



# LAB: DEPLOYING DOCKER



## STEP 1: CREATE A VIRTUAL MACHINE

1. Log in to your Azure Portal. Then go to create resources.
2. Now you need to create a virtual machine which should be based on Linux.
3. For that follow the images below (if you know how to deploy the virtual then you can skip these steps)

### Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ	Free Trial
Resource group * ⓘ	app-grp
	<a href="#">Create new</a>

### Instance details

Virtual machine name * ⓘ	dockervm
Region * ⓘ	(Europe) North Europe
Availability options ⓘ	No infrastructure redundancy required
Security type ⓘ	Standard
Image * ⓘ	Ubuntu Server 22.04 LTS - x64 Gen2 (free services eligible)

[See all images](#) | [Configure VM generation](#)



This image is compatible with additional security features. [Click here to swap to the Trusted launch security type.](#)

Size \* ⓘ Standard\_B1s - 1 vcpu, 1 GiB memory (₹647.83/month) (free services eligible) ✓  
[See all sizes](#)

Enable Hibernation (preview) ⓘ

☐

**i** To enable Hibernation, you must register your subscription. [Learn more](#) ⓘ

### Administrator account

Authentication type ⓘ

☐ SSH public key  
☒ Password

Username \* ⓘ

demour ✓

Password \* ⓘ

..... ✓

Confirm password \* ⓘ

..... ✓

### Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports \* ⓘ

☐ None  
☒ Allow selected ports

Select inbound ports \*

SSH (22) ✓

**i** All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

[Review + create](#)

[< Previous](#)

[Next : Disks >](#)

4. After these steps simply move to review page and create your virtual machine.
5. Then you need to open Putty and run your virtual machine using user name and password.
6. For that after the deployment, go to resources and copy your public IP address.

### ✓ Your deployment is complete



Deployment name: CreateVm-canonical.0001-com-ubuntu-server-j...  
Subscription: [Free Trial](#)  
Resource group: [app-grp](#)

Start time: 12/30/2023, 11:49:52 PM

Correlation ID: 4d0a665a-9002-414a-97c6-aa1806d6cb7a

✓ **Deployment details**

^ **Next steps**

[Setup auto-shutdown](#) Recommended

[Monitor VM health, performance and network dependencies](#) Recommended

[Run a script inside the virtual machine](#) Recommended

[Go to resource](#)

[Create another VM](#)



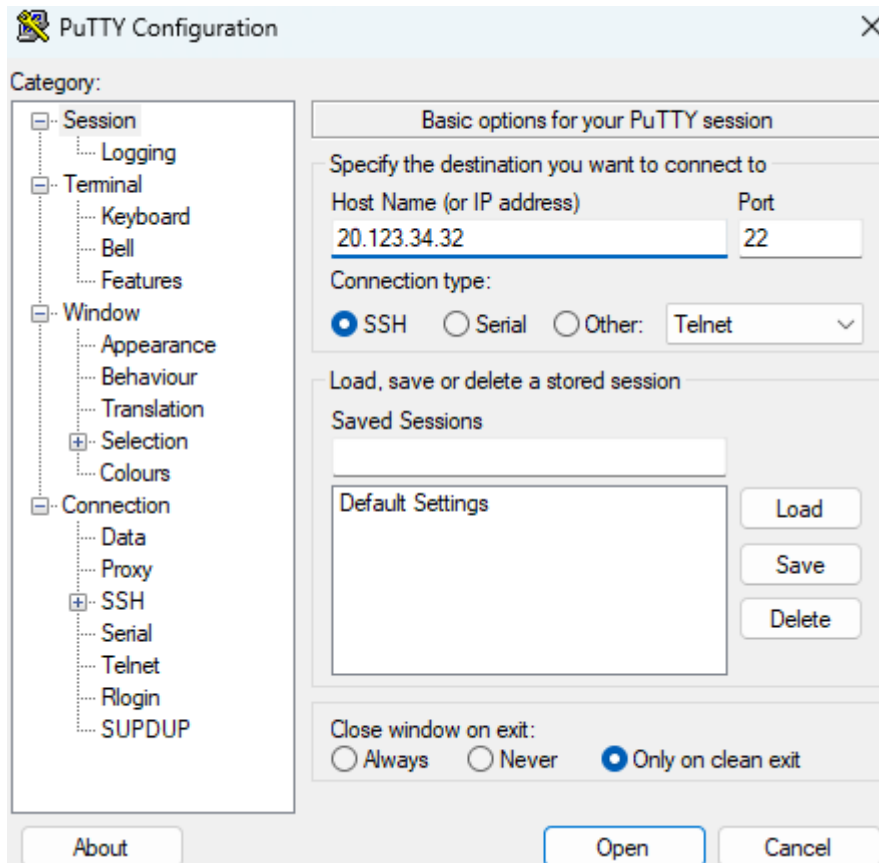
Connecting using

Public IP address | 20.123.34.32



## STEP 2: INSTALL DOCKER ON YOUR VIRTUAL MACHINE

1. Now open putty and paste your IP there.



2. Click on open then give your user's name and password.
3. Once you are in your virtual machine, now you need to install docker in it.
4. For that you are going to use a link and copy all of its command step by step in the virtual machine.
5. The link to install docker is <https://docs.docker.com/engine/install/ubuntu/>
6. You can either use this site for the latest commands but I am putting every command step by step below or you can use these.

**sudo apt-get update**

**sudo apt-get install ca-certificates curl gnupg**

```
sudo install -m 0755 -d /etc/apt/keyrings
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg  
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

```
echo \  
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \  
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt-get update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin  
docker-compose-plugin
```

```
sudo docker --version
```

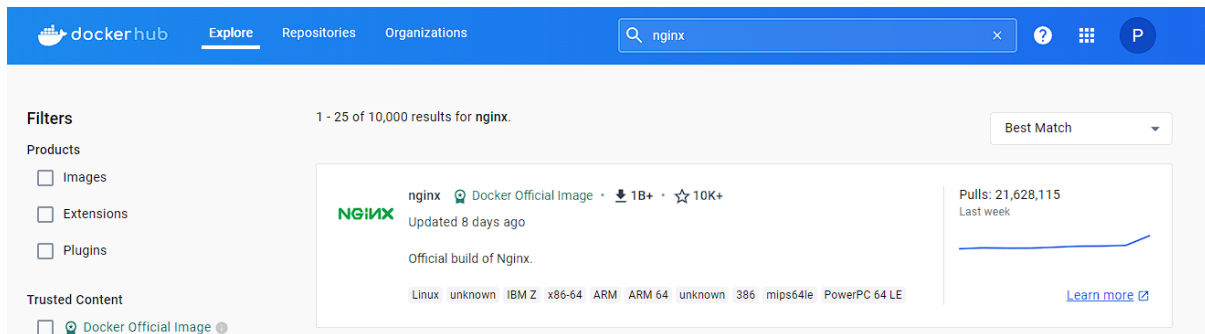
```
demousr@dockervm: ~  
login as: demousr  
demousr@20.123.34.32's password:  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-1018-azure x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Sat Dec 30 18:23:02 UTC 2023  
  
System load:  0.20556640625      Processes:            105  
Usage of /:   5.1% of 28.89GB    Users logged in:     0  
Memory usage: 31%               IPv4 address for eth0: 10.0.0.4  
Swap usage:   0%  
  
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 list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
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.  
  
demousr@dockervm:~$
```

7. Hence the docker is installed successfully.

```
demousr@dockervm:~$ sudo docker --version  
Docker version 24.0.7, build afdd53b  
demousr@dockervm:~$
```

### STEP 3: INSTALL NGINX ON YOUR VIRTUAL MACHINE

1. To install NGINX on your virtual machine you need to go to a website named **Docker Hub**.  
<https://hub.docker.com/>
2. On this website, first you need to create your account. Then you need to search NGINX in the search bar and open the first link.



3. After opening this you need to scroll down a little there you can see multiple commands and you can play with those as well (if you want to), but for the time being you need to use this command

**sudo docker run --name mynginx -p 80:80 -d nginx**

4. I'm telling Docker, please use the nginx image that is available in Docker Hub. What is command will first do is, it will download the image from Docker Hub and then, why the run command is going to create a container out of that image. It's going to give a name to the container. So, I'm giving my own name here. I'm saying please run the process in the background. I'm doing a port mapping of 80 to 80 now. My engine X server is going to run, has a container on the Azure VM.

```
demouser@dockervm: ~
demouser@dockervm:~$ sudo docker run --name mynginx -p 80:80 -d nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
af107e978371: Pull complete
336balf05c3e: Pull complete
8c37d2ff6efa: Pull complete
51d6357098de: Pull complete
782f1ecce57d: Pull complete
5e98d351b073: Pull complete
7b73345df136: Pull complete
Digest: sha256:2bdc49f2f8ae8d8dc50ed00f2ee56d00385c6f8bc8a8b320d0a294d9e3b49026
Status: Downloaded newer image for nginx:latest
8e7ec69d05351532f385d8cacab94d6dc728bc2c474dfac8564b05c2cfe064e
demouser@dockervm:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx         latest   d453dd892d93   2 months ago   187MB
demouser@dockervm:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
8e7ec69d0535   nginx    "/docker-entrypoint..." About a minute ago Up About a minute 0.0.0.0:80->80/tcp, :::80->80/tcp   mynginx
demouser@dockervm:~$
```

5. So, now you need to copy your public IP address and paste it in a new tab.

