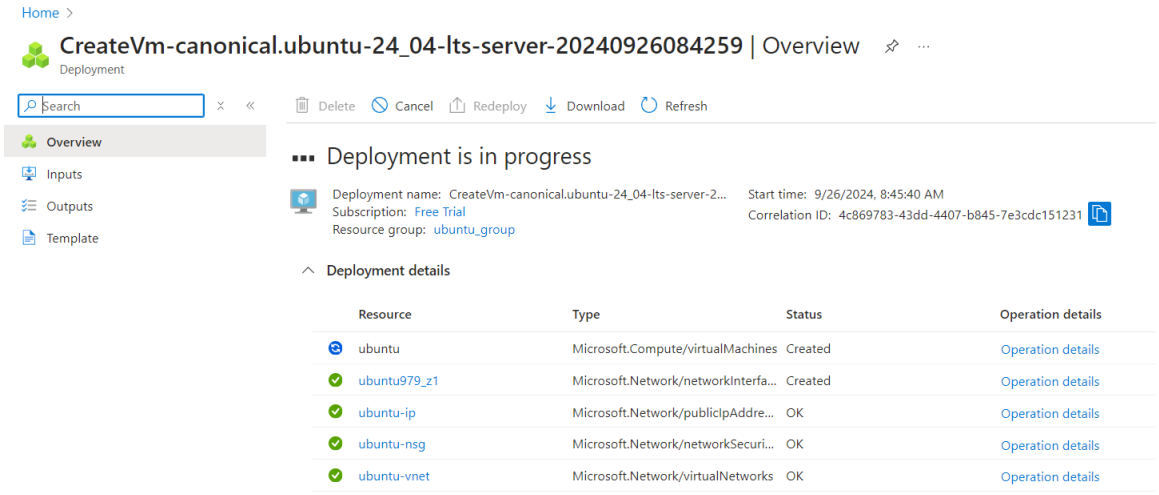
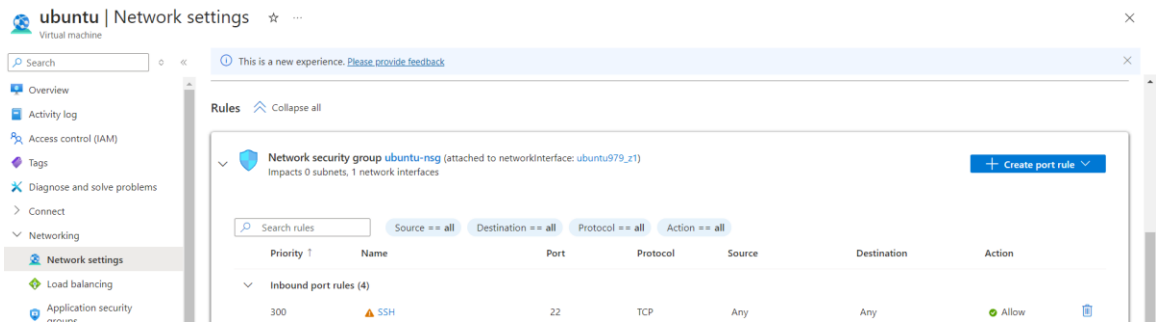
Step 1: Create a Virtual Machine in Azure

1. **Log in to the Azure Portal**:
    - Go to the Azure Portal and sign in with your account.
2. **Create a New Virtual Machine**:
    - In the Azure Portal, click on **"Create a resource"** from the left-hand menu.
3. **Configure the VM Basics**:
    - **Subscription**: Choose your subscription.
    - **Resource Group**: You can either create a new resource group or select an existing one.
    - **Virtual Machine Name**: Give your VM a name.
    - **Region**: Select **"West US"** from the dropdown menu.
    - **Availability Options**: Choose according to your needs (No infrastructure redundancy, Availability zone, etc.).
    - **Image**: Select **"Ubuntu"** from the list of available images (choose the specific version you need, e.g., Ubuntu 20.04 LTS).
    - **Size**: Choose the VM size based on your requirements.
4. **Configure Administrator Account**:
    - Choose the **Authentication type** (SSH public key or password). If using SSH, you'll need to generate an SSH key pair if you haven't already.
    - Enter the **Username** and **SSH public key** (if applicable).
5. **Configure Networking**:
    - Under the **Networking** tab, ensure a new virtual network and subnet are created or select an existing one.
    - Make sure to allow **Public IP** to connect to your VM.
6. **Open SSH Port**:
    - In the **Networking** section, add an inbound port rule to allow **SSH (port 22)**.
7. **Review + Create**:

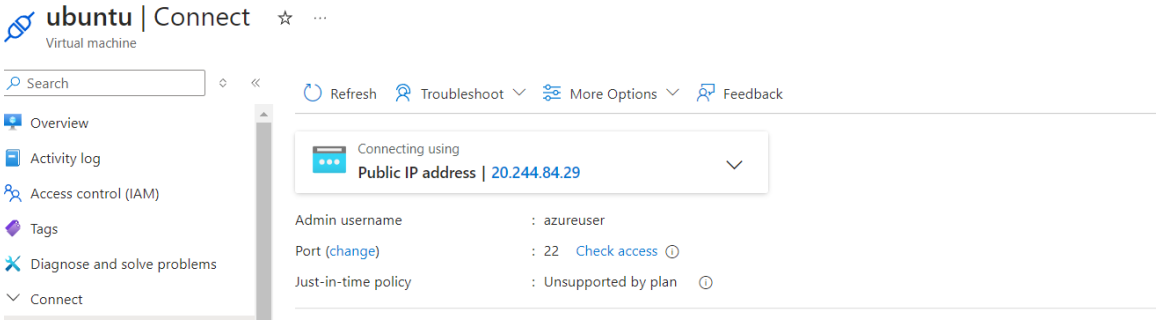
- Review your settings and click **"Create"** to provision the VM. This may take a few minutes.





Step 2: Connect to the Linux VM using Terminal

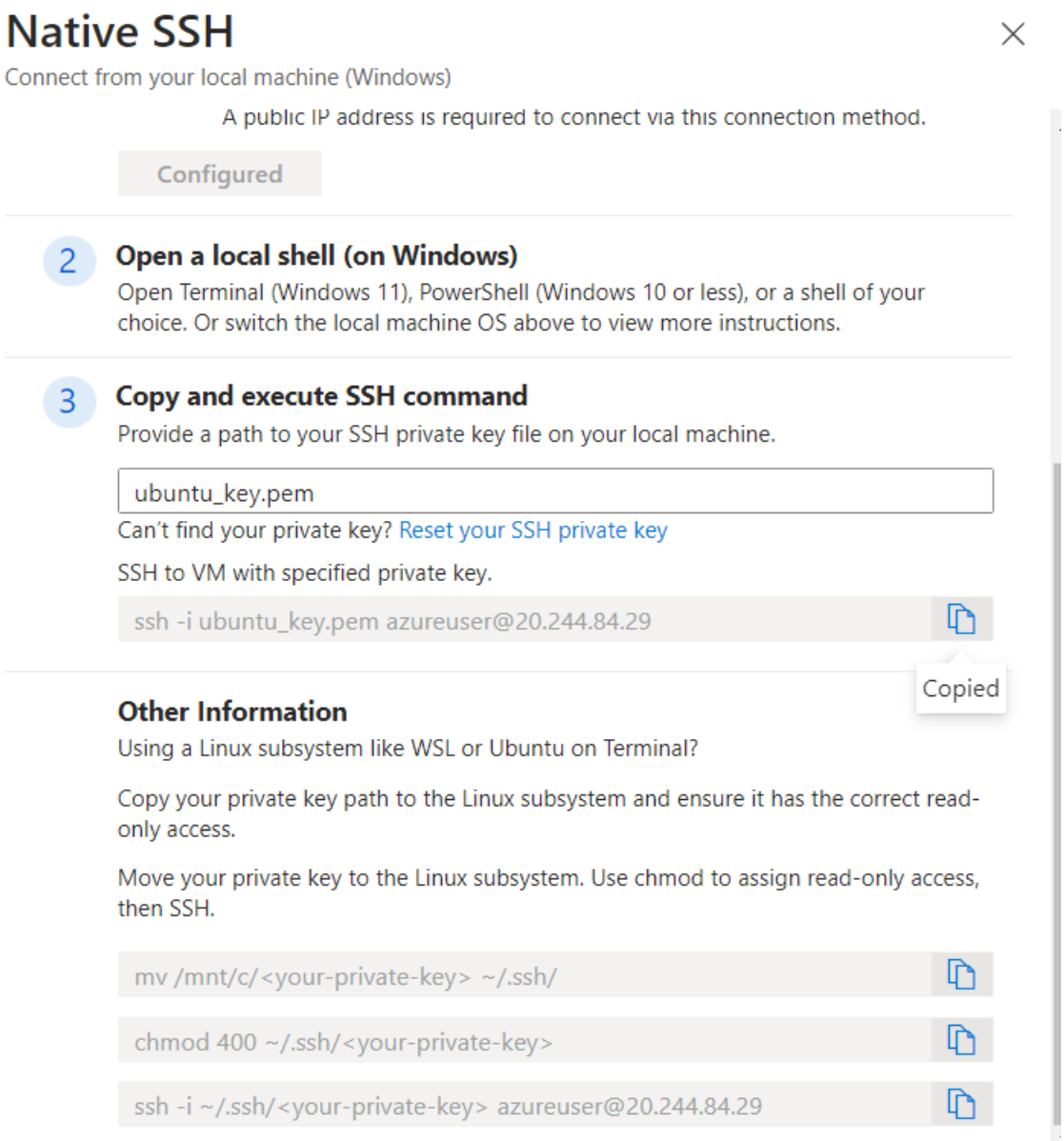
1. **Get the Public IP Address**:
   - Once the VM is created, go to the **"Overview"** page of your VM in the Azure Portal.
   - Note the **Public IP address** of the VM.



2. **Open Terminal**:

- On your local machine, open a terminal (Linux, macOS, or Windows with WSL).

3. **Connect to the VM**:
   - Use the following command to connect via SSH:



4. **Accept the SSH Key**:
   - The first time you connect, you'll be asked to confirm the authenticity of the host. Type **"yes"** and hit **Enter**.

```
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Mohd Shahid\Downloads>ssh -i ubuntu_key.pem azureuser@20.244.84.29
The authenticity of host '20.244.84.29 (20.244.84.29)' can't be established.
ED25519 key fingerprint is SHA256:4fA3FO9zmhQZqMWLJteU0fMl/9dm04iFCuVxz7LxbzE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

- If you used a password for authentication, enter it when prompted.

```
System information as of Thu Sep 26 03:20:44 UTC 2024

  System load:  0.09              Processes:             112
  Usage of /:   5.0% of 28.02GB   Users logged in:       0
  Memory usage: 28%               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.

azureuser@ubuntu:~$
```

Step 3: Install Docker

Once connected to your VM, follow these steps to install Docker:

Update the package index

sudo apt update

Install Docker

sudo apt install -y docker.io

Start Docker service

sudo systemctl start docker

Enable Docker to start on boot

sudo systemctl enable docker

Step 4: Pull the hshar/webapp Repository

Now that Docker is installed, you can pull the hshar/webapp repository:

docker pull hshar/webapp

Step 5: Create a New File in the Docker Image

To create a new file in the hshar/webapp image, you'll need to run a container from that image and then create the file within that container.

1. **Run a Container from the Image:**

sudo docker run -it --name my_webapp_container hshar/webapp /bin/bash

This command will start an interactive terminal session inside the container.

2. **Create a New File:** Inside the container, create a new file:

touch new_file.txt

echo "This is a new file created in the hshar/webapp container." > new_file.txt

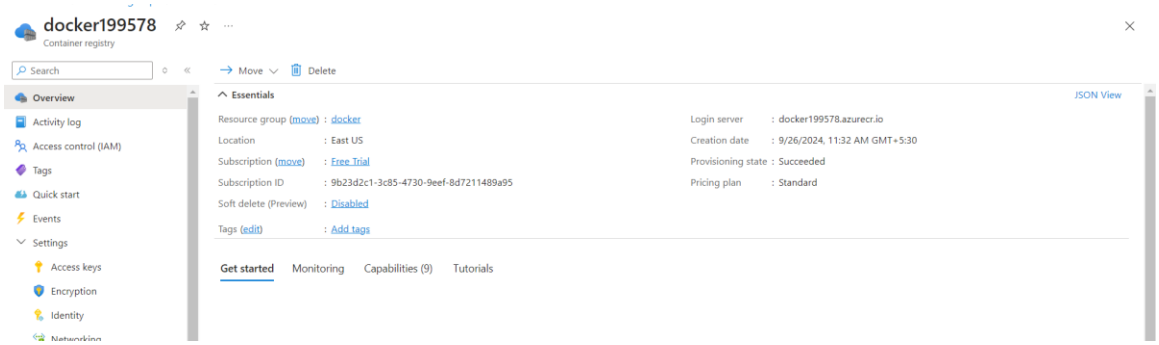
3. **Exit the Container:** To exit the container while keeping it running:

exit

Step 6: Create an Azure Container Registry and Connect It to Docker Running in VM

1. **Create a New Container Registry:**
    - In the Azure Portal, click on "Create a resource."
    - Search for "Container Registry" and select it.
    - Click "Create."

2. **Configure the Container Registry:**
    - Fill out the necessary information (Subscription, Resource Group, Registry Name, Location, SKU).
    - Click "Review + Create" and then "Create."



3. **Log in to the Azure Container Registry from the VM:**
    - Run the following command in your VM

      az acr login --name docker199578

```
root@Docker:/home/azureuser# az acr login --name docker199578
Unable to get AAD authorization tokens with message: Please run 'az login' to setup account.
Unable to get admin user credentials with message: Please run 'az login' to setup account.
Username: docker199578
Password:
Login Succeeded
```

Step 7: Upload the Image to Azure Container Registry

1. **Tag the Docker Image:**

   sudo docker commit my_webapp_container docker199578.azurecr.io/my_webapp_image:latest

```
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
root@Docker:/home/azureuser# docker commit my_webapp_container docker199578.azurecr.io/my_webapp_image:latest
sha256:727e36c28ad58c4af6c1d3bf7580d189e0248e5e5510fad22f3c2c48188af22d
root@Docker:/home/azureuser#
```

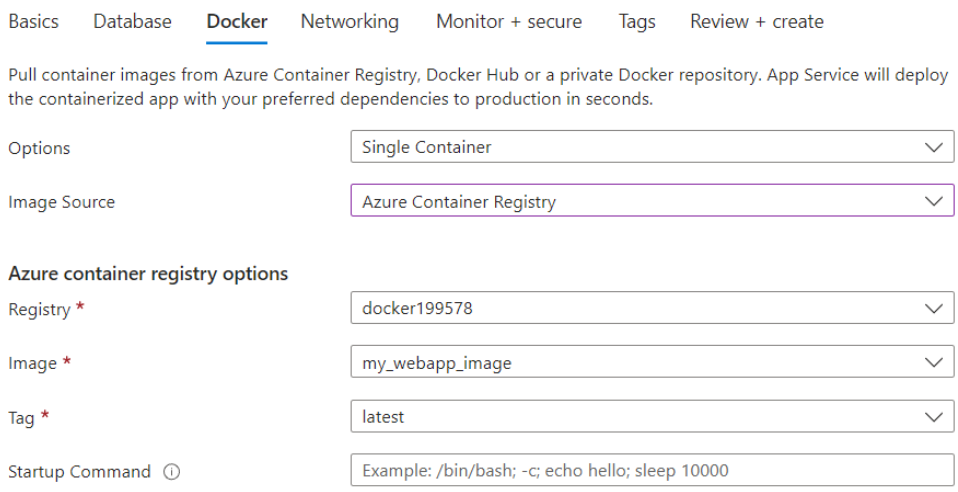
2. **Push the Docker Image to the Registry:**

   sudo docker push docker199578.azurecr.io/my_webapp_image:latest

```
root@Docker:/home/azureuser# docker push docker199578.azurecr.io/my_webapp_image:latest
The push refers to repository [docker199578.azurecr.io/my_webapp_image]
431d03f382ae: Pushed
```

Step 8: Create an App Service to Deploy the Same Image

1. **Create a New App Service:**
    - In the Azure Portal, click on "Create a resource."
    - Search for "Web App" and select it.
    - Click "Create."
2. **Configure the App Service:**
    - **Subscription:** Choose your subscription.
    - **Resource Group:** Use the same resource group as your Container Registry.
    - **Name:** dockerwebapp199578.
    - **Publish:** Choose "Docker Container."
    - **Operating System:** Select Linux.
    - **Region:** Choose the same region as your resources.
    - **App Service Plan:** Create a new plan or use an existing one.
3. **Configure Docker Settings:**
    - **Image Source:** Select "Azure Container Registry."
    - **Registry:** Choose your created container registry.
    - **Image:** Enter my_webapp_image.
    - **Tag:** Enter latest.



4. **Review + Create:**
    - Review your settings and click "Create."