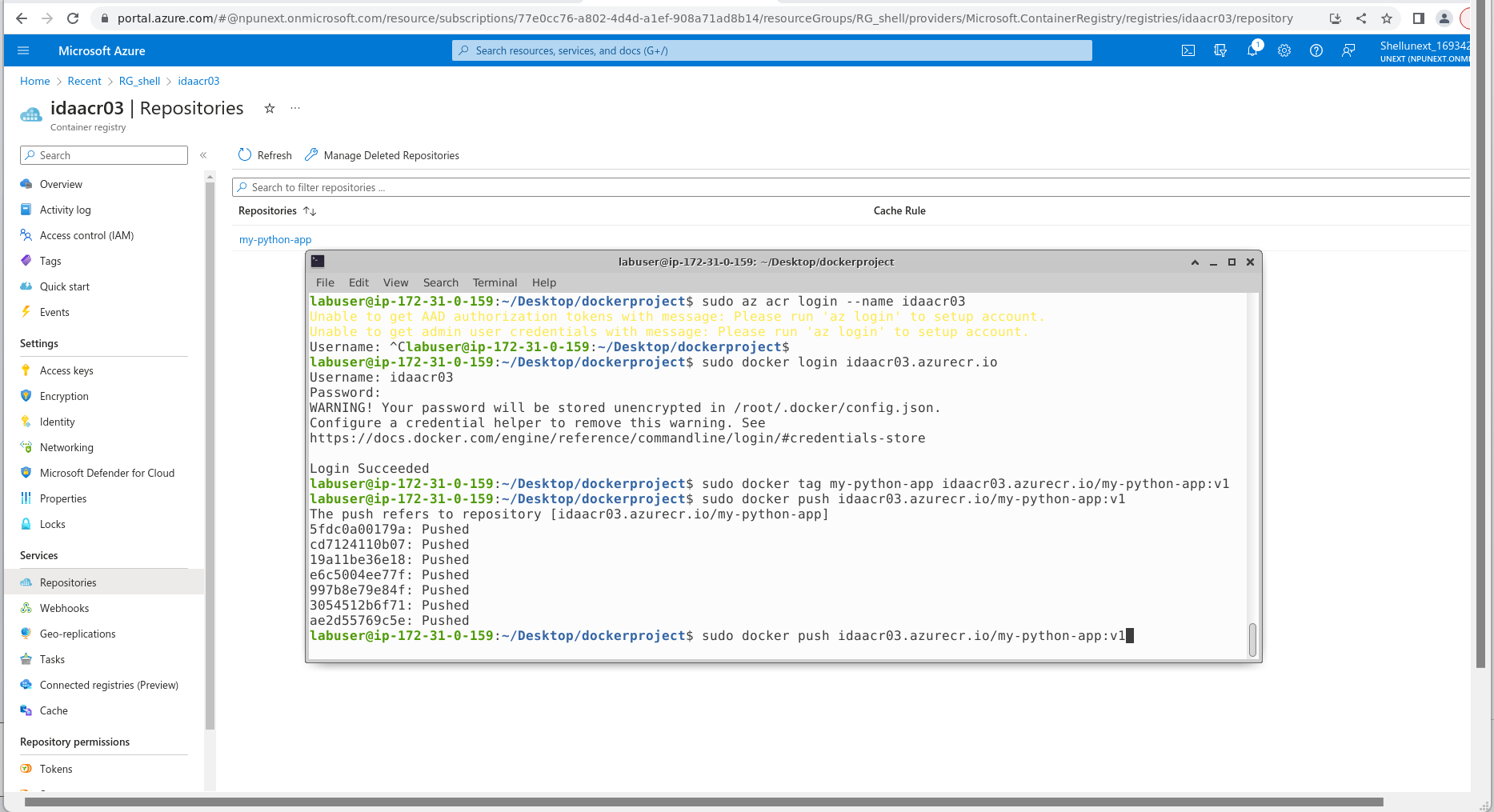
To create an image:

Sudo docker build -t app-name

Sudo docker run app-name



We use Azure CLI or Azure Container Registry(ACR) to push the docker image to the cloud.



Steps to Push the Created Docker Image into Azure Container Registry:

Install Azure CLI:



Az login:

Az login -u username -p password or az login

Az RG:



Az ACR:



Az ACR login:



For username: name of the acr

For password: go to access keys of azure acr in the portal-> click on admin user, copy the password

Tag:



Push:



Kubernetes:

In Azure Portal, create a Kubernetes Cluster: kubecluster03

* Create a folder called Kubernetes in the VM
* Add two yaml files: deployment.yaml, and service.yaml
* Add the content in the yaml files
* Run the kube control(kubectl) command from inside the Kubernetes directory for deployment.yaml

**kubectl apply -f deployment.yaml**

* Run the kube control command from inside the Kubernetes directory for deployment.yaml

**kubectl apply -f service.yaml**

* attach the kuberneted cluster with the acr

**az aks get-credentials --resource-group RG\_shell --name kubecluster03**

**az aks update -n manikube -g RG\_shell --attach-acr idaacr03**

* Get the IP address of the Kubernetes cluster

**kubectl get svc my-python-app**

or

use the Azure Portal to connect the image to the Kubernetes Cluster

* create a starter application
* create a single image application

view application



