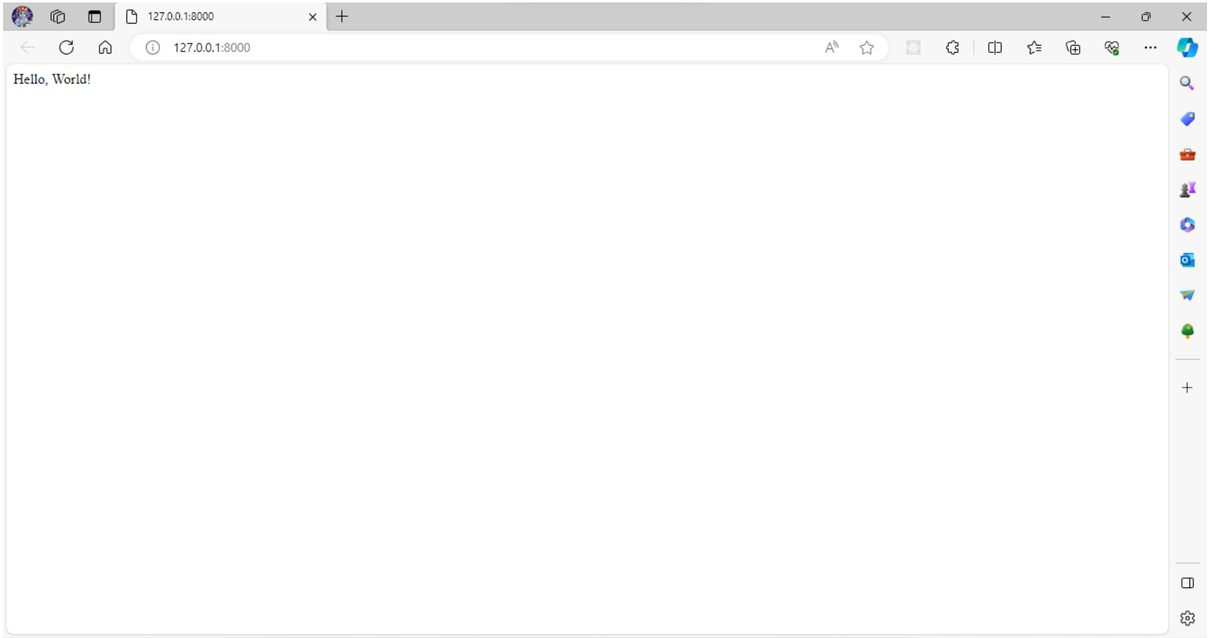
Efficient Django Deployment: Dockerized Application on Azure VM

Submitted by – Aditya Bhavesh Shah

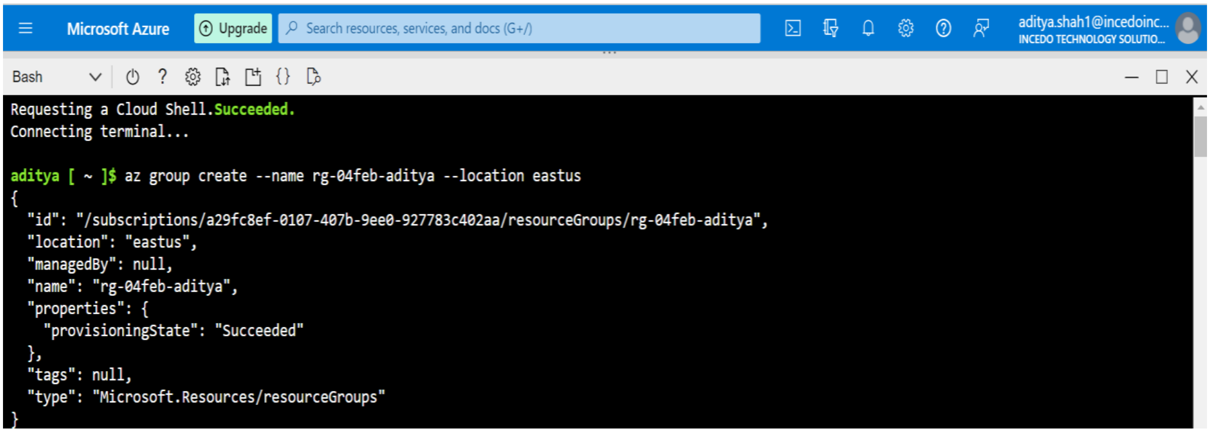
Task 1 Django Simple Hello World Application (run on local machine) –



Hello, World!

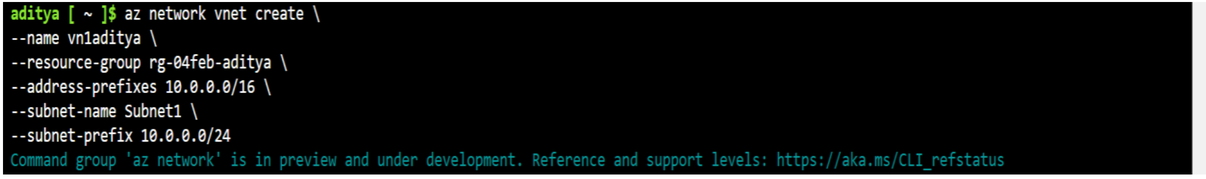
Task 2 Order & Connect Azure VM

Step 1 Create a Resource Group –

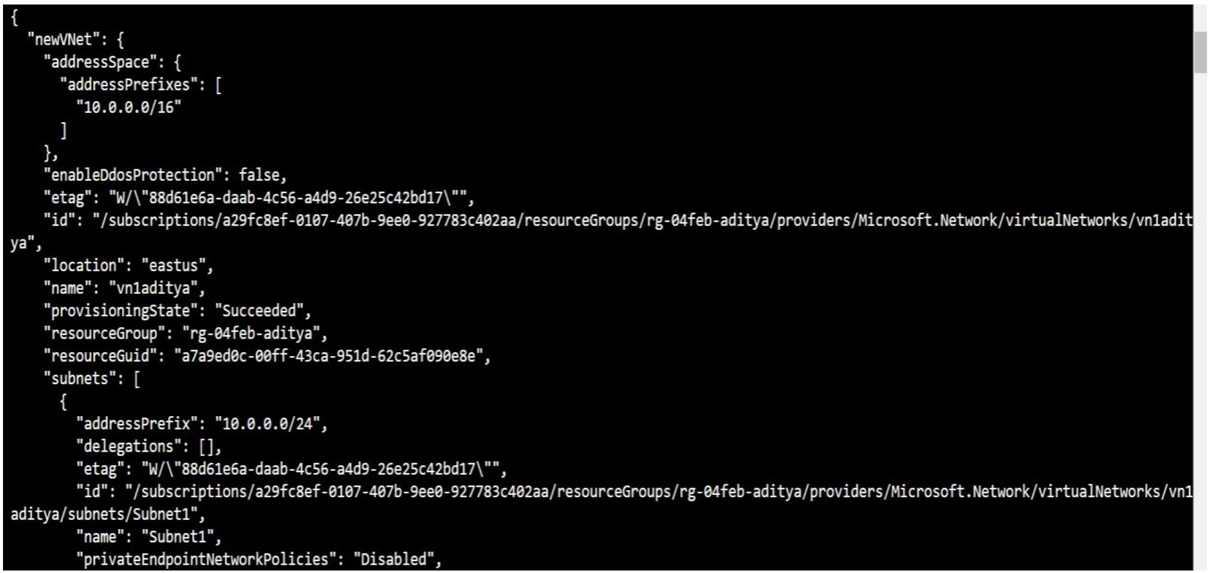

Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
aditya.shah1@incedoinc...
INCEDO TECHNOLOGY SOLUTIO...
Bash

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...

aditya [ ~ ]$ az group create --name rg-04feb-aditya --location eastus
{
  "id": "/subscriptions/a29fc8ef-0107-407b-9ee0-927783c402aa/resourceGroups/rg-04feb-aditya",
  "location": "eastus",
  "managedBy": null,
  "name": "rg-04feb-aditya",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
```

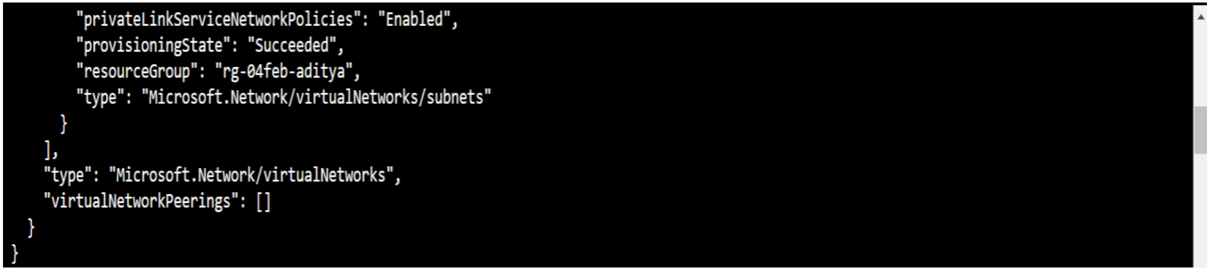
Step 2 Create Vnet –



```
aditya [ ~ ]$ az network vnet create \
--name vnladitya \
--resource-group rg-04feb-aditya \
--address-prefixes 10.0.0.0/16 \
--subnet-name Subnet1 \
--subnet-prefix 10.0.0.0/24
Command group 'az network' is in preview and under development. Reference and support levels: https://aka.ms/CLI_refstatus
```

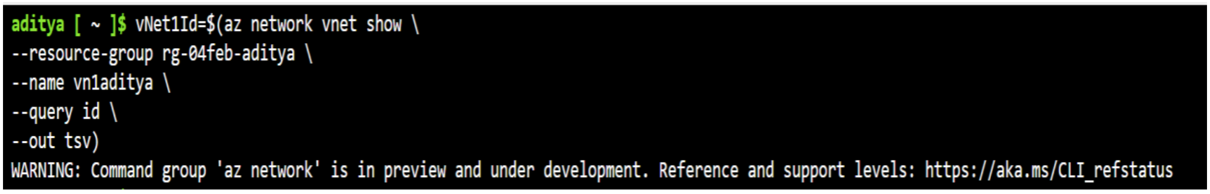


```
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/16"
      ]
    },
    "enableDdosProtection": false,
    "etag": "W/\"88d61e6a-daab-4c56-a4d9-26e25c42bd17\"",
    "id": "/subscriptions/a29fc8ef-0107-407b-9ee0-927783c402aa/resourceGroups/rg-04feb-aditya/providers/Microsoft.Network/virtualNetworks/vn1adit
ya",
    "location": "eastus",
    "name": "vn1aditya",
    "provisioningState": "Succeeded",
    "resourceGroup": "rg-04feb-aditya",
    "resourceGuid": "a7a9ed0c-00ff-43ca-951d-62c5af090e8e",
    "subnets": [
      {
        "addressPrefix": "10.0.0.0/24",
        "delegations": [],
        "etag": "W/\"88d61e6a-daab-4c56-a4d9-26e25c42bd17\"",
        "id": "/subscriptions/a29fc8ef-0107-407b-9ee0-927783c402aa/resourceGroups/rg-04feb-aditya/providers/Microsoft.Network/virtualNetworks/vn1
aditya/subnets/Subnet1",
        "name": "Subnet1",
        "privateEndpointNetworkPolicies": "Disabled",
```



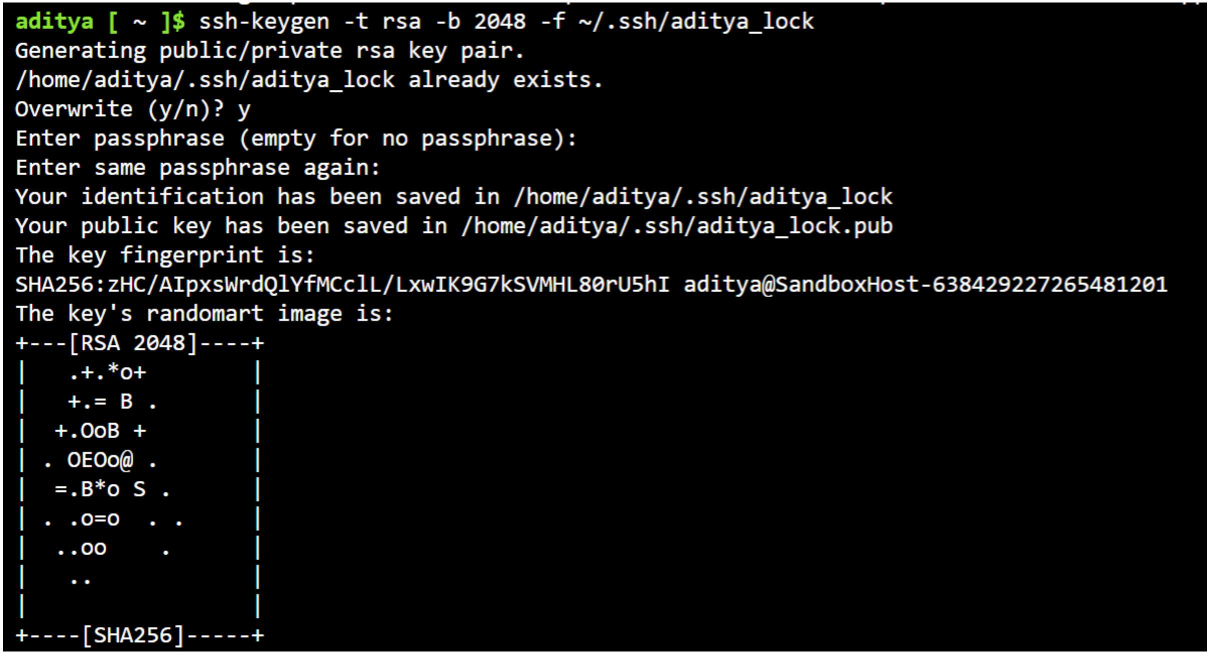
```
        "privateLinkServiceNetworkPolicies": "Enabled",
        "provisioningState": "Succeeded",
        "resourceGroup": "rg-04feb-aditya",
        "type": "Microsoft.Network/virtualNetworks/subnets"
      }
    ],
    "type": "Microsoft.Network/virtualNetworks",
    "virtualNetworkPeerings": []
  }
}
```

Step 3 Create Vnet Id -



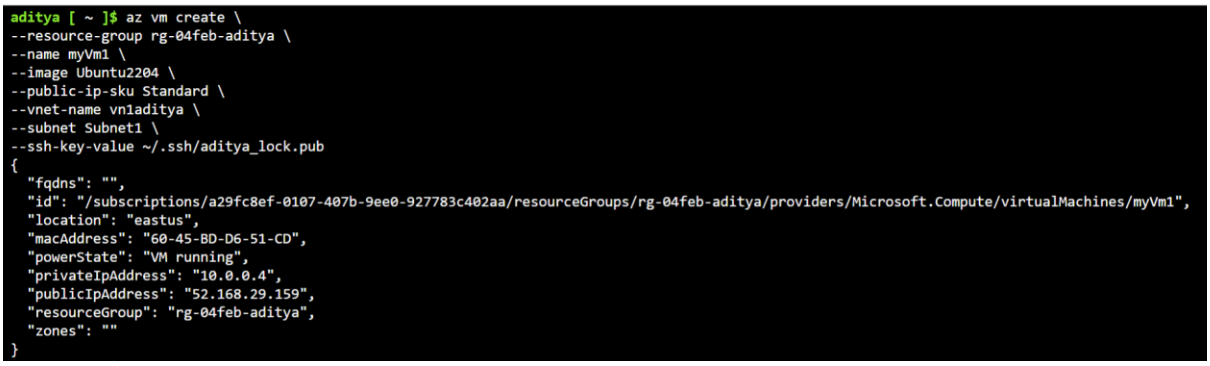
```
aditya [ ~ ]$ vNet1Id=$(az network vnet show \
--resource-group rg-04feb-aditya \
--name vn1aditya \
--query id \
--out tsv)
WARNING: Command group 'az network' is in preview and under development. Reference and support levels: https://aka.ms/CLI_refstatus
```

Step 4 Create a SSH Key



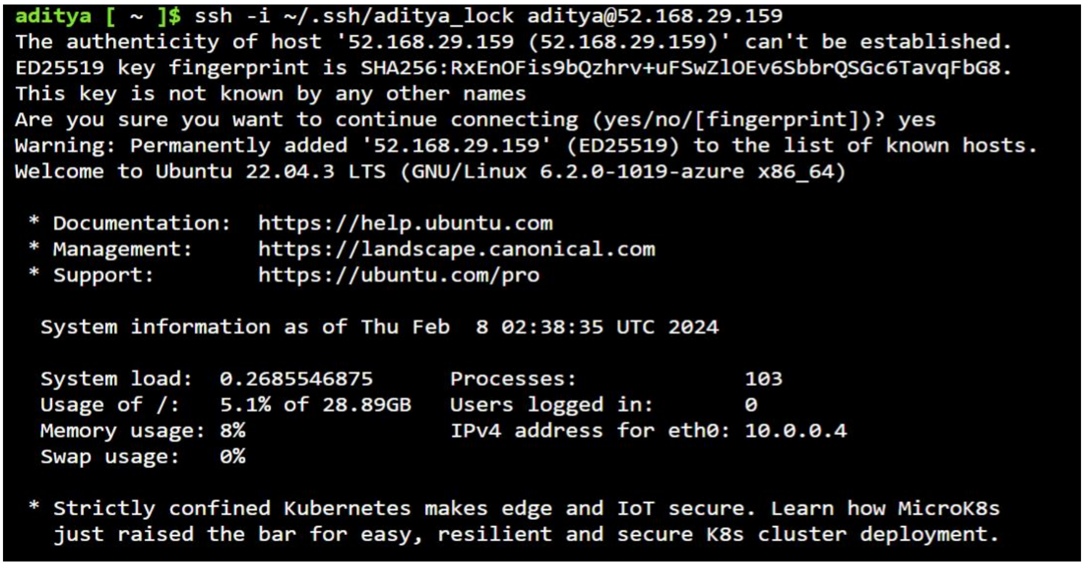
```
aditya [ ~ ]$ ssh-keygen -t rsa -b 2048 -f ~/.ssh/aditya_lock
Generating public/private rsa key pair.
/home/aditya/.ssh/aditya_lock already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/aditya/.ssh/aditya_lock
Your public key has been saved in /home/aditya/.ssh/aditya_lock.pub
The key fingerprint is:
SHA256:zHC/AIpxsWrdQlYfMCclL/LxwIK9G7kSVMHL80rU5hI aditya@SandboxHost-638429227265481201
The key's randomart image is:
+---[RSA 2048]----+
|   .+.*o+        |
|   +.= B .       |
|  +.OoB +        |
| . OEOo@ .       |
|  =.B*o S .      |
| . .o=o  . .     |
|  ..oo    .      |
|   ..            |
|                 |
+----[SHA256]-----+
```

5 Create a VM

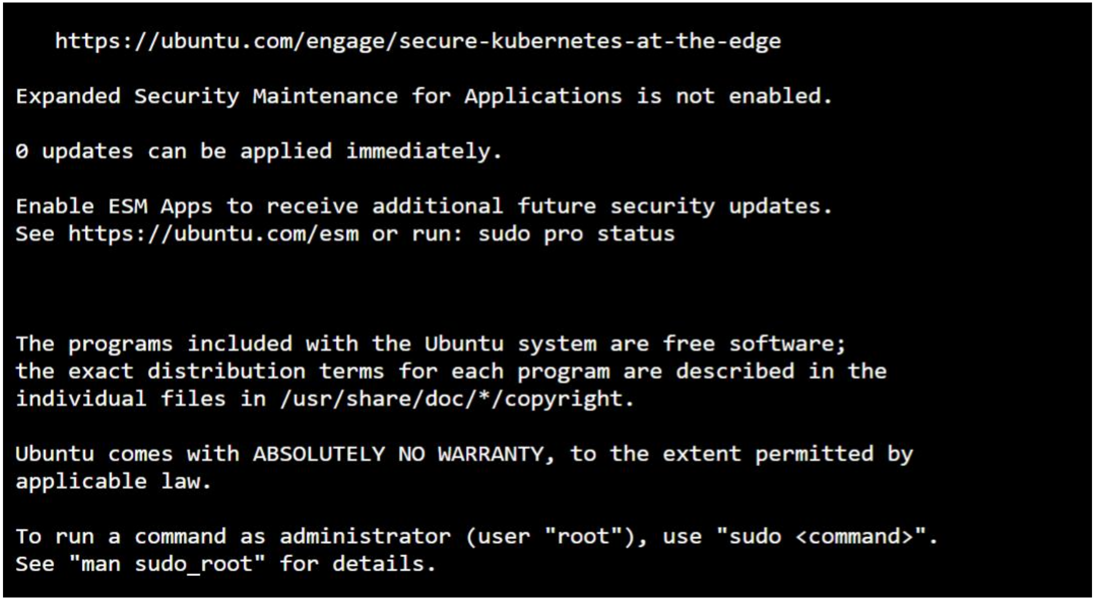


```
aditya [ ~ ]$ az vm create \
--resource-group rg-04feb-aditya \
--name myVm1 \
--image Ubuntu2204 \
--public-ip-sku Standard \
--vnet-name vn1aditya \
--subnet Subnet1 \
--ssh-key-value ~/.ssh/aditya_lock.pub
{
  "fqdns": "",
  "id": "/subscriptions/a29fc8ef-0107-407b-9ee0-927783c402aa/resourceGroups/rg-04feb-aditya/providers/Microsoft.Compute/virtualMachines/myVm1",
  "location": "eastus",
  "macAddress": "60-45-BD-D6-51-CD",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "52.168.29.159",
  "resourceGroup": "rg-04feb-aditya",
  "zones": ""
}
```

Step 6 SSH Call on VM with Public IP Address



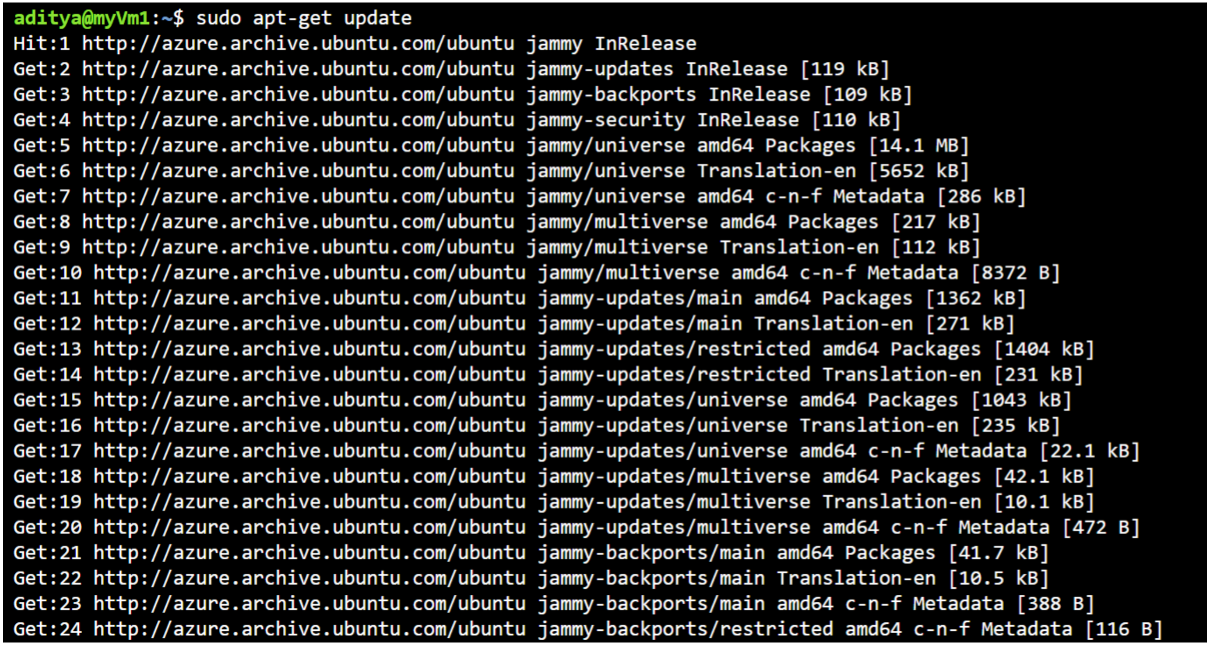
```
aditya [ ~ ]$ ssh -i ~/.ssh/aditya_lock aditya@52.168.29.159
The authenticity of host '52.168.29.159 (52.168.29.159)' can't be established.
ED25519 key fingerprint is SHA256:RxEnOFis9bQzhrv+uFSwZlOEv6SbbrQSGc6TavqFbG8.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.168.29.159' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1019-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

  System information as of Thu Feb  8 02:38:35 UTC 2024

  System load:  0.2685546875      Processes:             103
  Usage of /:   5.1% of 28.89GB   Users logged in:       0
  Memory usage: 8%                IPv4 address for eth0: 10.0.0.4
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.
```



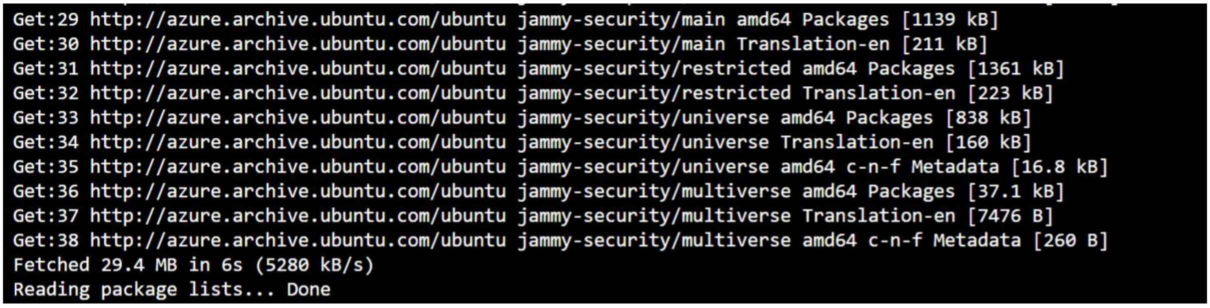
```
   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

Step 7 Installing Docker on VM

i) sudo apt-get update

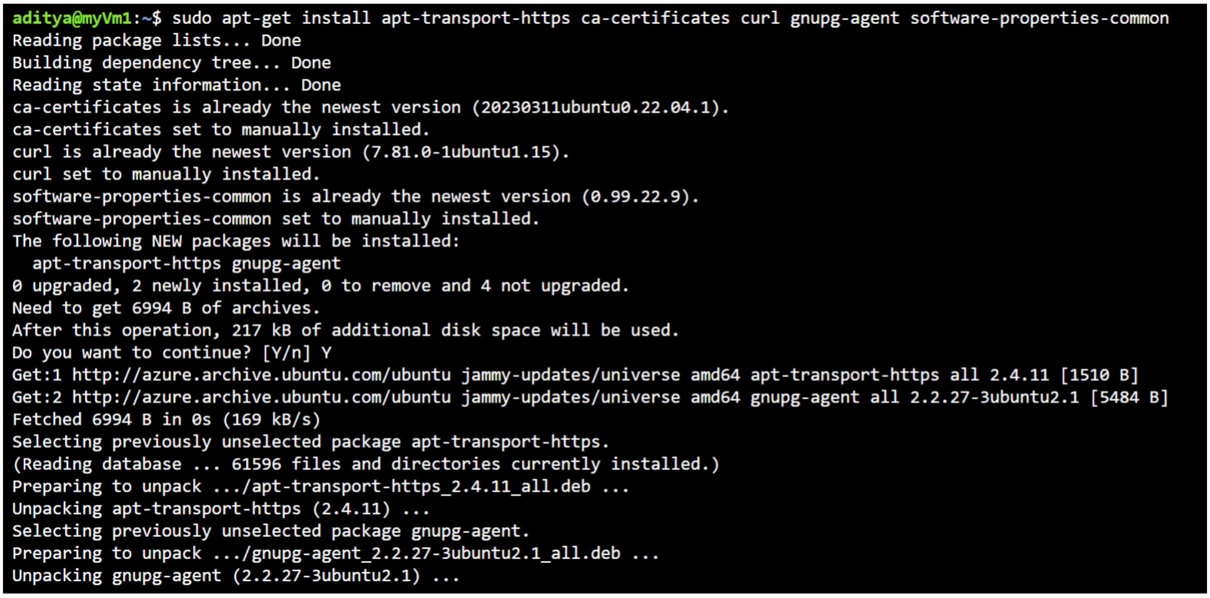


```
aditya@myVm1:~$ sudo apt-get update
Hit:1 http://azure.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://azure.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://azure.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1362 kB]
Get:12 http://azure.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [271 kB]
Get:13 http://azure.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [1404 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [231 kB]
Get:15 http://azure.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1043 kB]
Get:16 http://azure.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [235 kB]
Get:17 http://azure.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [22.1 kB]
Get:18 http://azure.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [42.1 kB]
Get:19 http://azure.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [10.1 kB]
Get:20 http://azure.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [472 B]
Get:21 http://azure.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [41.7 kB]
Get:22 http://azure.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [10.5 kB]
Get:23 http://azure.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:24 http://azure.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
```

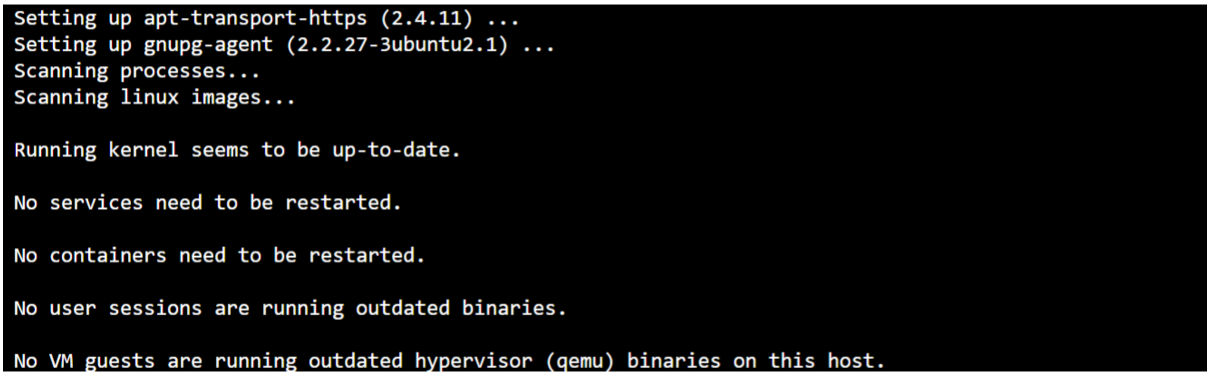


```
Get:29 http://azure.archive.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1139 kB]
Get:30 http://azure.archive.ubuntu.com/ubuntu jammy-security/main Translation-en [211 kB]
Get:31 http://azure.archive.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1361 kB]
Get:32 http://azure.archive.ubuntu.com/ubuntu jammy-security/restricted Translation-en [223 kB]
Get:33 http://azure.archive.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [838 kB]
Get:34 http://azure.archive.ubuntu.com/ubuntu jammy-security/universe Translation-en [160 kB]
Get:35 http://azure.archive.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:36 http://azure.archive.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [37.1 kB]
Get:37 http://azure.archive.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7476 B]
Get:38 http://azure.archive.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 29.4 MB in 6s (5280 kB/s)
Reading package lists... Done
```

ii) sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common

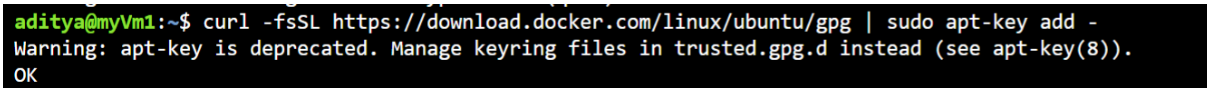


```
aditya@myVm1:~$ sudo apt-get install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ca-certificates is already the newest version (20230311ubuntu0.22.04.1).
ca-certificates set to manually installed.
curl is already the newest version (7.81.0-1ubuntu1.15).
curl set to manually installed.
software-properties-common is already the newest version (0.99.22.9).
software-properties-common set to manually installed.
The following NEW packages will be installed:
  apt-transport-https gnupg-agent
0 upgraded, 2 newly installed, 0 to remove and 4 not upgraded.
Need to get 6994 B of archives.
After this operation, 217 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://azure.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 apt-transport-https all 2.4.11 [1510 B]
Get:2 http://azure.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 gnupg-agent all 2.2.27-3ubuntu2.1 [5484 B]
Fetched 6994 B in 0s (169 kB/s)
Selecting previously unselected package apt-transport-https.
(Reading database ... 61596 files and directories currently installed.)
Preparing to unpack .../apt-transport-https_2.4.11_all.deb ...
Unpacking apt-transport-https (2.4.11) ...
Selecting previously unselected package gnupg-agent.
Preparing to unpack .../gnupg-agent_2.2.27-3ubuntu2.1_all.deb ...
Unpacking gnupg-agent (2.2.27-3ubuntu2.1) ...
```



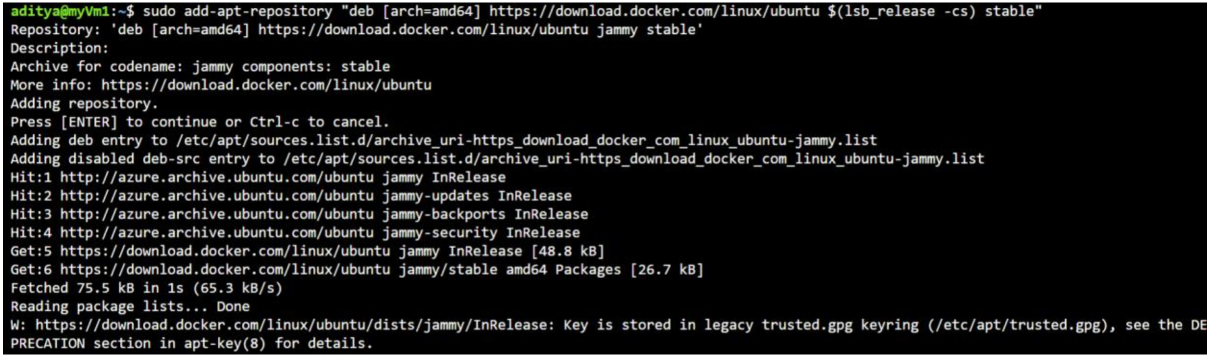
```
Setting up apt-transport-https (2.4.11) ...
Setting up gnupg-agent (2.2.27-3ubuntu2.1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

iii) curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -



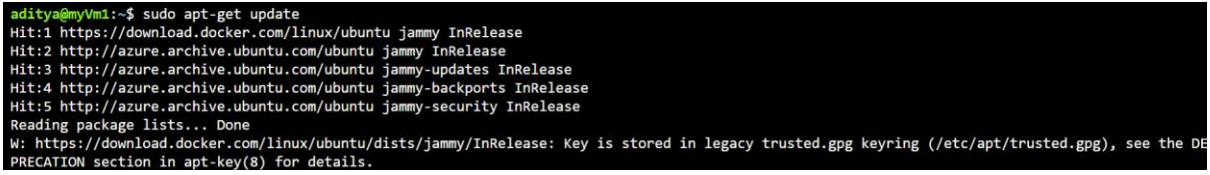
```
aditya@myVm1:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

iv) sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"



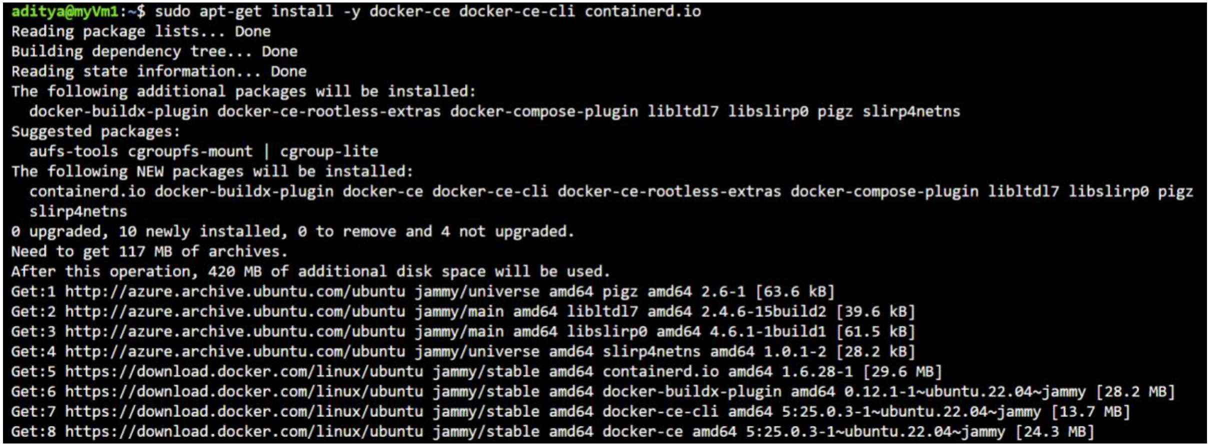
```
aditya@myVm1:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Repository: 'deb [arch=amd64] https://download.docker.com/linux/ubuntu jammy stable'
Description:
Archive for codename: jammy components: stable
More info: https://download.docker.com/linux/ubuntu
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-jammy.list
Hit:1 http://azure.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu jammy-security InRelease
Get:5 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [26.7 kB]
Fetched 75.5 kB in 1s (65.3 kB/s)
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DE
PRECATION section in apt-key(8) for details.
```

v) sudo apt-get update

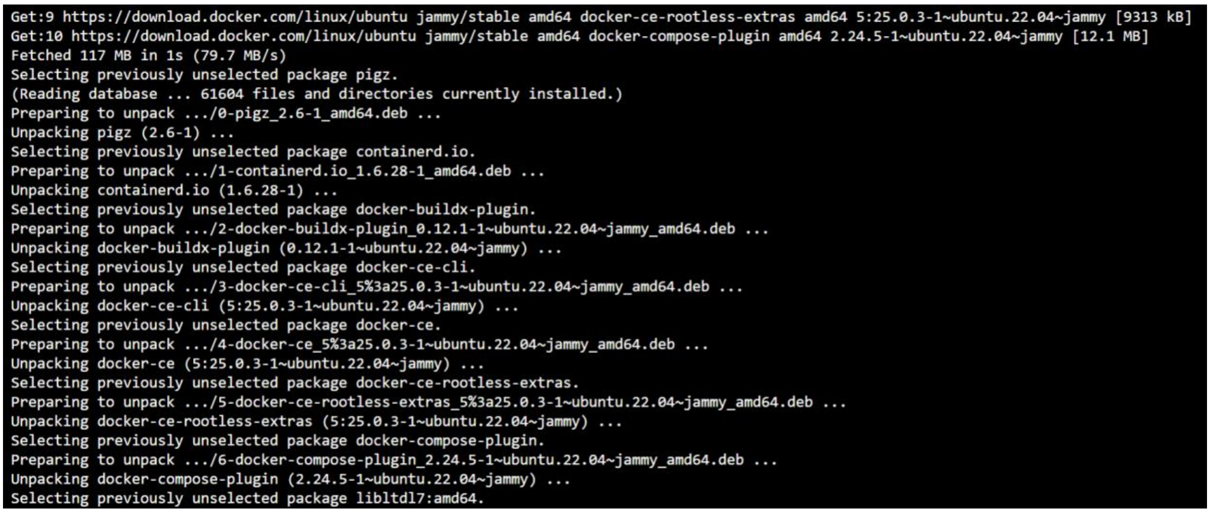


```
aditya@myVm1:~$ sudo apt-get update
Hit:1 https://download.docker.com/linux/ubuntu jammy InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:5 http://azure.archive.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
W: https://download.docker.com/linux/ubuntu/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DE
PRECATION section in apt-key(8) for details.
```

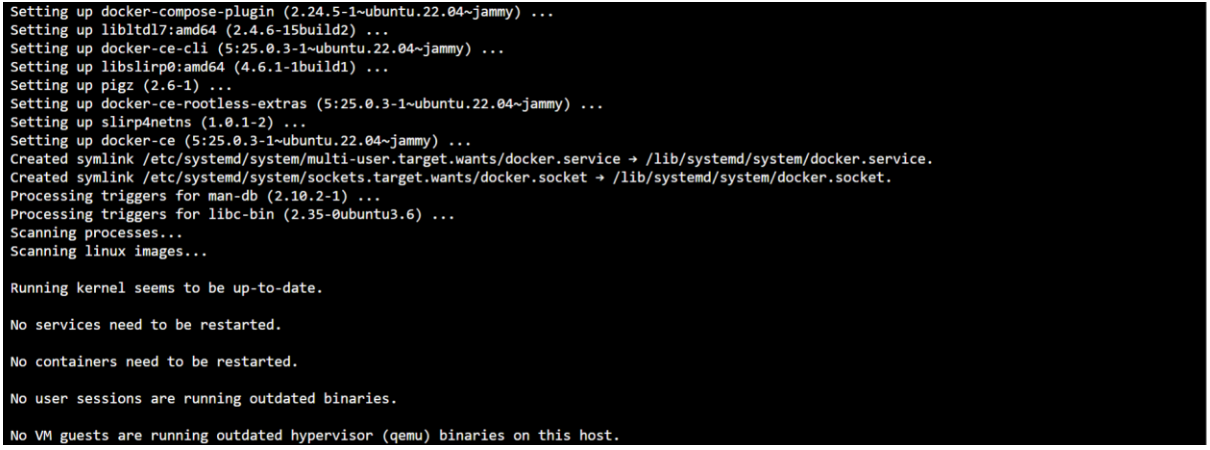
vi) sudo apt-get install -y docker-ce docker-ce-cli containerd.io



```
aditya@myVm1:~$ sudo apt-get install -y docker-ce docker-ce-cli containerd.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-buildx-plugin docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-buildx-plugin docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin libltdl7 libslirp0 pigz
  slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 4 not upgraded.
Need to get 117 MB of archives.
After this operation, 420 MB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu jammy/main amd64 libltdl7 amd64 2.4.6-15build2 [39.6 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1build1 [61.5 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu jammy/universe amd64 slirp4netns amd64 1.0.1-2 [28.2 kB]
Get:5 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.28-1 [29.6 MB]
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-buildx-plugin amd64 0.12.1-1~ubuntu.22.04~jammy [28.2 MB]
Get:7 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:25.0.3-1~ubuntu.22.04~jammy [13.7 MB]
Get:8 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:25.0.3-1~ubuntu.22.04~jammy [24.3 MB]
```



```
Get:9 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-rootless-extras amd64 5:25.0.3-1~ubuntu.22.04~jammy [9313 kB]
Get:10 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-compose-plugin amd64 2.24.5-1~ubuntu.22.04~jammy [12.1 MB]
Fetched 117 MB in 1s (79.7 MB/s)
Selecting previously unselected package pigz.
(Reading database ... 61604 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.6-1_amd64.deb ...
Unpacking pigz (2.6-1) ...
Selecting previously unselected package containerd.io.
Preparing to unpack .../1-containerd.io_1.6.28-1_amd64.deb ...
Unpacking containerd.io (1.6.28-1) ...
Selecting previously unselected package docker-buildx-plugin.
Preparing to unpack .../2-docker-buildx-plugin_0.12.1-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-buildx-plugin (0.12.1-1~ubuntu.22.04~jammy) ...
Selecting previously unselected package docker-ce-cli.
Preparing to unpack .../3-docker-ce-cli_5%3a25.0.3-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-ce-cli (5:25.0.3-1~ubuntu.22.04~jammy) ...
Selecting previously unselected package docker-ce.
Preparing to unpack .../4-docker-ce_5%3a25.0.3-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-ce (5:25.0.3-1~ubuntu.22.04~jammy) ...
Selecting previously unselected package docker-ce-rootless-extras.
Preparing to unpack .../5-docker-ce-rootless-extras_5%3a25.0.3-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-ce-rootless-extras (5:25.0.3-1~ubuntu.22.04~jammy) ...
Selecting previously unselected package docker-compose-plugin.
Preparing to unpack .../6-docker-compose-plugin_2.24.5-1~ubuntu.22.04~jammy_amd64.deb ...
Unpacking docker-compose-plugin (2.24.5-1~ubuntu.22.04~jammy) ...
Selecting previously unselected package libltdl7:amd64.
```

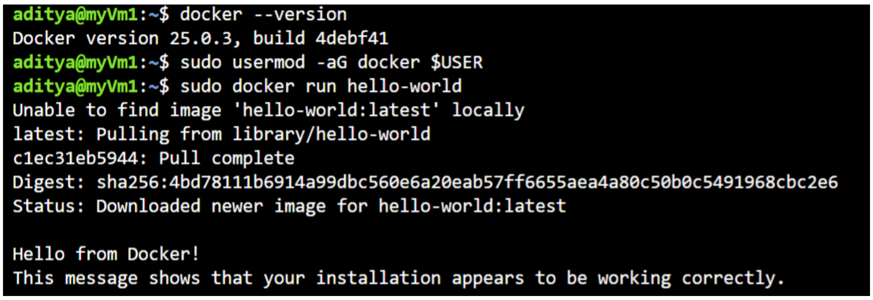


```
Setting up docker-compose-plugin (2.24.5-1~ubuntu.22.04~jammy) ...
Setting up libltdl7:amd64 (2.4.6-15build2) ...
Setting up docker-ce-cli (5:25.0.3-1~ubuntu.22.04~jammy) ...
Setting up libslirp0:amd64 (4.6.1-1build1) ...
Setting up pigz (2.6-1) ...
Setting up docker-ce-rootless-extras (5:25.0.3-1~ubuntu.22.04~jammy) ...
Setting up slirp4netns (1.0.1-2) ...
Setting up docker-ce (5:25.0.3-1~ubuntu.22.04~jammy) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

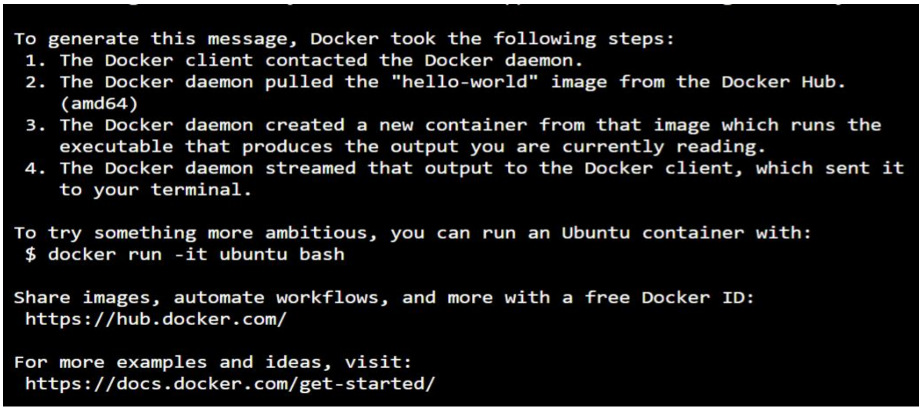
vii) docker –version

viii) sudo usermod -aG docker $USER

ix) sudo docker run hello-world



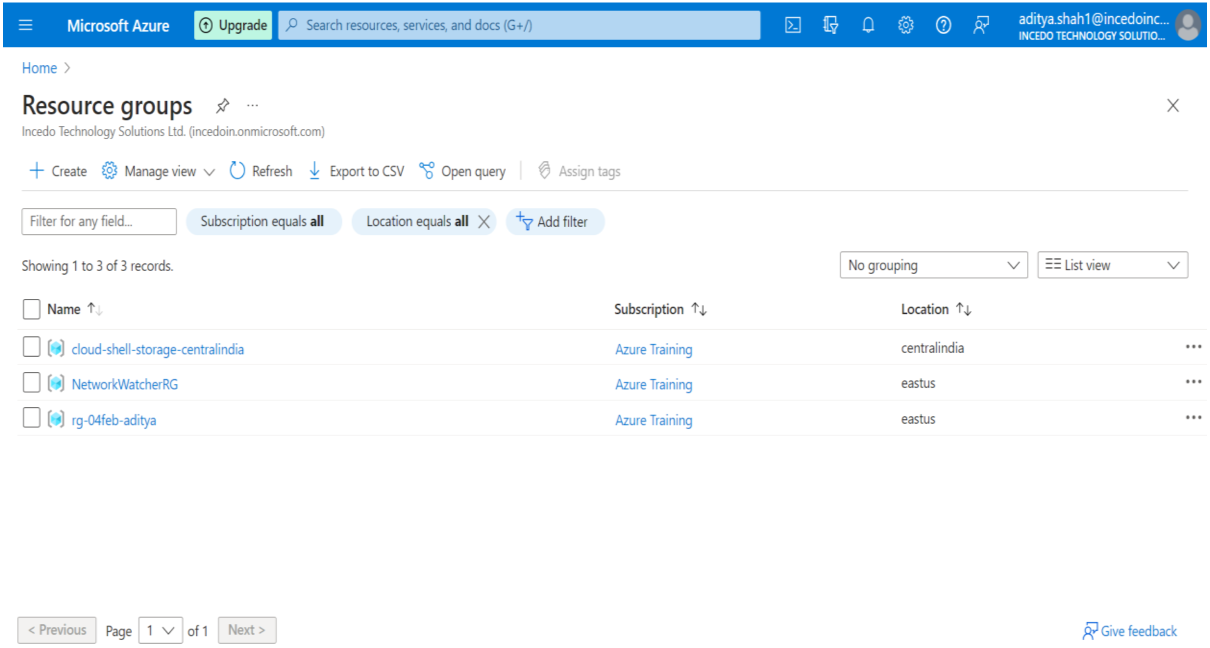
```
aditya@myVm1:~$ docker --version
Docker version 25.0.3, build 4debf41
aditya@myVm1:~$ sudo usermod -aG docker $USER
aditya@myVm1:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:4bd78111b6914a99dbc560e6a20eab57ff6655aea4a80c50b0c5491968cbc2e6
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```



```
To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Output –

Resource Group Created –



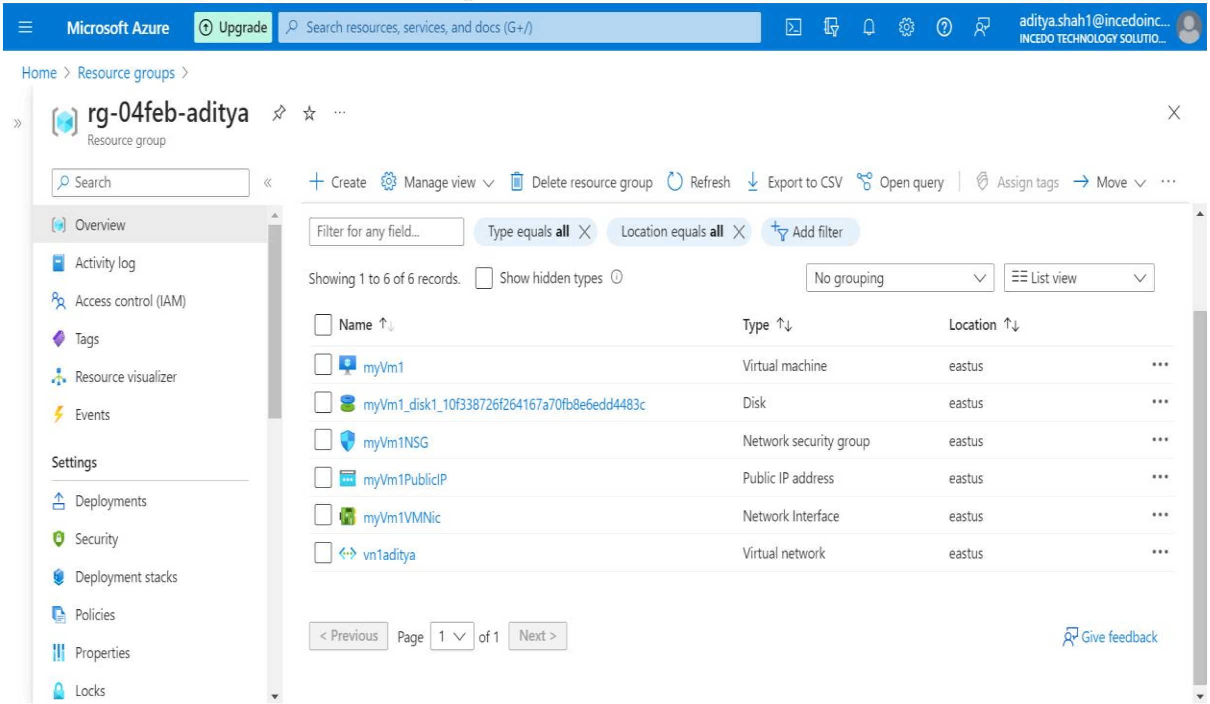
Resource groups

Incedo Technology Solutions Ltd. (incedoin.onmicrosoft.com)

Showing 1 to 3 of 3 records.

Name	Subscription	Location
cloud-shell-storage-centralindia	Azure Training	centralindia
NetworkWatcherRG	Azure Training	eastus
rg-04feb-aditya	Azure Training	eastus

Vnet, VM created in Resource Group –



rg-04feb-aditya

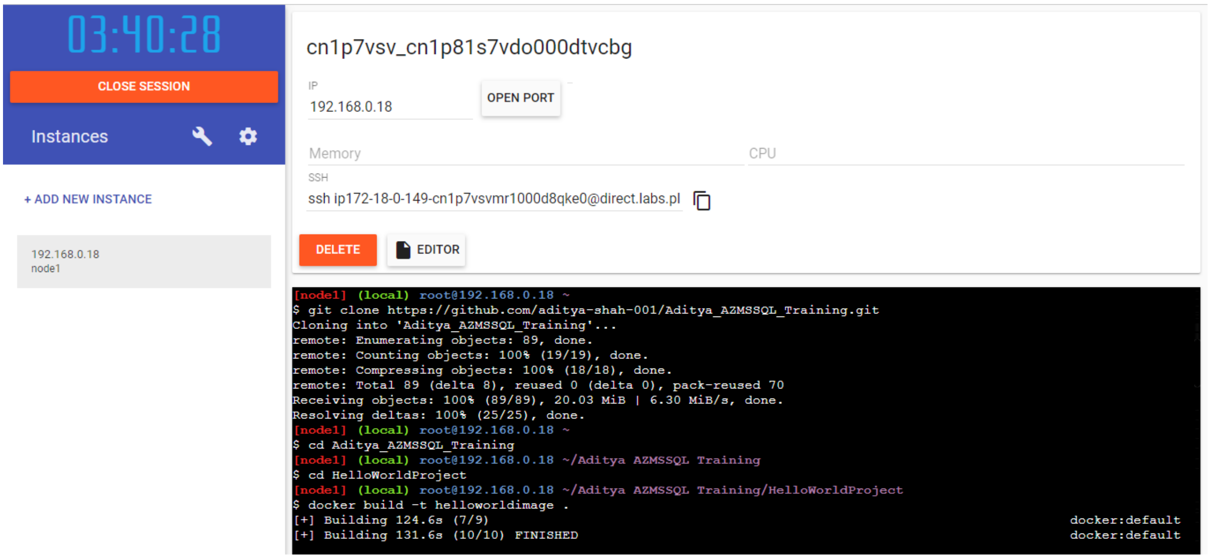
Resource group

Showing 1 to 6 of 6 records.

Name	Type	Location
myVm1	Virtual machine	eastus
myVm1_disk1_10f338726f264167a70fb8e6edd4483c	Disk	eastus
myVm1NSG	Network security group	eastus
myVm1PublicIP	Public IP address	eastus
myVm1VMNic	Network Interface	eastus
vn1aditya	Virtual network	eastus

Task 3 Django Application Dockerization using Docker Playground:

Step 1 git clone https://github.com/aditya-shah-001/Aditya_AZMSSQL_Training.git



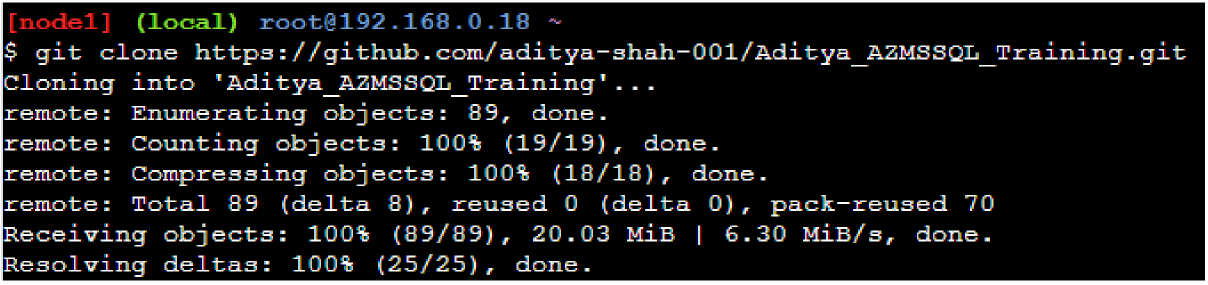
cn1p7vsv_cn1p81s7vdo000dtvcbg

IP 192.168.0.18

SSH ssh ip172-18-0-149-cn1p7vsvmr1000d8qke0@direct.labs.pl

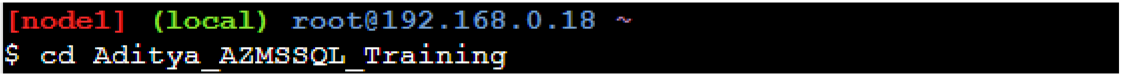
192.168.0.18
node1

```
[node1] (local) root@192.168.0.18 ~
$ git clone https://github.com/aditya-shah-001/Aditya_AZMSSQL_Training.git
Cloning into 'Aditya_AZMSSQL_Training'...
remote: Enumerating objects: 89, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 89 (delta 8), reused 0 (delta 0), pack-reused 70
Receiving objects: 100% (89/89), 20.03 MiB | 6.30 MiB/s, done.
Resolving deltas: 100% (25/25), done.
[node1] (local) root@192.168.0.18 ~
$ cd Aditya_AZMSSQL_Training
[node1] (local) root@192.168.0.18 ~/Aditya AZMSSQL Training
$ cd HelloWorldProject
[node1] (local) root@192.168.0.18 ~/Aditya AZMSSQL Training/HelloWorldProject
$ docker build -t helloworldimage .
[+] Building 124.6s (7/9)                                                docker:default
[+] Building 131.6s (10/10) FINISHED                                     docker:default
```



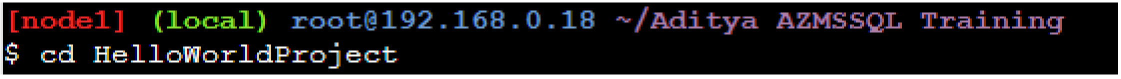
```
[node1] (local) root@192.168.0.18 ~
$ git clone https://github.com/aditya-shah-001/Aditya_AZMSSQL_Training.git
Cloning into 'Aditya_AZMSSQL_Training'...
remote: Enumerating objects: 89, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 89 (delta 8), reused 0 (delta 0), pack-reused 70
Receiving objects: 100% (89/89), 20.03 MiB | 6.30 MiB/s, done.
Resolving deltas: 100% (25/25), done.
```

Step 2 Navigating to the Repository - cd Aditya_AZMSSQL_Training



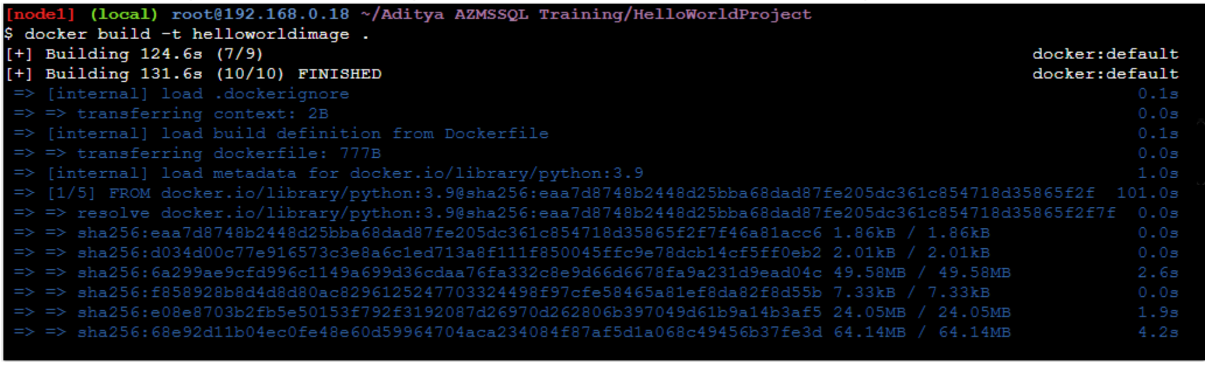
```
[node1] (local) root@192.168.0.18 ~
$ cd Aditya_AZMSSQL_Training
```

Step 3 Navigating to the Folder - cd HelloWorldProject

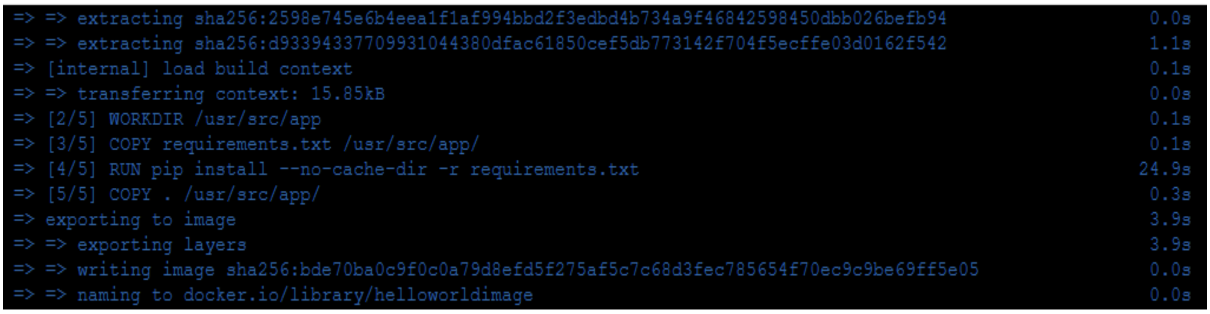


```
[node1] (local) root@192.168.0.18 ~/Aditya AZMSSQL Training
$ cd HelloWorldProject
```

Step 4 Building the Docker Image - docker build –t helloworldimage

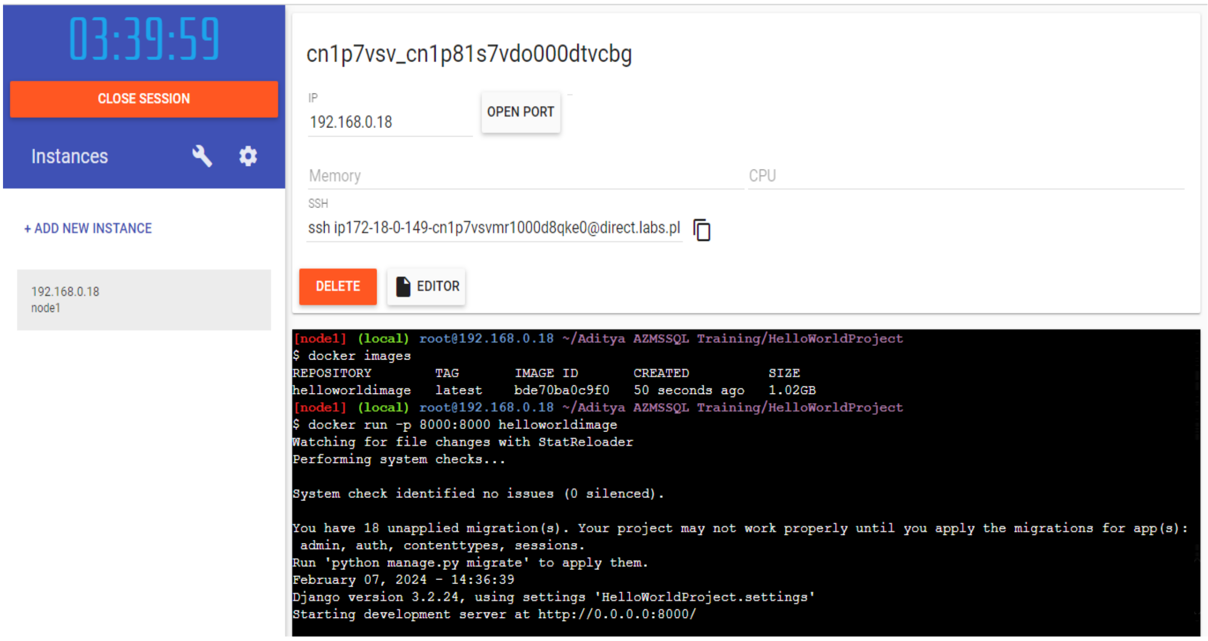


```
[node1] (local) root@192.168.0.18 ~/Aditya AZMSSQL Training/HelloWorldProject
$ docker build -t helloworldimage .
[+] Building 124.6s (7/9)                                                       docker:default
[+] Building 131.6s (10/10) FINISHED                                            docker:default
 => [internal] load .dockerignore                                                        0.1s
 => => transferring context: 2B                                                          0.0s
 => [internal] load build definition from Dockerfile                                     0.1s
 => => transferring dockerfile: 777B                                                     0.0s
 => [internal] load metadata for docker.io/library/python:3.9                            1.0s
 => [1/5] FROM docker.io/library/python:3.9@sha256:eaa7d8748b2448d25bba68dad87fe205dc361c854718d35865f2f  101.0s
 => => resolve docker.io/library/python:3.9@sha256:eaa7d8748b2448d25bba68dad87fe205dc361c854718d35865f2f7f  0.0s
 => => sha256:eaa7d8748b2448d25bba68dad87fe205dc361c854718d35865f2f7f46a81acc6 1.86kB / 1.86kB    0.0s
 => => sha256:d034d00c77e916573c3e8a6c1ed713a8f111f850045ffc9e78dcb14cf5ff0eb2 2.01kB / 2.01kB    0.0s
 => => sha256:6a299ae9cfd996c1149a699d36cdaa76fa332c8e9d66d6678fa9a231d9ead04c 49.58MB / 49.58MB  2.6s
 => => sha256:f858928b8d4d8d80ac8296125247703324498f97cfe58465a81ef8da82f8d55b 7.33kB / 7.33kB    0.0s
 => => sha256:e08e8703b2fb5e50153f792f3192087d26970d262806b397049d61b9a14b3af5 24.05MB / 24.05MB  1.9s
 => => sha256:68e92d11b04ec0fe48e60d59964704aca234084f87af5d1a068c49456b37fe3d 64.14MB / 64.14MB  4.2s
```



```
=> => extracting sha256:2598e745e6b4eea1f1af994bbd2f3edbd4b734a9f46842598450dbb026befb94        0.0s
=> => extracting sha256:d93394337709931044380dfac61850cef5db773142f704f5ecffe03d0162f542       1.1s
=> [internal] load build context                                                                0.1s
=> => transferring context: 15.85kB                                                             0.0s
=> [2/5] WORKDIR /usr/src/app                                                                   0.1s
=> [3/5] COPY requirements.txt /usr/src/app/                                                    0.1s
=> [4/5] RUN pip install --no-cache-dir -r requirements.txt                                    24.9s
=> [5/5] COPY . /usr/src/app/                                                                   0.3s
=> exporting to image                                                                           3.9s
=> => exporting layers                                                                          3.9s
=> => writing image sha256:bde70ba0c9f0c0a79d8efd5f275af5c7c68d3fec785654f70ec9c9be69ff5e05     0.0s
=> => naming to docker.io/library/helloworldimage                                               0.0s
```

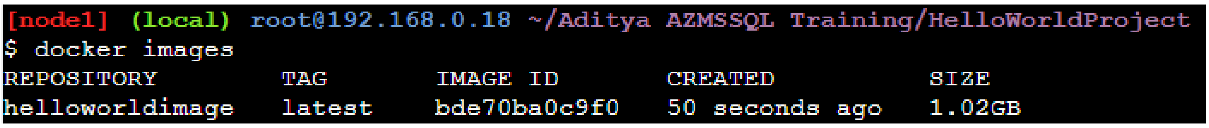
Docker Image Created –



IP: 192.168.0.18

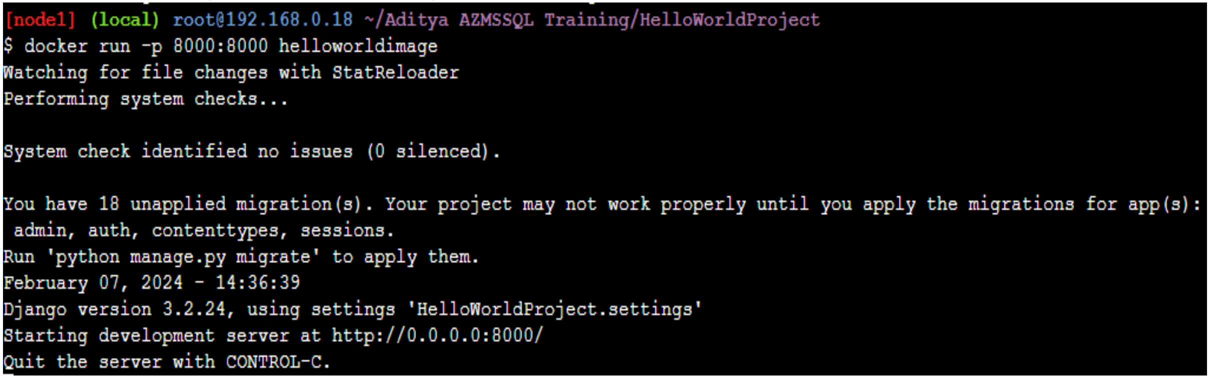
SSH: ssh ip172-18-0-149-cn1p7vsvmr1000d8qke0@direct.labs.pl

```
[node1] (local) root@192.168.0.18 ~/Aditya AZMSSQL Training/HelloWorldProject
$ docker images
REPOSITORY        TAG       IMAGE ID       CREATED          SIZE
helloworldimage   latest    bde70ba0c9f0   50 seconds ago   1.02GB
[node1] (local) root@192.168.0.18 ~/Aditya AZMSSQL Training/HelloWorldProject
$ docker run -p 8000:8000 helloworldimage
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s):
 admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 07, 2024 - 14:36:39
Django version 3.2.24, using settings 'HelloWorldProject.settings'
Starting development server at http://0.0.0.0:8000/
```



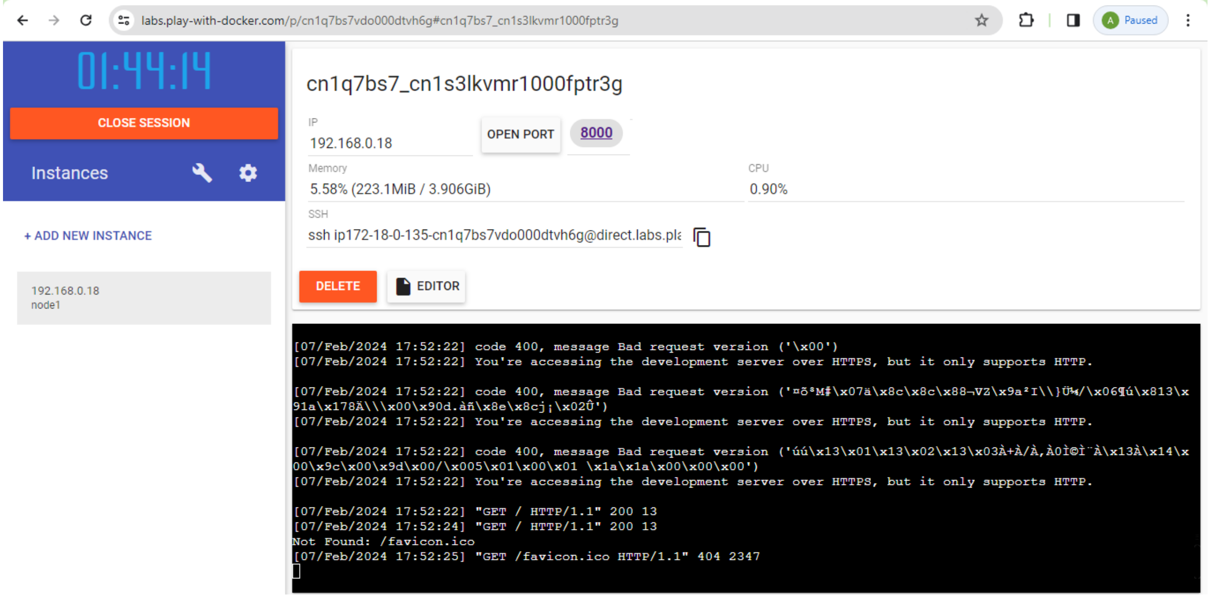
```
[node1] (local) root@192.168.0.18 ~/Aditya AZMSSQL Training/HelloWorldProject
$ docker images
REPOSITORY        TAG       IMAGE ID       CREATED          SIZE
helloworldimage   latest    bde70ba0c9f0   50 seconds ago   1.02GB
```

Step 5 Executing the Docker Image –



```
[node1] (local) root@192.168.0.18 ~/Aditya AZMSSQL Training/HelloWorldProject
$ docker run -p 8000:8000 helloworldimage
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s):
 admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 07, 2024 - 14:36:39
Django version 3.2.24, using settings 'HelloWorldProject.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

Step 6 Open Port 8000 for the executed Image –



cn1q7bs7_cn1s3lkvmr1000fptr3g

IP: 192.168.0.18

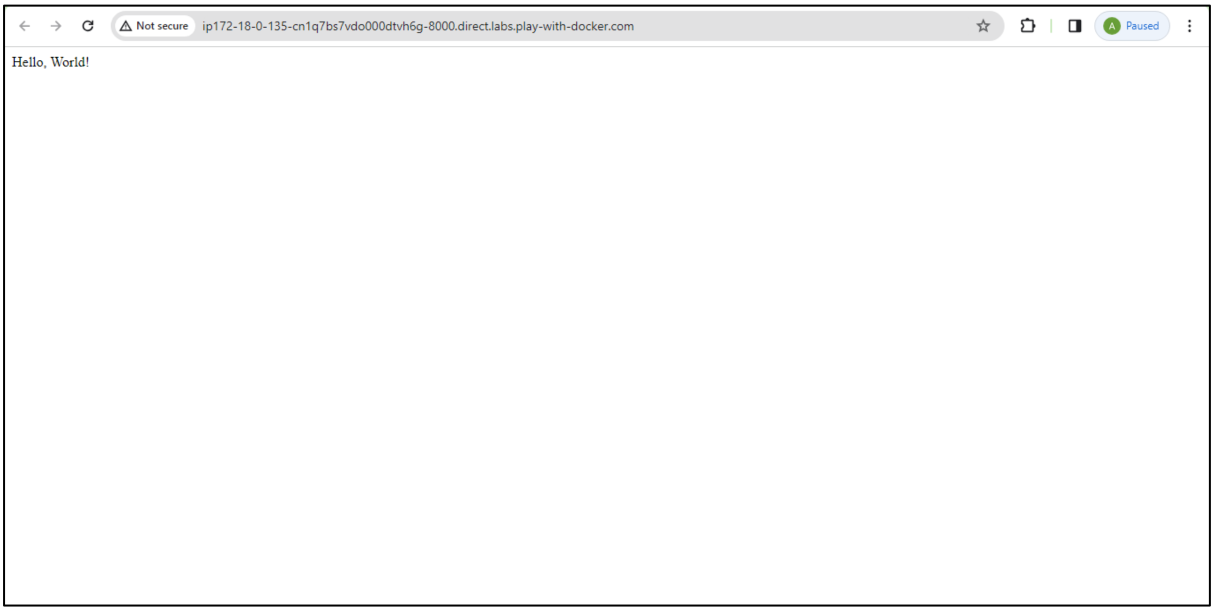
Memory: 5.58% (223.1MiB / 3.906GiB)

CPU: 0.90%

SSH: ssh ip172-18-0-135-cn1q7bs7vdo000dtvh6g@direct.labs.pl

```
[07/Feb/2024 17:52:22] code 400, message Bad request version ('\x00')
[07/Feb/2024 17:52:22] You're accessing the development server over HTTPS, but it only supports HTTP.

[07/Feb/2024 17:52:22] code 400, message Bad request version ('¤õ³M‡\x07ä\x8c\x8c\x88¬VZ\x9a²I\\}Ü¼/\x06¶ú\x813\x
91a\x178Ä\\\x00\x90d.àñ\x8e\x8cj¡\x02Û')
[07/Feb/2024 17:52:22] You're accessing the development server over HTTPS, but it only supports HTTP.

[07/Feb/2024 17:52:22] code 400, message Bad request version ('úú\x13\x01\x13\x02\x13\x03À+À/À,À0Ì©Ì¨À\x13À\x14\x
00\x9c\x00\x9d\x00/\x005\x01\x00\x01 \x1a\x1a\x00\x00\x00')
[07/Feb/2024 17:52:22] You're accessing the development server over HTTPS, but it only supports HTTP.

[07/Feb/2024 17:52:22] "GET / HTTP/1.1" 200 13
[07/Feb/2024 17:52:24] "GET / HTTP/1.1" 200 13
Not Found: /favicon.ico
[07/Feb/2024 17:52:25] "GET /favicon.ico HTTP/1.1" 404 2347
```

Step 7 Docker Image running on 8000 Port –


Not secure
ip172-18-0-135-cn1q7bs7vdo000dtvh6g-8000.direct.labs.play-with-docker.com
Paused

Hello, World!