

DEPLOYING A STATIC “Hello World” WEB APP ON AZURE

This use case walks through the deployment of a simple "Hello World" web app using HTML and CSS, containerized with Docker, and deployed via Azure Kubernetes Service (AKS), with CI/CD pipelines managed through Azure DevOps.

First, log in to the Azure portal and verify that the correct subscription is selected:

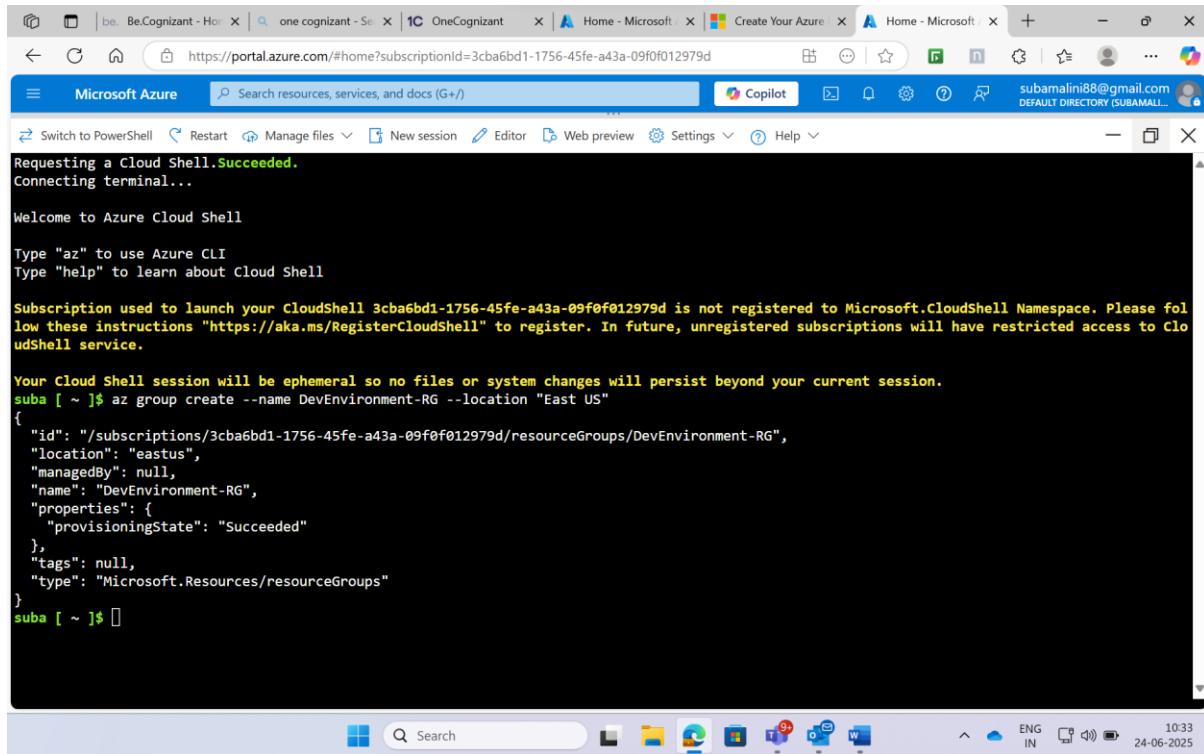
The screenshot shows the Microsoft Azure portal homepage. At the top, there's a navigation bar with links like 'Create Your Azure', 'Home - Microsoft', and 'Copilot'. Below the bar, the main content area is divided into sections: 'Azure services' (with links to 'Create a resource', 'Subscriptions', 'Quickstart Center', 'Azure AI Foundry', 'Kubernetes services', 'Virtual machines', 'App Services', 'Storage accounts', 'SQL databases', and 'More services'), 'Resources' (with tabs for 'Recent' and 'Favorite', and a table showing no recent resources), and 'Navigate' (with links to 'Subscriptions', 'Resource groups', 'All resources', and 'Dashboard'). At the bottom, there's a toolbar with icons for various Azure services and a status bar showing 'ENG IN' and the date '24-06-2025'.

Then, switch to the Bash shell in the Azure Cloud Shell environment:

The screenshot shows the Microsoft Azure portal homepage again, but with a 'Welcome to Azure Cloud Shell' dialog box overlaid on the bottom half of the screen. The dialog box contains the text: 'Welcome to Azure Cloud Shell', 'Select Bash or PowerShell. You can change shells any time via the environment selector in the Cloud Shell toolbar.', and 'The most recently used environment will be the default for your next session.' It has two buttons at the bottom: 'Bash' and 'PowerShell'. The rest of the page below the dialog box is a dark gray placeholder for the Cloud Shell interface.

Create an Azure Resource Group

- **Name:** DevEnvironment-RG
- **Region:** East US
- **Purpose:** To group all resources related to this project.
- **Bash Command:** az group create --name DevEnvironment-RG --location "East US"



The screenshot shows a Microsoft Edge browser window with multiple tabs open. The active tab is titled 'Microsoft Azure' and contains a terminal session. The terminal output is as follows:

```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...
Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

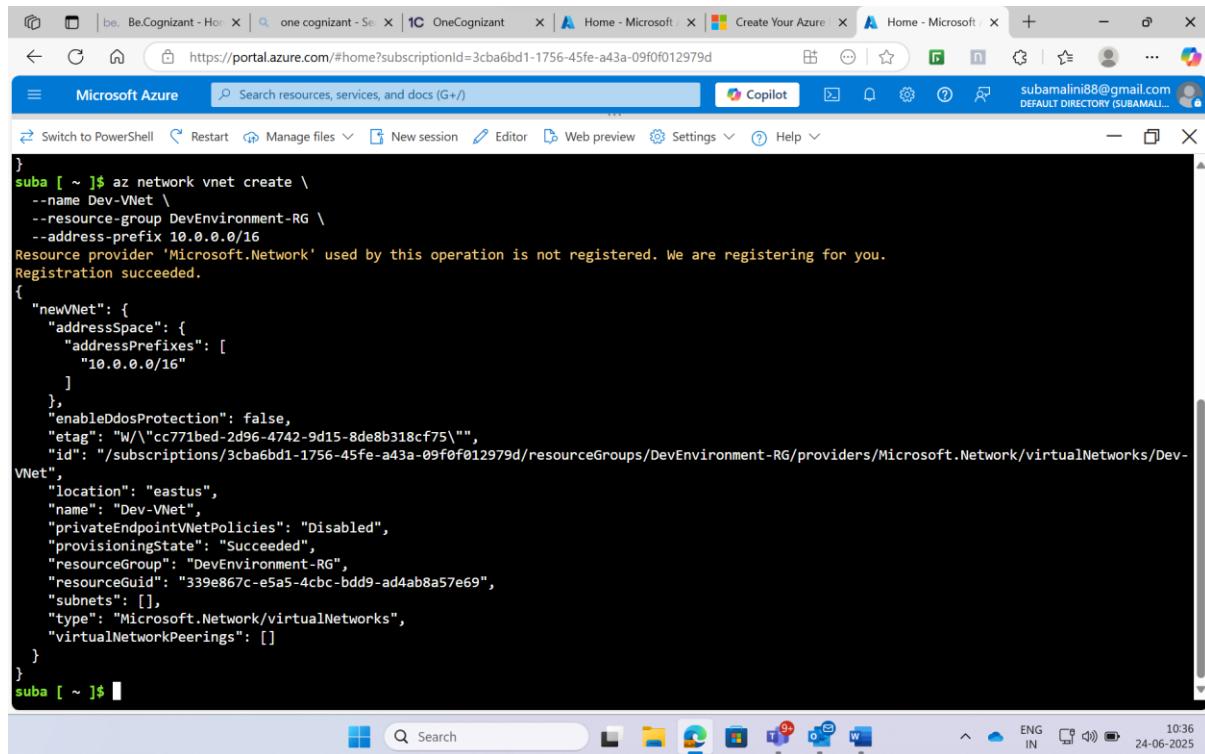
Subscription used to launch your CloudShell 3cba6bd1-1756-45fe-a43a-09f0f012979d is not registered to Microsoft.CloudShell Namespace. Please follow these instructions "https://aka.ms/RegisterCloudShell" to register. In future, unregistered subscriptions will have restricted access to CloudShell service.

Your Cloud Shell session will be ephemeral so no files or system changes will persist beyond your current session.
suba [ ~ ]$ az group create --name DevEnvironment-RG --location "East US"
{
  "id": "/subscriptions/3cba6bd1-1756-45fe-a43a-09f0f012979d/resourceGroups/DevEnvironment-RG",
  "location": "eastus",
  "managedBy": null,
  "name": "DevEnvironment-RG",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
suba [ ~ ]$ 
```

Virtual Network (VNet)

- **Name:** Dev-VNet
- **Address Space:** 10.0.0.0/16
- **Purpose:** Enable communication between resources in the Azure environment.
- **Bash Command:**

```
az network vnet create \
    --name Dev-VNet \
    --resource-group DevEnvironment-RG \
    --address-prefix 10.0.0.0/16
```

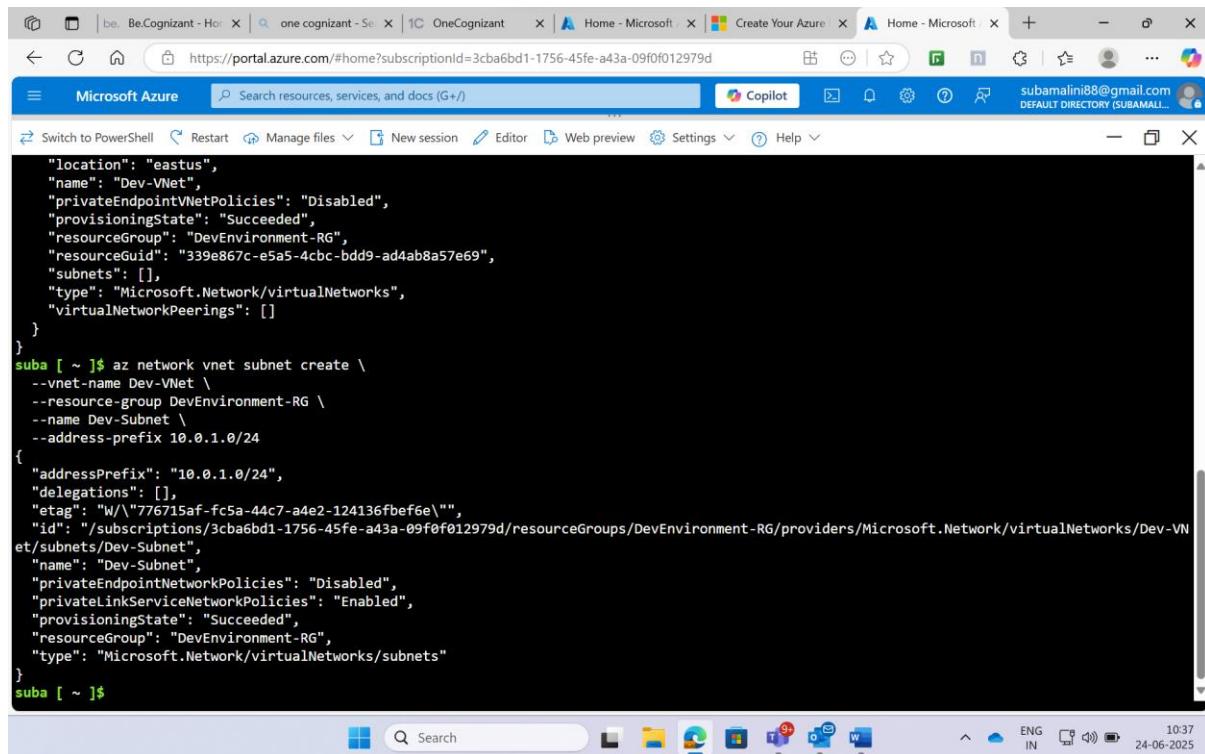


```
suba [ ~ ]$ az network vnet create \
    --name Dev-VNet \
    --resource-group DevEnvironment-RG \
    --address-prefix 10.0.0.0/16
Resource provider 'Microsoft.Network' used by this operation is not registered. We are registering for you.
Registration succeeded.
{
  "newVNet": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/16"
      ]
    },
    "enableDdosProtection": false,
    "etag": "W/\\"cc771bed-2d96-4742-9d15-8de8b318cf75\\"",
    "id": "/subscriptions/3cba6bd1-1756-45fe-a43a-09f0f012979d/resourceGroups/DevEnvironment-RG/providers/Microsoft.Network/virtualNetworks/DevVNet",
    "location": "eastus",
    "name": "Dev-VNet",
    "privateEndpointVNetPolicies": "Disabled",
    "provisioningState": "Succeeded",
    "resourceGroup": "DevEnvironment-RG",
    "resourceGuid": "339e867c-e5a5-4cbc-bdd9-ad4ab8a57e69",
    "subnets": [],
    "type": "Microsoft.Network/virtualNetworks",
    "virtualNetworkPeerings": []
  }
}
suba [ ~ ]$
```

Subnet:

- **Name:** Dev-Subnet
- **Address Range:** 10.0.1.0/24
- **Purpose:** Assign IP addresses for resources inside the VNet.
- **Bash Command:**

```
az network vnet subnet create \
    --vnet-name Dev-VNet \
    --resource-group DevEnvironment-RG \
    --name Dev-Subnet \
    --address-prefix 10.0.1.0/24
```



```
suba [ ~ ]$ az network vnet subnet create \
    --vnet-name Dev-VNet \
    --resource-group DevEnvironment-RG \
    --name Dev-Subnet \
    --address-prefix 10.0.1.0/24
{
  "addressPrefix": "10.0.1.0/24",
  "delegations": [],
  "etag": "W/\"776715af-fc5a-44c7-a4e2-124136fbef6e\"",
  "id": "/subscriptions/3cba6bd1-1756-45fe-a43a-09f0f012979d/resourceGroups/DevEnvironment-RG/providers/Microsoft.Network/virtualNetworks/Dev-VNet/subnets/Dev-Subnet",
  "name": "Dev-Subnet",
  "privateEndpointNetworkPolicies": "Disabled",
  "privateLinkServiceNetworkPolicies": "Enabled",
  "provisioningState": "Succeeded",
  "resourceGroup": "DevEnvironment-RG",
  "type": "Microsoft.Network/virtualNetworks/subnets"
}
suba [ ~ ]$
```

Storage Account

- **Name:** devstorage<uniqueid> (e.g., devstorage12345)
- **Location:** East US
- **SKU:** Standard_LRS
- **Purpose:** Store logs and application-related data.

The screenshot shows the 'Create a storage account' wizard in the Microsoft Azure portal. The 'Basics' tab is selected. In the 'Project details' section, 'Subscription' is set to 'Azure subscription 1' and 'Resource group' is set to 'DevEnvironment-RG'. In the 'Instance details' section, 'Storage account name' is 'devstorage4107' and 'Region' is '(US) East US'. At the bottom, there are 'Previous', 'Next', and 'Review + create' buttons, along with a 'Give feedback' link.

Unique Storage account name: devstorage4107

The screenshot shows the 'Overview' page for the 'devstorage4107' storage account. The left sidebar includes links for Activity log, Tags, Diagnose and solve problems, Access Control (IAM), Data migration, Events, Storage browser, Storage Mover, Partner solutions, Resource visualizer, Data storage, Security + networking, Data management, and Settings. The main pane displays 'Essentials' information: Resource group (move) to 'DevEnvironment-RG', Location 'eastus', Primary/Secondary Location 'Primary: East US, Secondary: West US', Subscription (move) to 'Azure subscription 1', and various account properties like Performance 'Standard', Replication 'Read-access geo-redundant storage (RA-GRS)', Account kind 'StorageV2 (general purpose v2)', Provisioning state 'Succeeded', and Created '24/6/2025, 10:54:26 am'. The 'Properties' tab is selected. At the bottom, there are 'Blob service' and 'Security' tabs, along with a note to 'Add or remove favorites by pressing Ctrl+Shift+F'.

Virtual Machine

- **Name:** Dev-VM
- **Operating System:** Ubuntu 24.04
- **Size:** Standard_B1s (1 vCPU, 1 GB RAM, Free Tier eligible)
- **Admin Username:** azure-dev-user
- **Tags:** Project=DevEnvironment, Owner=azure-dev-user
- **Purpose:** Development environment to host tools and test applications.
- **Bash Command:**

```
az vm create \
--resource-group DevEnvironment-RG \
--name Dev-VM \
--image Ubuntu24.04 \
--size Standard_B1s \
--admin-username azure-dev-user \
--admin-password Suka@41963113 \
--vnet-name Dev-VNet \
--subnet Dev-Subnet \
--tags Project=DevEnvironment Owner=azure-dev-user
```

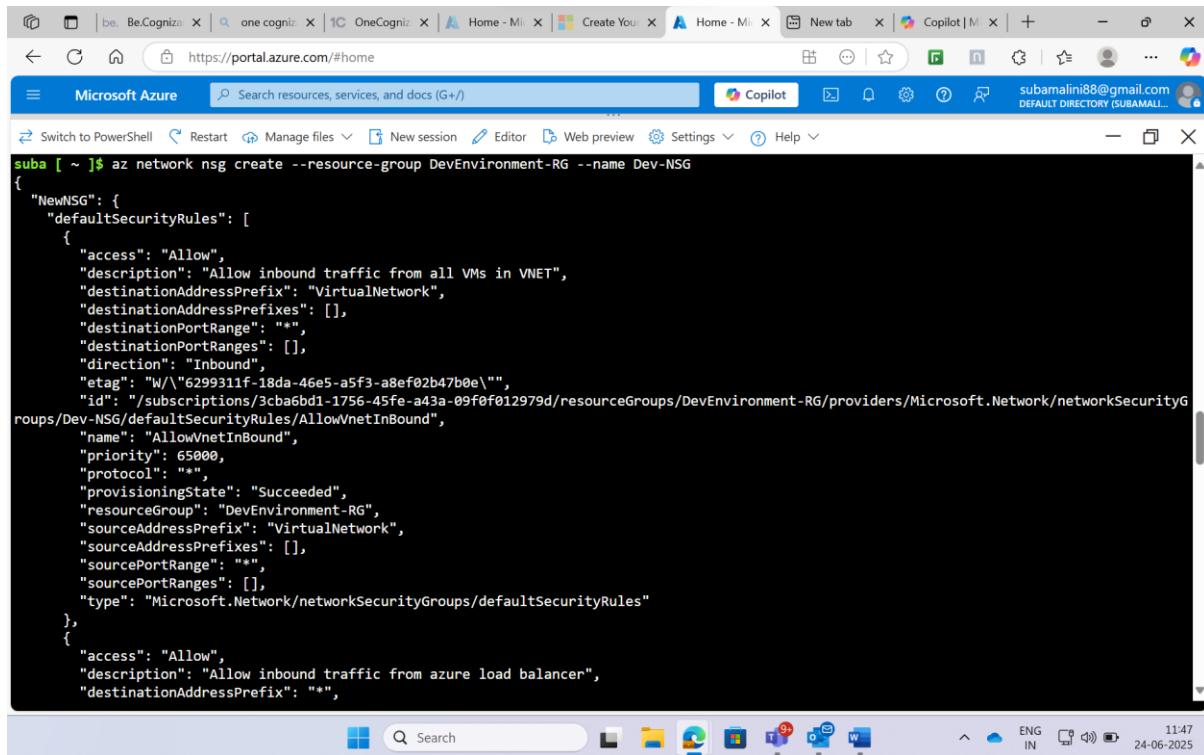
The screenshot shows the Microsoft Azure portal interface for a virtual machine named 'Dev-VM'. The left sidebar contains navigation links like Home, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Connect, Networking (Network settings, Load balancing, Application security groups, Network manager), Settings, Availability + scale, and Security. The main content area displays the VM's details under the 'Essentials' section. Key information includes:

- Resource group: DevEnvironment-RG
- Operating system: Linux (ubuntu 24.04)
- Status: Running
- Location: East US
- Public IP address: 172.206.207.208
- Virtual network/subnet: Dev-VNet/Dev-Subnet
- DNS name: Not configured
- Health state: -
- Time created: 6/24/2025, 6:04 AM UTC

At the bottom of the page, there are 'Tags (edit)' and 'Add tags' buttons. The browser's address bar shows the URL: https://portal.azure.com/?ocid=AIDcmmf1elj9v5_SEM_k_4af80855900e108ff7aa60379a3fcf40_k_&cid=free-... . The status bar at the bottom right shows the date and time: 30-06-2025 10:32 ENG IN.

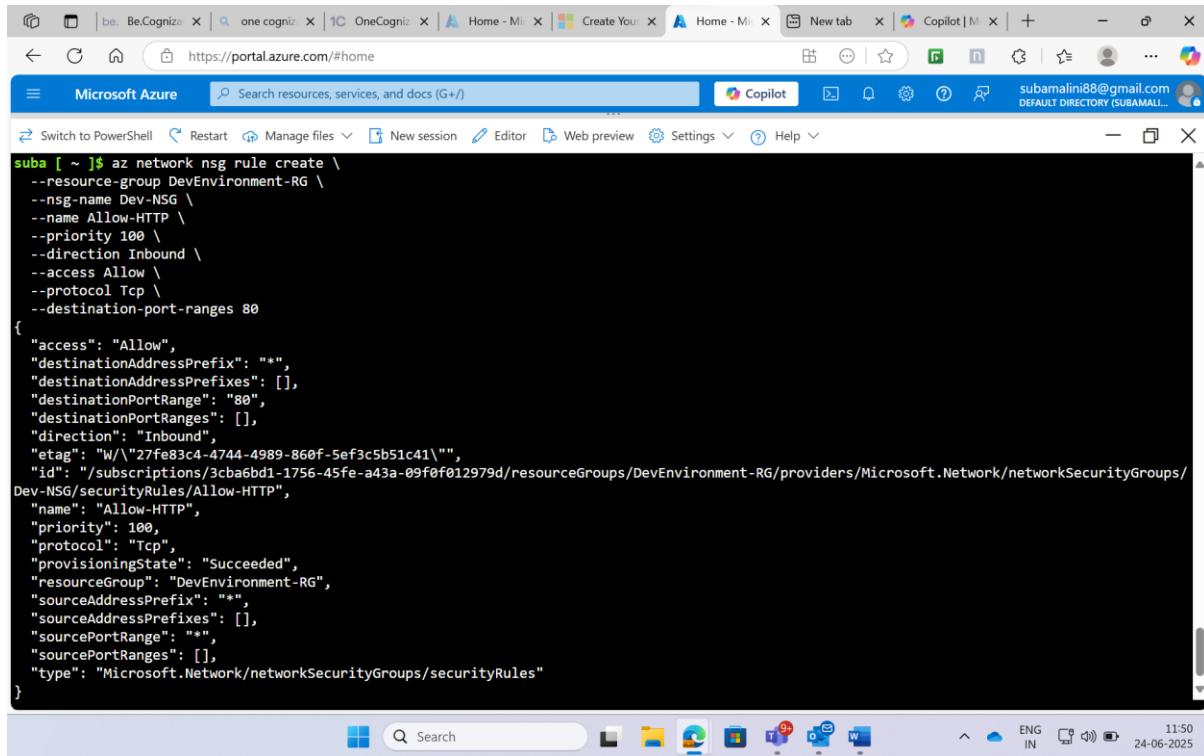
Network Security Group (NSG)

- **Name:** Dev-NSG
- **Rules:** Allow HTTP traffic on port 80 and SSH traffic on port 22.
- **Bash Command to Create NSG:** az network nsg create --resource-group DevEnvironment-RG --name Dev-NSG



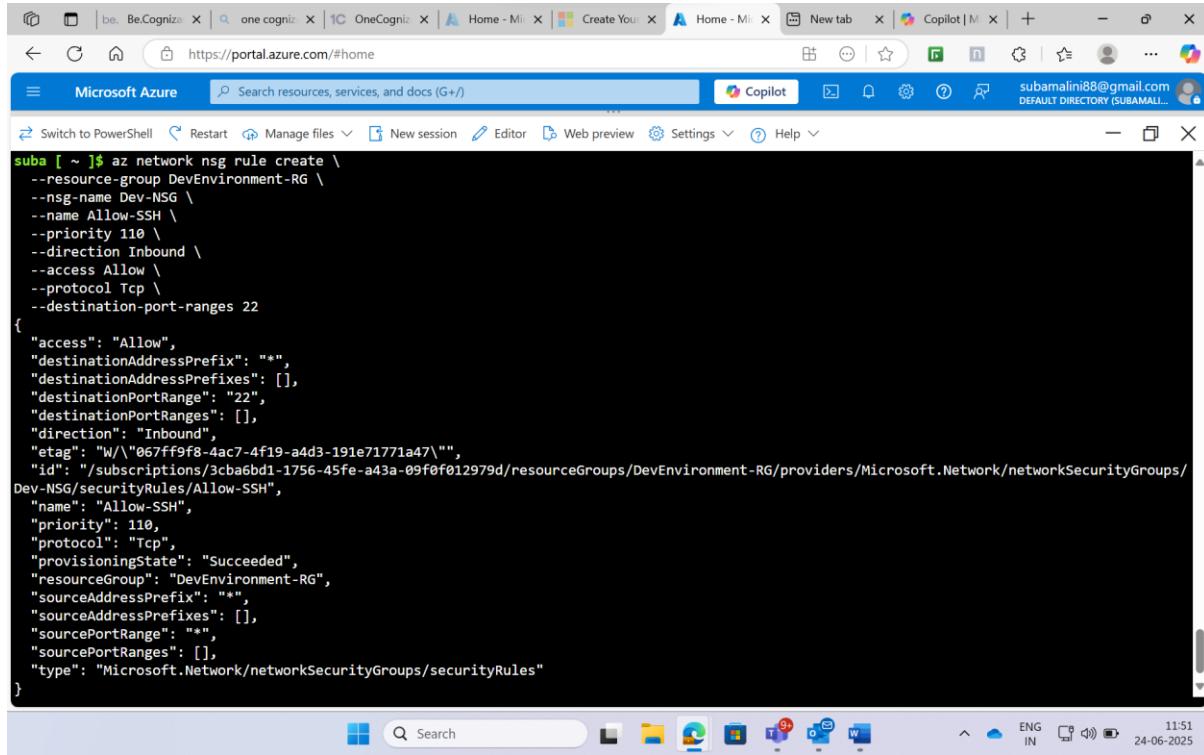
```
suba [ ~ ]$ az network nsg create --resource-group DevEnvironment-RG --name Dev-NSG
{
  "NewNSG": {
    "defaultSecurityRules": [
      {
        "access": "Allow",
        "description": "Allow inbound traffic from all VMs in VNET",
        "destinationAddressPrefix": "VirtualNetwork",
        "destinationAddressPrefixes": [],
        "destinationPortRange": "*",
        "destinationPortRanges": [],
        "direction": "Inbound",
        "etag": "W\"6299311f-18da-46e5-a5f3-a8ef02b47b0e\"",
        "id": "/subscriptions/3cba6bd1-1756-45fe-a43a-09f0f012979d/resourceGroups/DevEnvironment-RG/providers/Microsoft.Network/networkSecurityGroups/Dev-NSG/defaultSecurityRules/AllowVnetInBound",
        "name": "AllowVnetInBound",
        "priority": 65000,
        "protocol": "*",
        "provisioningState": "Succeeded",
        "resourceGroup": "DevEnvironment-RG",
        "sourceAddressPrefix": "VirtualNetwork",
        "sourceAddressPrefixes": [],
        "sourcePortRange": "*",
        "sourcePortRanges": [],
        "type": "Microsoft.Network/networkSecurityGroups/defaultSecurityRules"
      },
      {
        "access": "Allow",
        "description": "Allow inbound traffic from azure load balancer",
        "destinationAddressPrefix": "*"
      }
    ]
  }
}
```

Create an inbound security rule with a priority of 100 and allow traffic on destination port 80



```
suba [ ~ ]$ az network nsg rule create \
--resource-group DevEnvironment-RG \
--nsg-name Dev-NSG \
--name Allow-HTTP \
--priority 100 \
--direction Inbound \
--access Allow \
--protocol Tcp \
--destination-port-ranges 80
{
  "access": "Allow",
  "destinationAddressPrefix": "*",
  "destinationAddressPrefixes": [],
  "destinationPortRange": "80",
  "destinationPortRanges": [],
  "direction": "Inbound",
  "etag": "W/\"27fe83c4-4744-4989-860f-5ef3c5b51c41\"",
  "id": "/subscriptions/3cba6bd1-1756-45fe-a43a-09f0f012979d/resourceGroups/DevEnvironment-RG/providers/Microsoft.Network/networkSecurityGroups/Dev-NSG/securityRules/Allow-HTTP",
  "name": "Allow-HTTP",
  "priority": 100,
  "protocol": "Tcp",
  "provisioningState": "Succeeded",
  "resourceGroup": "DevEnvironment-RG",
  "sourceAddressPrefix": "*",
  "sourceAddressPrefixes": [],
  "sourcePortRange": "*",
  "sourcePortRanges": [],
  "type": "Microsoft.Network/networkSecurityGroups/securityRules"
}
```

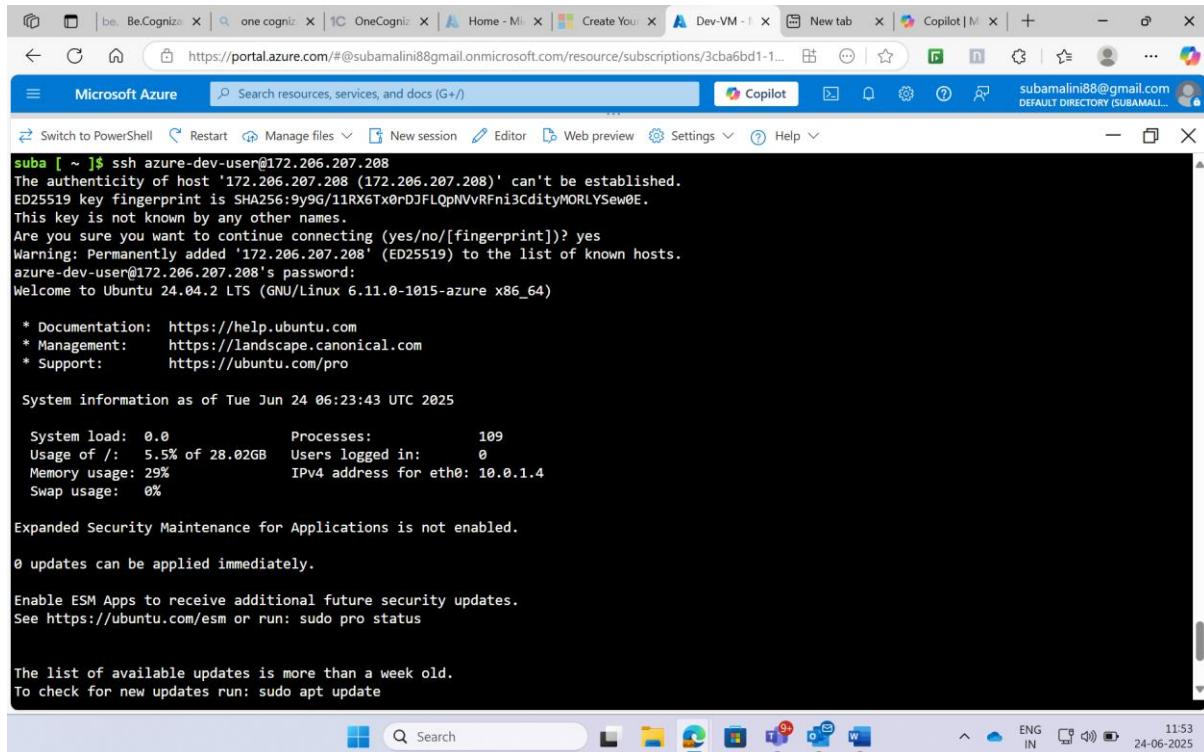
Create an inbound security rule with a priority of 110 and allow traffic on destination port 22:



```
suba [ ~ ]$ az network nsg rule create \
--resource-group DevEnvironment-RG \
--nsg-name Dev-NSG \
--name Allow-SSH \
--priority 110 \
--direction Inbound \
--access Allow \
--protocol Tcp \
--destination-port-ranges 22
{
  "access": "Allow",
  "destinationAddressPrefix": "*",
  "destinationAddressPrefixes": [],
  "destinationPortRange": "22",
  "destinationPortRanges": [],
  "direction": "Inbound",
  "etag": "W/\"067ff9f8-4ac7-4f19-a4d3-191e71771a47\"",
  "id": "/subscriptions/3cba6bd1-1756-45fe-a43a-09f0f012979d/resourceGroups/DevEnvironment-RG/providers/Microsoft.Network/networkSecurityGroups/Dev-NSG/securityRules/Allow-SSH",
  "name": "Allow-SSH",
  "priority": 110,
  "protocol": "Tcp",
  "provisioningState": "Succeeded",
  "resourceGroup": "DevEnvironment-RG",
  "sourceAddressPrefix": "*",
  "sourceAddressPrefixes": [],
  "sourcePortRange": "*",
  "sourcePortRanges": [],
  "type": "Microsoft.Network/networkSecurityGroups/securityRules"
}
```

VM password:Suka@41963113

Switching to the azure-dev-user account using the public IP address:



```
suba [ ~ ]$ ssh azure-dev-user@172.206.207.208
The authenticity of host '172.206.207.208 (172.206.207.208)' can't be established.
ED25519 key fingerprint is SHA256:9y9G/11RX6Tx0rDJFLQpNVvRFni3CdityMORLYSew0E.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.206.207.208' (ED25519) to the list of known hosts.
azure-dev-user@172.206.207.208's password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.11.0-1015-azure x86_64)

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

System information as of Tue Jun 24 06:23:43 UTC 2025

 System load: 0.0      Processes:          109
 Usage of /: 5.5% of 28.02GB  Users logged in:     0
 Memory usage: 29%           IPv4 address for eth0: 10.0.1.4
 Swap usage:  0%

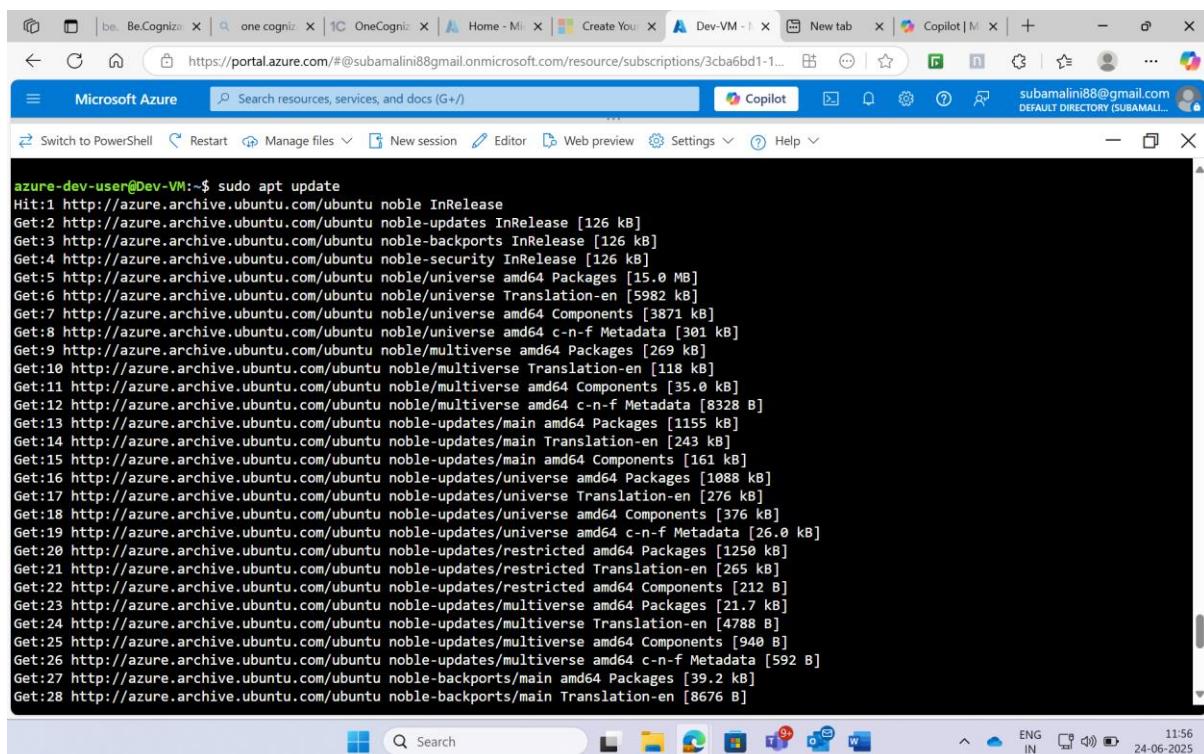
Expanded Security Maintenance for Applications is not enabled.

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
```

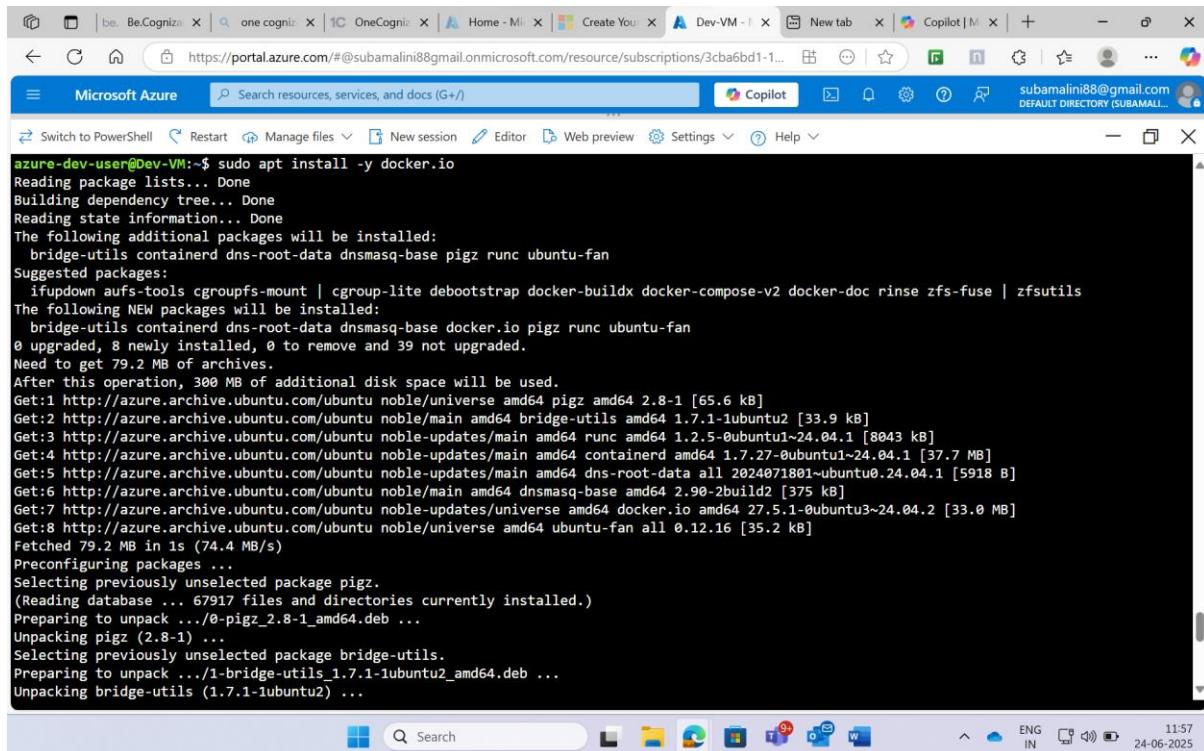
Before installing any application, it's important to update the system to ensure all packages are current:



```
azure-dev-user@Dev-VM:~$ sudo apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 Packages [15.0 MB]
Get:6 http://azure.archive.ubuntu.com/ubuntu noble/universe Translation-en [5982 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 Components [3871 kB]
Get:8 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 c-n-f Metadata [301 kB]
Get:9 http://azure.archive.ubuntu.com/ubuntu noble/multiverse amd64 Packages [269 kB]
Get:10 http://azure.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:11 http://azure.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:12 http://azure.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:13 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1155 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [243 kB]
Get:15 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [161 kB]
Get:16 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1088 kB]
Get:17 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [276 kB]
Get:18 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [376 kB]
Get:19 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [26.0 kB]
Get:20 http://azure.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Packages [1250 kB]
Get:21 http://azure.archive.ubuntu.com/ubuntu noble-updates/restricted Translation-en [265 kB]
Get:22 http://azure.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:23 http://azure.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Packages [21.7 kB]
Get:24 http://azure.archive.ubuntu.com/ubuntu noble-updates/multiverse Translation-en [4788 B]
Get:25 http://azure.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:26 http://azure.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [592 B]
Get:27 http://azure.archive.ubuntu.com/ubuntu noble-backports/main amd64 Packages [39.2 kB]
Get:28 http://azure.archive.ubuntu.com/ubuntu noble-backports/main Translation-en [8676 B]
```

Docker Installation on VM

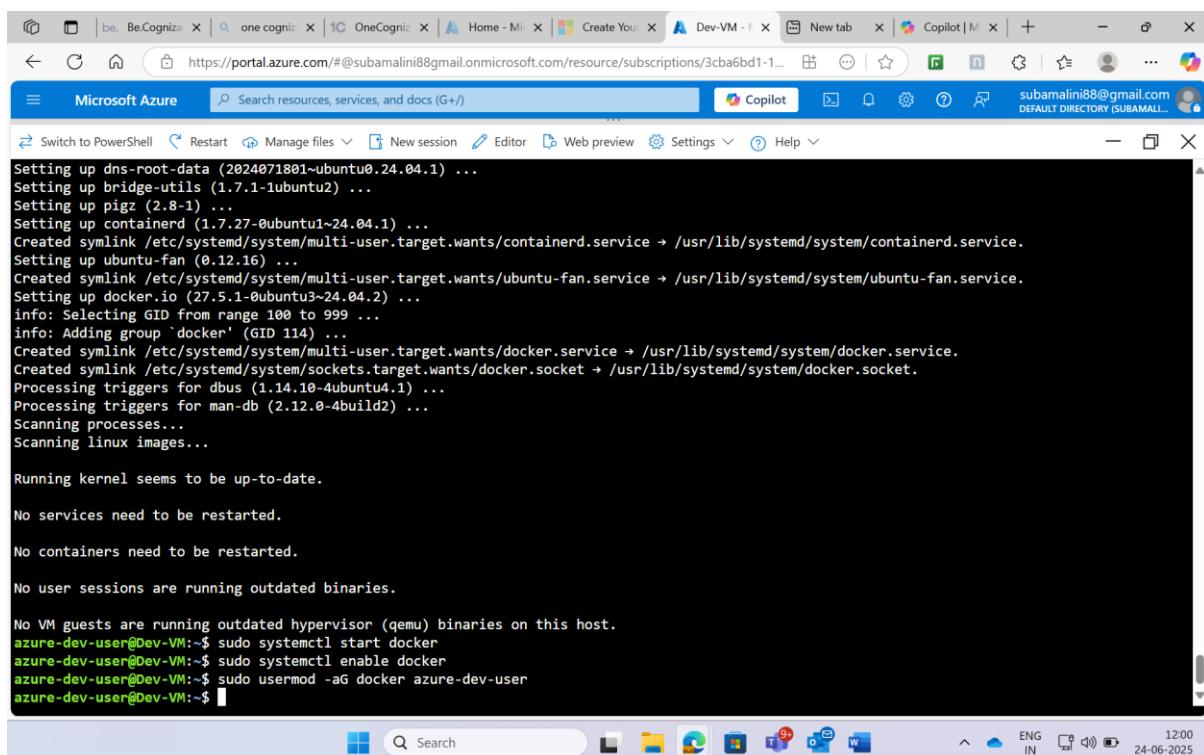
- **Version:** Latest version available via apt
- **Purpose:** Containerize the application for deployment.



```
azur...-user@Dev-VM:~$ sudo apt install -y docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 39 not upgraded.
Need to get 79.2 MB of archives.
After this operation, 300 MB of additional disk space will be used.
Get:1 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://azure.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.2.5-0ubuntu1~24.04.1 [8043 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.27-0ubuntu1~24.04.1 [37.7 MB]
Get:5 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 dns-root-data all 2024071801-ubuntu0.24.04.1 [5918 B]
Get:6 http://azure.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.90-2build2 [375 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 27.5.1-0ubuntu3-24.04.2 [33.0 MB]
Get:8 http://azure.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 79.2 MB in 1s (74.4 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 67917 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7.1-1ubuntu2_amd64.deb ...
Unpacking bridge-utils (1.7.1-1ubuntu2) ...

```

Start and enable the docker:



```
Setting up dns-root-data (2024071801-ubuntu0.24.04.1) ...
Setting up bridge-utils (1.7.1-1ubuntu2) ...
Setting up pigz (2.8-1) ...
Setting up containerd (1.7.27-0ubuntu1~24.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /usr/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service → /usr/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (27.5.1-0ubuntu3-24.04.2) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group `docker` (GID 114) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
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.
azur...-user@Dev-VM:~$ sudo systemctl start docker
azur...-user@Dev-VM:~$ sudo systemctl enable docker
azur...-user@Dev-VM:~$ sudo usermod -aG docker azure-dev-user
azur...-user@Dev-VM:~$ 
```

Git Installation on VM

- **Version:** Latest version available via apt
- **Purpose:** Version control for application files.

The screenshot shows a Microsoft Azure Dev VM terminal window. The user has run several commands to install Docker and Git. The output of the commands is as follows:

```
Setting up docker.io (27.5.1-0ubuntu3~24.04.2) ...
info: Selecting GID from range 100 to 999 ...
info: Adding group `docker` (GID 114) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
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.
azure-dev-user@Dev-VM:~$ sudo systemctl start docker
azure-dev-user@Dev-VM:~$ sudo systemctl enable docker
azure-dev-user@Dev-VM:~$ sudo usermod -aG docker azure-dev-user
azure-dev-user@Dev-VM:~$ sudo apt install -y git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.2).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 39 not upgraded.
azure-dev-user@Dev-VM:~$
```

HTML and CSS Web Application

- **Folder Name:** hello-world-app
- **File Name:** index.html

Then create the folder named “hello-world-app” then create the HTML page:

The screenshot shows a Microsoft Azure Dev VM terminal window. The user has created a new folder named "hello-world-app" and changed into it. They then created an HTML file named "index.html" and opened it with the "vi" editor. The output of the commands is as follows:

```
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

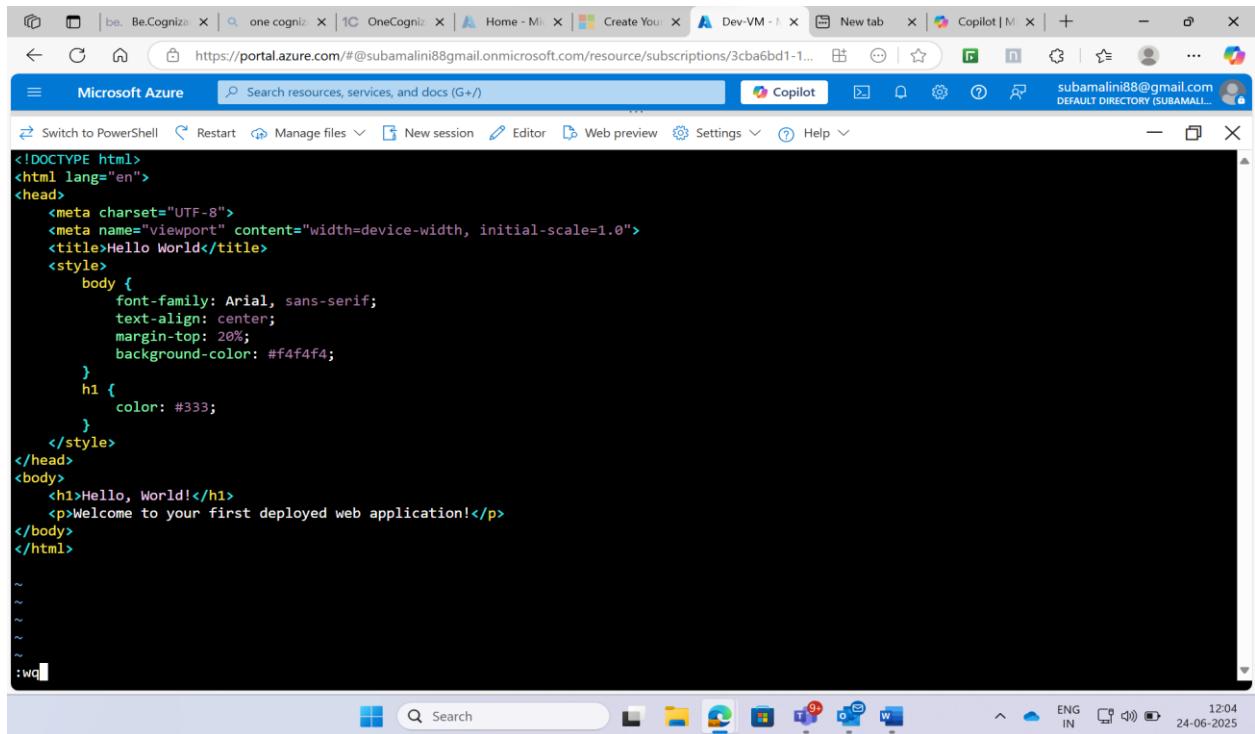
No services need to be restarted.

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.
azure-dev-user@Dev-VM:~$ sudo systemctl start docker
azure-dev-user@Dev-VM:~$ sudo systemctl enable docker
azure-dev-user@Dev-VM:~$ sudo usermod -aG docker azure-dev-user
azure-dev-user@Dev-VM:~$ sudo apt install -y git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.2).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 39 not upgraded.
azure-dev-user@Dev-VM:~$ mkdir hello-world-app
azure-dev-user@Dev-VM:~$ cd hello-world-app
azure-dev-user@Dev-VM:~/hello-world-app$ vi index.html
azure-dev-user@Dev-VM:~/hello-world-app$
```

Content of html:



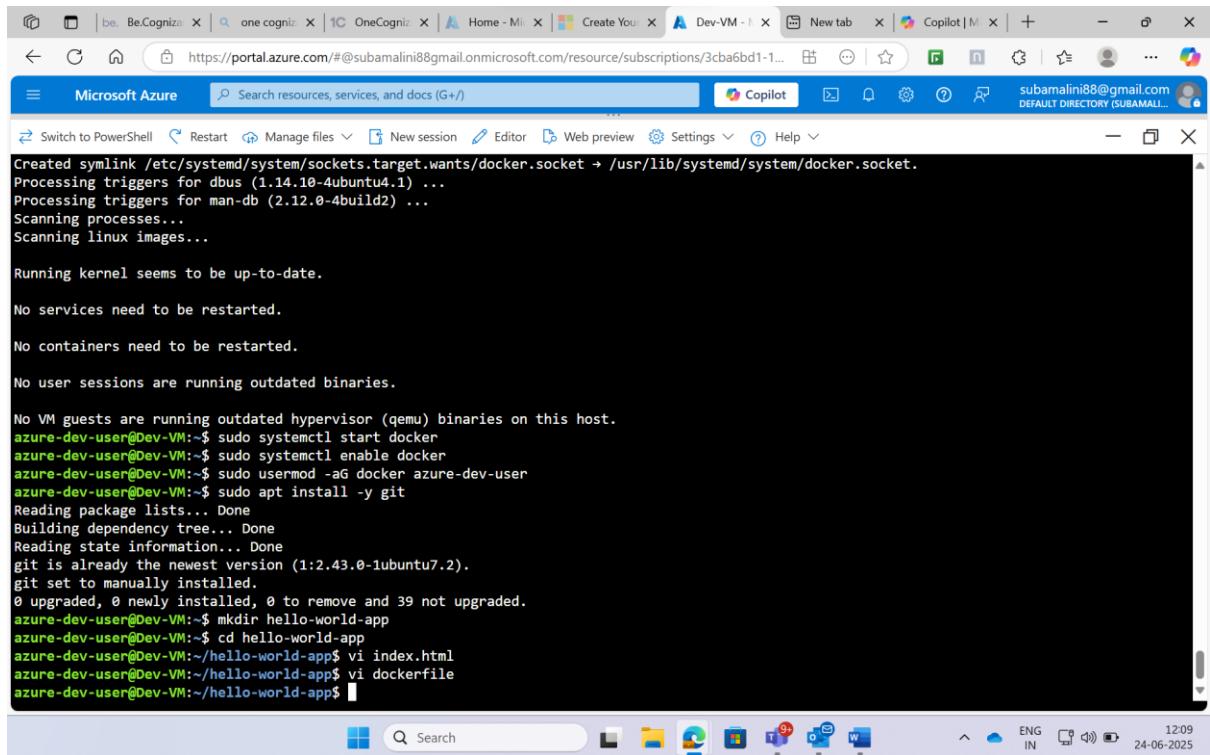
The screenshot shows the Microsoft Edge browser window with the URL <https://portal.azure.com/#@subamalin88@gmail.onmicrosoft.com/resource/subscriptions/3cba6bd1-1...>. The page title is "Microsoft Azure". The content area displays the source code of a static web application:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello World</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin-top: 20px;
            background-color: #f4f4f4;
        }
        h1 {
            color: #333;
        }
    </style>
</head>
<body>
    <h1>Hello, World!</h1>
    <p>Welcome to your first deployed web application!</p>
</body>
</html>
```

At the bottom of the code editor, there is a command line prompt with the text ":wq".

Create a Dockerfile:

- **Purpose:** Containerize the static web application.



The screenshot shows the Microsoft Edge browser window with the same URL as the previous screenshot. The content area displays the terminal output of a Docker build process:

```
Created symlink /etc/systemd/system/docker.socket → /usr/lib/systemd/system/docker.socket.
Processing triggers for dbus (1.14.10-4ubuntu4.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

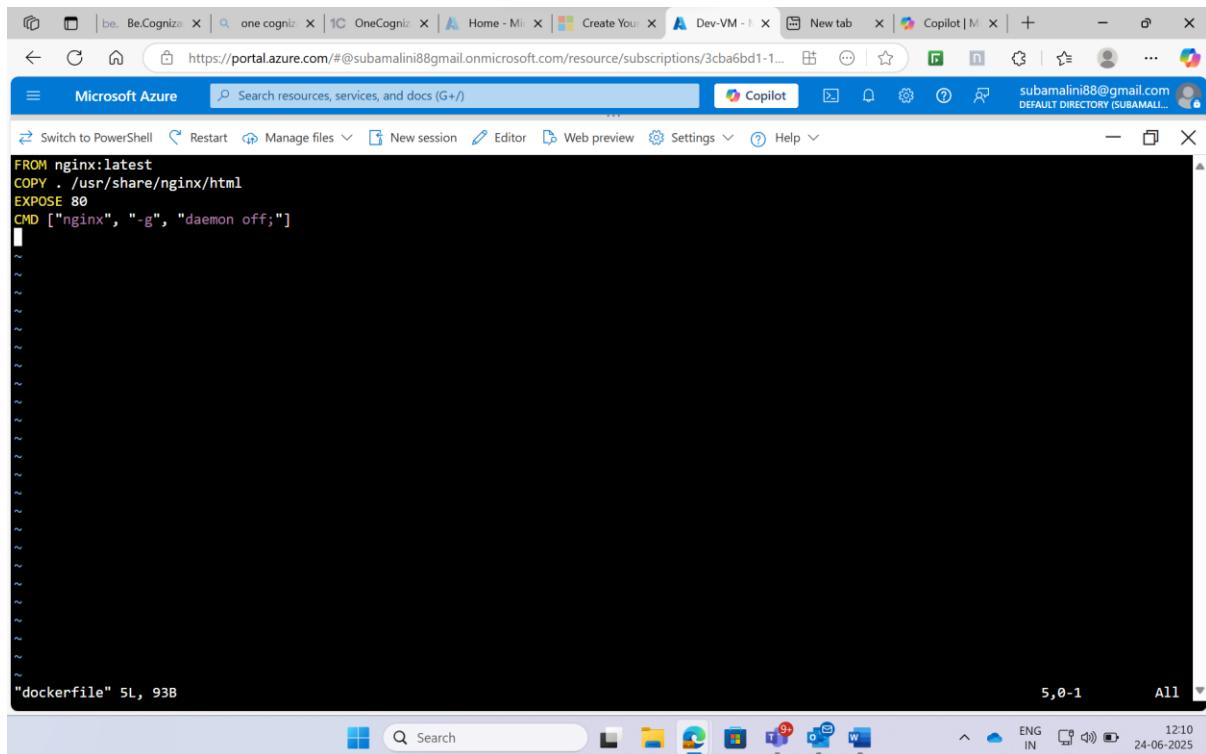
No services need to be restarted.

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.
azure-dev-user@Dev-VM:~$ sudo systemctl start docker
azure-dev-user@Dev-VM:~$ sudo systemctl enable docker
azure-dev-user@Dev-VM:~$ sudo usermod -aG docker azure-dev-user
azure-dev-user@Dev-VM:~$ sudo apt install -y git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.2).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 39 not upgraded.
azure-dev-user@Dev-VM:~$ mkdir hello-world-app
azure-dev-user@Dev-VM:~$ cd hello-world-app
azure-dev-user@Dev-VM:~/hello-world-app$ vi index.html
azure-dev-user@Dev-VM:~/hello-world-app$ vi dockerfile
azure-dev-user@Dev-VM:~/hello-world-app$
```

Content of dockerfile:

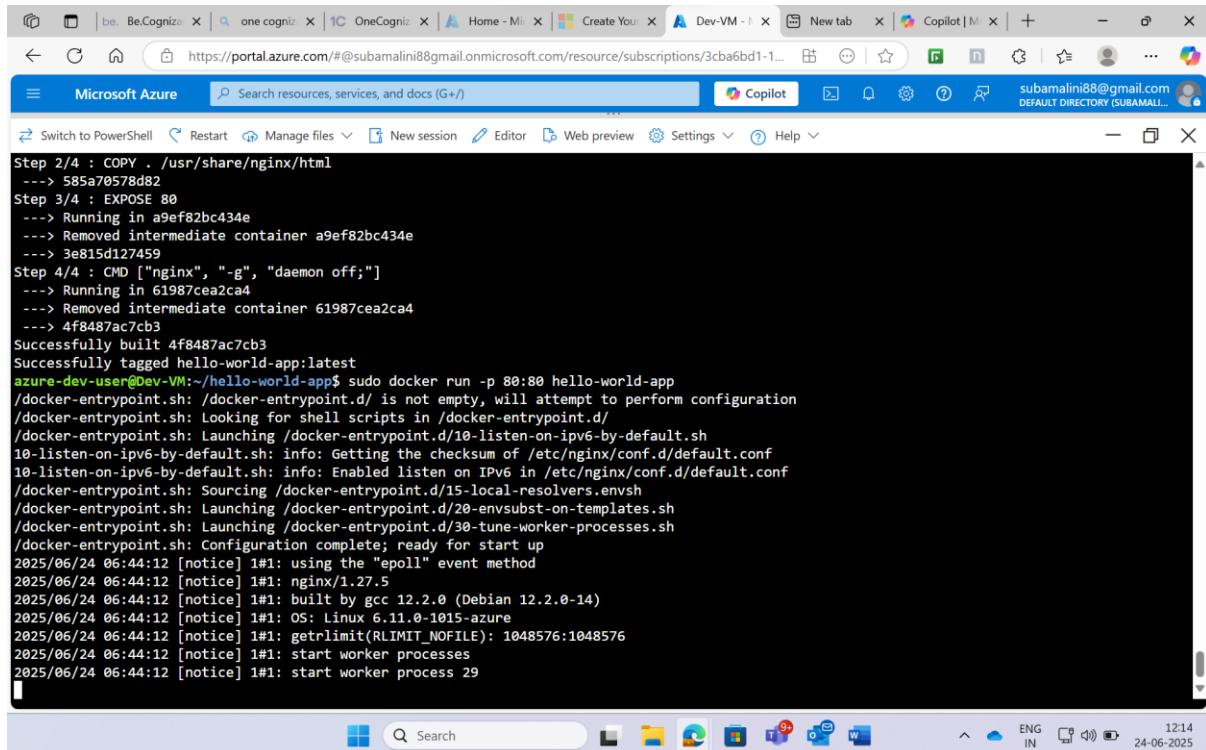


```
FROM nginx:latest
COPY . /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

"dockerfile" 5L, 93B

Build docker image and run it locally:

To run a Docker container in the background, we need to include the -d flag in the docker run command.



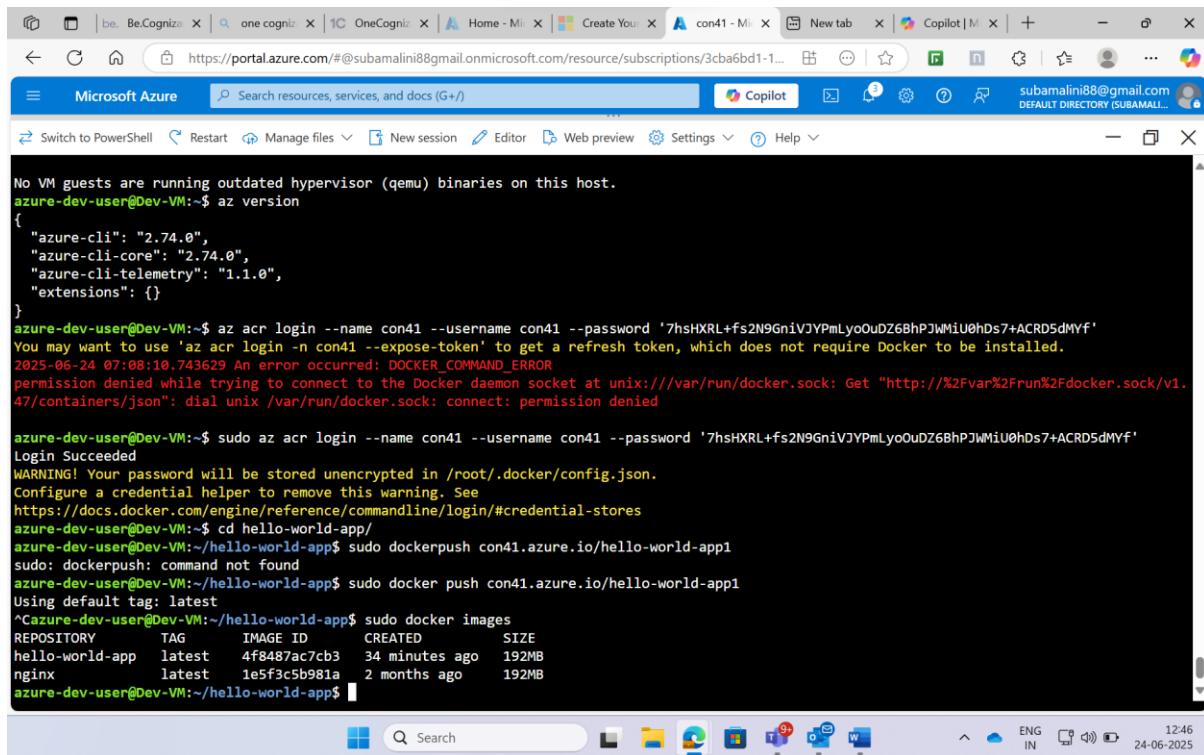
```
Step 2/4 : COPY . /usr/share/nginx/html
--> 585a70578d82
Step 3/4 : EXPOSE 80
--> Running in a9ef82bc434e
--> Removed intermediate container a9ef82bc434e
--> 3e815d127459
Step 4/4 : CMD ["nginx", "-g", "daemon off;"]
--> Running in 61987cea2ca4
--> Removed intermediate container 61987cea2ca4
--> 4f8487ac7cb3
Successfully built 4f8487ac7cb3
Successfully tagged hello-world-app:latest
azure-dev-user@Dev-VM:~/hello-world-app$ sudo docker run -p 80:80 hello-world-app
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2025/06/24 06:44:12 [notice] 1#1: using the "epoll" event method
2025/06/24 06:44:12 [notice] 1#1: nginx/1.27.5
2025/06/24 06:44:12 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2025/06/24 06:44:12 [notice] 1#1: OS: Linux 6.11.0-1015-azure
2025/06/24 06:44:12 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2025/06/24 06:44:12 [notice] 1#1: start worker processes
2025/06/24 06:44:12 [notice] 1#1: start worker process 29
```

Create a container registry with unique registry name:

The screenshot shows the Microsoft Container Registry Overview page. The main content area displays a green checkmark indicating "Your deployment is complete". Below this, deployment details are listed: Deployment name: Microsoft.ContainerRegistry, Subscription: Azure subscription 1, Start time: 6/24/2025, 12:25:06 PM, Correlation ID: 09af9ef4-8eb4-4ab5-9b28-947... Resource group: DevEnvironment-RG. There are sections for "Deployment details" and "Next steps", with a "Go to resource" button. A sidebar on the right provides links to Cost management, Microsoft Defender for Cloud, Free Microsoft tutorials, and Work with an expert.

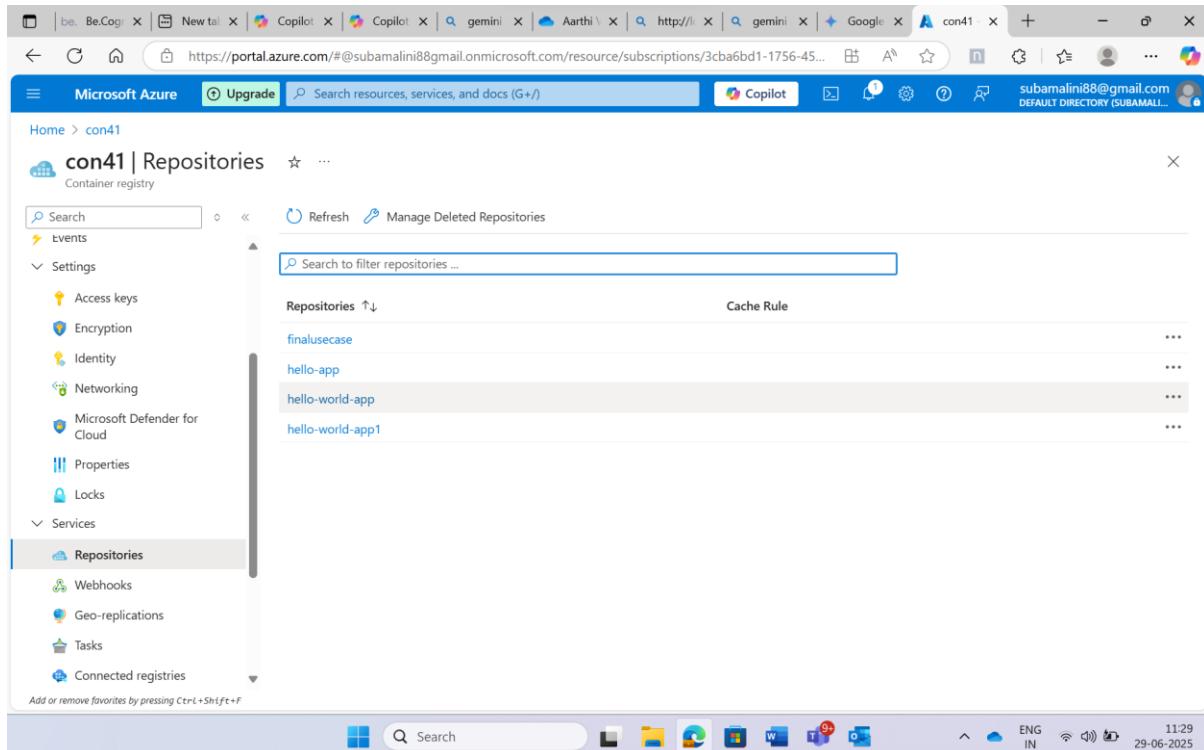
The screenshot shows the Microsoft Container Registry Access keys page for the "con41" registry. It lists access keys: Name: password, Password: 7hsHXRL+fs2N9GniVJYPmLyoOuDZ6BhPJW..., Status: Copied. There is also a "Regenerate" button. The sidebar on the left includes options like Overview, Activity log, Access control (IAM), Tags, Quick start, Resource visualizer, Events, Settings, and Access keys (which is selected). The bottom of the screen shows the Windows taskbar with various pinned icons and system status.

We are logging in ACR using registry name, username and password
Then push the docker image from local to the ACR:



```
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
azure-dev-user@Dev-VM:~$ az version  
{  
    "azure-cli": "2.74.0",  
    "azure-cli-core": "2.74.0",  
    "azure-cli-telemetry": "1.1.0",  
    "extensions": {}  
}  
azure-dev-user@Dev-VM:~$ az acr login --name con41 --username con41 --password '7hsHXRL+fs2N9GniVJYPMlyOuDZ6BhPJWMiU0hDs7+ACRD5dMYF'  
You may want to use 'az acr login -n con41 --expose-token' to get a refresh token, which does not require Docker to be installed.  
2025-06-24 07:08:08 @0.743629 An error occurred: DOCKER_COMMAND_ERROR  
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%%2Fvar%%2Frun%%2Fdocker.sock/v1.47/containers/json": dial unix /var/run/docker.sock: connect: permission denied  
  
azure-dev-user@Dev-VM:~$ sudo az acr login --name con41 --username con41 --password '7hsHXRL+fs2N9GniVJYPMlyOuDZ6BhPJWMiU0hDs7+ACRD5dMYF'  
Login Succeeded  
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.  
Configure a credential helper to remove this warning. See  
https://docs.docker.com/engine/reference/commandline/login/#credential-stores  
azure-dev-user@Dev-VM:~$ cd hello-world-app/  
azure-dev-user@Dev-VM:~/hello-world-app$ sudo dockerpush con41.azure.io/hello-world-app1  
sudo: dockerpush: command not found  
azure-dev-user@Dev-VM:~/hello-world-app$ sudo docker push con41.azure.io/hello-world-app1  
Using default tag: latest  
^Cazure-dev-user@Dev-VM:~/hello-world-app$ sudo docker images  
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE  
hello-world-app    latest   4f8487ac7cb3   34 minutes ago  192MB  
nginx              latest   1e5f3c5b981a   2 months ago   192MB  
azure-dev-user@Dev-VM:~/hello-world-app$
```

The hello-world-app file will be displayed at Repositories



Azure Kubernetes Service (AKS)

- **Cluster Name:** Dev-AKS
- **Node Count:** 1
- **Node Size:** Standard_B2s (Free Tier eligible)

The screenshot shows the 'Create Kubernetes cluster' wizard in the Azure portal. It has three main sections: 'Project details', 'Cluster details', and a summary step.

Project details: Subscription is set to 'Azure subscription 1' and Resource group is 'DevEnvironment-RG'. A 'Create new' button is available.

Cluster details: Cluster preset configuration is 'Dev/Test'. A note says: 'To quickly customize your Kubernetes cluster, choose one of the preset configurations above. You can modify these configurations at any time.' A 'Compare presets' link is provided. The Kubernetes cluster name is 'aks41'.

Summary Step: Buttons for 'Previous', 'Next', and 'Review + create' are visible. A 'Give feedback' link is also present.

Set the node size as Standard_D2_v6 and node count to 1 and set max pods as 30

The screenshot shows the AKS service configuration page for 'aks1741'. The left sidebar includes 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Monitor', 'Diagnose and solve problems', 'Microsoft Defender for Cloud (preview)', 'Cost analysis', 'Resource visualizer', 'Namespaces', 'Workloads', 'Services and ingresses', 'Storage', and 'Configuration' sections.

Kubernetes services: Encryption type is 'Encryption at-rest with a platform-managed key'. Virtual node pools are 'Not enabled'.

Node pools: Node pools are '1 node pool'. Kubernetes versions are '1.31.9'. Node sizes are 'Standard_D2s_v6'.

Upgrades: Kubernetes version is '1.31.9'. Auto Upgrade Type is 'Patch'. Automatic upgrade scheduler is 'Every week on Sunday (recommended)'. Node upgrade channel type is 'Node Image'. Node upgrade channel scheduler is 'Every week on Sunday (recommended)'.

Networking: API server address is 'aks1741-dns-7thd0epp.hcp.eastus.azmk8s.io'. Network configuration is 'Azure CNI Overlay'. Pod CIDR is '10.244.0.0/16'. Service CIDR is '10.0.0.0/16'. DNS service IP is '10.0.0.10'. Cilium dataplane is 'Not enabled'. Network Policy is 'None'. Load balancer is 'Standard'. Private cluster is 'Not enabled'. Authorized IP ranges are 'Not enabled'. Application Gateway ingress controller is 'Not enabled'.

Integrations: Container insights is 'Not enabled'. Workspace resource ID is '-'. Service Mesh - Istio is 'Not enabled'.

Make sure the AKS is connected with your container registry:

More events in the activity log → Dismiss all

Successfully attached the container registry

Successfully attached the container registry 'con41'. 4 minutes ago

₹16,894.81 credit remaining Subscription 'Azure subscription 1' has a remaining credit of ₹16,894.81. Upgrade to a Pay-As-You-Go subscription. an hour ago

Resource group: DevEnvironment-RG
Kubernetes version: 1.31.9
Power state: Running
Cluster operation status: Succeeded
Subscription: Azure subscription 1
Location: East US
Subscription ID: 3cba6bd1-1756-45fe-a43a-09f0f012979d
Fleet Manager: Click here to assign
Tags (edit) Add tags

After applied deployment.yml under Workloads.

Deployment name: All Add label filter

Name	Namespace	Ready	Age	CPU	Memory
coredns	kube-system	2/2	1 day	0%	5%
coredns-autoscaler	kube-system	1/1	1 day	0%	4%
konnectivity-agent	kube-system	2/2	1 day	0%	1%
konnectivity-agent-autosc	kube-system	1/1	1 day	0%	3%
metrics-server	kube-system	2/2	1 day	0%	12%
eraser-controller-manager	kube-system	1/1	1 day	0%	2%
azure-wi-webhook-control	kube-system	2/2	1 day	0%	5%
ama-metrics	kube-system	2/2	1 day	0.07%	1%
ama-metrics-ksm	kube-system	1/1	1 day	0%	0.81%
ama-metrics-operator-tar	kube-system	1/1	1 day	0%	0.84%
hello-world-app	default	1/1	1 day	-	-

After applied service.yaml under Services and ingresses:

The screenshot shows the Azure portal interface for the 'aks41' Kubernetes service. The left sidebar includes options like Overview, Activity log, Access control (IAM), Tags, Monitor, Diagnose and solve problems, Microsoft Defender for Cloud (preview), Cost analysis, Resource visualizer, Kubernetes resources (Namespaces, Workloads, Services and ingresses, Storage, Configuration), and a Copilot icon. The main content area displays the 'Services' tab of the 'Services and ingresses' blade. A table lists the following services:

Name	Namespace	Status	Type	Cluster IP	External IP	Ports
kubernetes	default	Ok	ClusterIP	10.0.0.1		443/TCP
kube-dns	kube-system	Ok	ClusterIP	10.0.0.10		53/UDP,53/TCP
metrics-server	kube-system	Ok	ClusterIP	10.0.39.234		443/TCP
azure-wi-webhook-web...	kube-system	Ok	ClusterIP	10.0.160.208		443/TCP
ama-metrics-ksm	kube-system	Ok	ClusterIP	10.0.113.245		8080/TCP
ama-metrics-operator-t...	kube-system	Ok	ClusterIP	10.0.25.165		80/TCP
network-observability	kube-system	Ok	ClusterIP	10.0.168.57		10093/TCP
hello-world-service	default	Ok	LoadBalancer	10.0.167.150	128.203.70.207	80:32330/TCP

The service.yaml got executed and IP address is created

The screenshot shows a web browser displaying a deployed application. The page content reads:

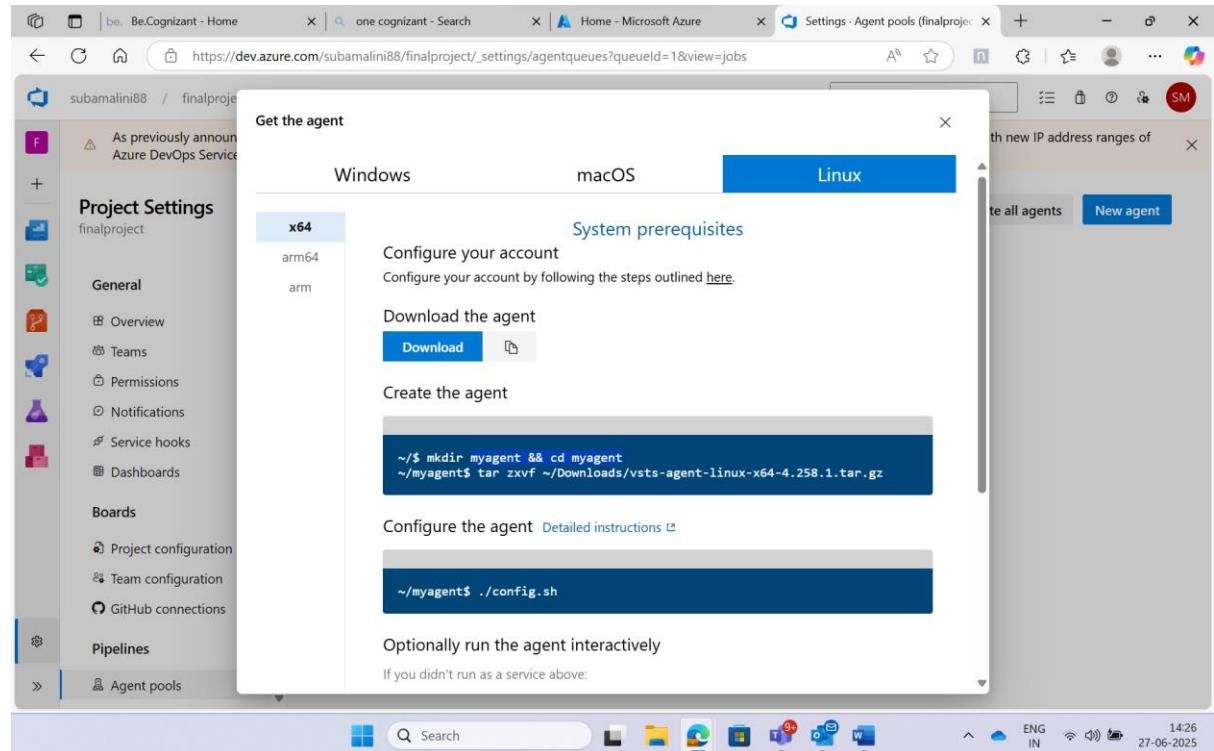
Hello, World!

Welcome to your first deployed web application!!

The browser taskbar shows several open tabs, including 'be. Be.Cog', 'aks174', 'Hello', 'Copilot', 'deploy', 'vnet a...', 'youtu...', 'Day-1...', and 'Day-1...'. The system tray at the bottom right of the screen shows the date and time as 11:14 on June 26, 2025.

To work in the Azure devOps we first configure the self-hosted agent:

Downloaded the Azure Pipelines agent package for Linux from the official Azure DevOps portal



Downloading the self-hosted agent:

A screenshot of a Microsoft Edge browser window showing a terminal session on an Azure VM named 'Dev-VM'. The session displays the process of downloading the Azure Pipelines agent. The terminal output includes system load statistics, security updates information, and the wget command used to download the agent. The wget command shows progress: 'vsts-agent-linux-x64-4.258.1.tar.gz 100%[=====]' and a download speed of '140.73M 145MB/s in 1.0s'. The terminal prompt is 'azure-dev-user@Dev-VM:~\$'. The browser's address bar shows the URL: https://portal.azure.com/#@subamalini88@gmail.onmicrosoft.com/resource/subscriptions/3cba6bd1-1756-45... The system tray at the bottom right shows the date as 27-06-2025.

We need to provide the end user agreements to download the self-hosted agent:

After the end user agreement, we can register the self-hosted agent using the personal access token:

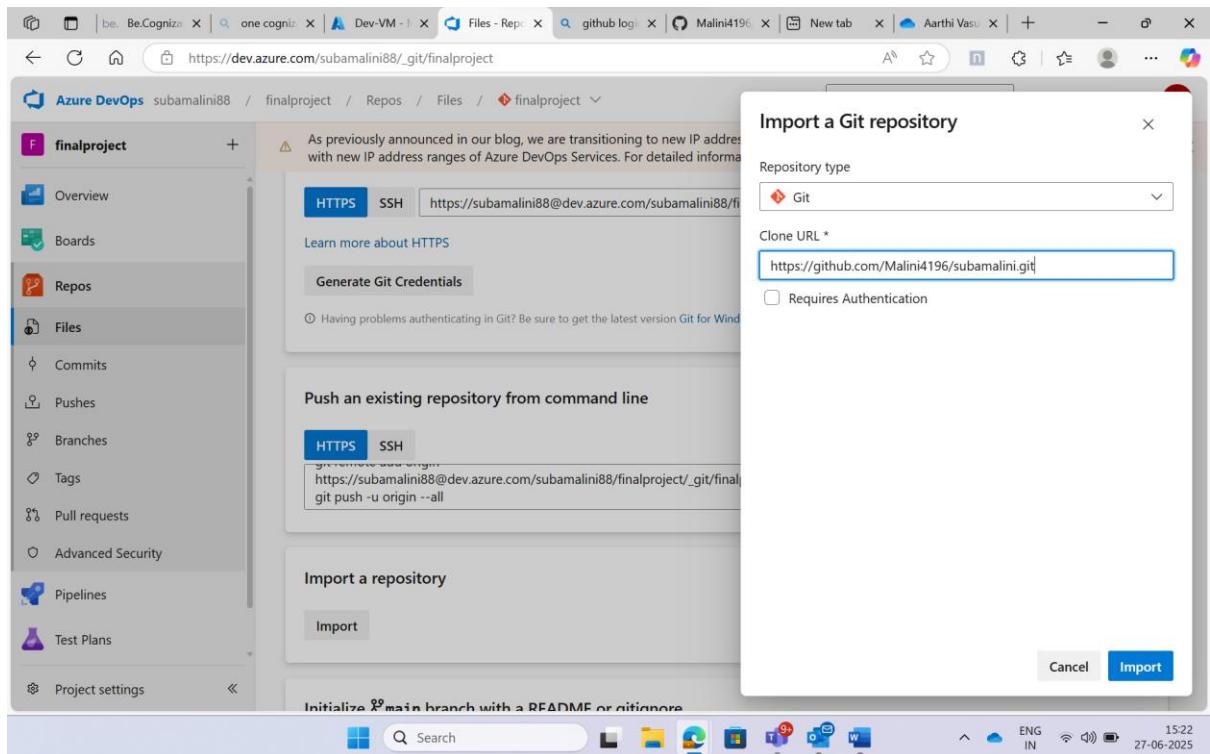
Self-hosted agent is created:

The screenshot shows the 'Agent pools' section of the Azure DevOps interface. On the left, there's a sidebar with project settings like General, Boards, Pipelines, and Agent pools. The 'Agent pools' tab is selected. In the main area, under the 'Default' pool, there's a table with columns: Name, Last run, Current status, Agent version, and Enabled. One row shows a 'self-host agent' which is marked as 'Online'. There are buttons for 'Update all agents' and 'New agent' at the top right of the table.

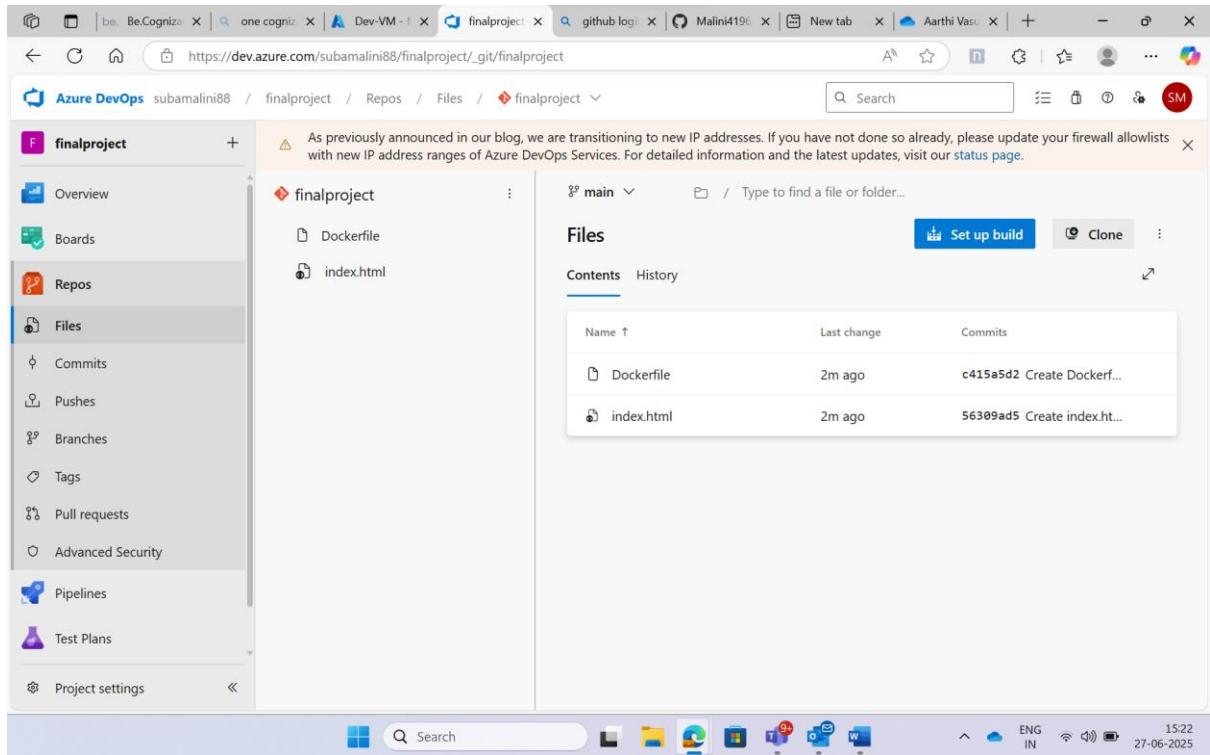
Then we need to commit our "index.html" and "Dockerfile" to the repository:

The screenshot shows a GitHub repository page for 'subamalini'. The 'Code' tab is selected. Under the 'main' branch, there are two files listed: 'Dockerfile' and 'index.html'. Both files were committed by 'Malini4196' with the message 'Create Dockerfile' and 'Create index.html' respectively, both occurring 'now'.

Then we need to clone the repository:



Now the files are cloned successfully:



Created a pipeline using classic editor to automate the build, test, and deployment process of applications, enabling continuous integration and delivery (CI/CD) in Azure DevOps:

The screenshot shows the Azure DevOps Pipelines interface. On the left, a sidebar menu for the project 'finalproject' includes options like Overview, Boards, Repos, Pipelines (which is selected), Environments, Releases, Library, Task groups, Deployment groups, Test Plans, Artifacts, and Project settings. The main content area features a large illustration of a robot and a person working together. Below the illustration, the text 'Create your first Pipeline' is prominently displayed, followed by the sub-instruction 'Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.' A blue 'Create Pipeline' button is located at the bottom of this section. At the very bottom of the screen, a Windows taskbar is visible with various pinned icons.

Select the classic editor:

The screenshot shows the 'Where is your code?' step of the pipeline creation wizard. The interface is similar to the previous one, with the 'Pipelines' option selected in the sidebar. The main area displays a list of source code providers: 'Azure Repos Git (YAML)', 'Bitbucket Cloud (YAML)', 'GitHub (YAML)', 'GitHub Enterprise Server (YAML)', 'Other Git (Any generic Git repository)', and 'Subversion'. Below this list, a note says 'Use the classic editor to create a pipeline without YAML.' The bottom of the screen shows the Windows taskbar.

Select the source as Azure Repos Git:

The screenshot shows the Azure DevOps Pipelines interface. On the left, there's a sidebar with project navigation. The main area has a large circular arrow icon. To the right, a section titled "Select a source" displays several options: "Azure Repos Git" (selected), "GitHub", "GitHub Enterprise Server", "Subversion", "Bitbucket Cloud", and "Other Git". Below this, under "Team project", "finalproject" is selected. Under "Repository", "finalproject" is also selected. Under "Default branch for manual and scheduled builds", "main" is selected. A note at the top right says: "As previously announced in our blog, we are transitioning to new IP addresses. If you have not done so already, please update your firewall allowlists with new IP address ranges of Azure DevOps Services. For detailed information and the latest updates, visit our status page."

Go with the empty job:

The screenshot shows the Azure DevOps Pipelines interface. The sidebar includes "Pipelines" (selected). The main area has a large circular arrow icon. To the right, a section titled "Select a template" shows a search bar with "Empty job" and a "Search" button. Below it, "Configuration as code" is mentioned with a "YAML" link. A "Featured" section lists several pipeline types: ".NET Desktop" (Build and test a .NET or Windows classic desktop solution), "Android" (Build, test, sign, and align an Android APK), ".NET" (Build and test an ASP.NET web application), and "Azure Web App for ASP.NET" (Build, package, test, and deploy an ASP.NET Azure Web App). A note at the top right is identical to the previous screenshot.

Set the Agent pool as Default since we are using classic editor.

The screenshot shows the Azure DevOps Pipeline Editor for a project named 'finalproject'. The pipeline consists of three main steps: 'Get sources', 'Agent job 1' (which contains 'Command Line Script', 'Copy Files to: \$(Build.ArtifactStagingDirectory)', and 'Publish Pipeline Artifact'), and 'Publish Pipeline Artifact'. On the right side of the editor, there is a panel for 'Agent job 1' where the 'Agent pool' dropdown is set to 'Default'. Other settings visible include 'Demands' and 'Execution plan'.

Added Command line, Copy files, Publish pipeline artifact to Agent jobs.

The screenshot shows the Azure DevOps Pipeline Editor with a search bar at the top right containing the text 'publish'. A list of tasks is displayed under the heading 'Add tasks': 'PyPI publisher', 'Publish Pipeline Artifacts', 'Publish Pipeline Metadata', and 'Index sources and publish symbols'. The pipeline structure is identical to the previous screenshot, with 'Get sources', 'Agent job 1' (containing 'Command Line Script', 'Copy Files to:', and 'Publish Pipeline Artifact'), and 'Publish Pipeline Artifact' steps.

Using command line script build the dockerfile. Log into ACR using az. Push the docker image into ACR.

The screenshot shows the Azure DevOps Pipeline Editor for a project named 'finalproject'. The pipeline consists of two main stages: 'Get sources' and 'Agent job 1'. The 'Agent job 1' stage contains a 'Command Line Script' task. The task's 'Script' field contains the following commands:

```
sudo docker build -t con41.azurecr.io/hello-app:${Build.BuildId} .  
az acr login --name con41 --username con41 --password  
7hsHXRL+fs2N9GniVJYPMlyoOuDZ6BhPJWMiUhDs7+ACRD5dM  
Yf  
sudo docker push con41.azurecr.io/hello-app:${Build.BuildId}
```

Copy files to Build.ArtifactStagingDirectory and target folder as Build.ArtifactStagingDirectory.

The screenshot shows the Azure DevOps Pipeline Editor for the same project 'finalproject'. The pipeline now includes a 'Copy Files to: \$(Build.ArtifactStagingDirectory)' task, which is part of the 'Agent job 1' stage. This task copies files from the 'Source Folder' to the 'Target Folder', both specified as '\$(Build.ArtifactStagingDirectory)'.

Add file or directory path as Build.ArtifactStagingDirectory

The screenshot shows the Azure DevOps Pipeline Editor for a project named 'finalproject'. On the left, the 'Pipelines' section is selected. The main area displays a pipeline named 'finalproject-Cl' with the following tasks:

- Get sources
- Agent job 1
 - Command Line Script
 - Copy Files to: \${Build.ArtifactStagingDirectory}
- Publish Pipeline Artifact

The 'Publish Pipeline Artifact' task is currently selected. In the configuration pane on the right, the 'File or directory path' field is set to '\$(Build.ArtifactStagingDirectory)'. Other settings include 'Display name' (Publish Pipeline Artifact), 'Artifact name' (empty), 'Artifact publish location' (Azure Pipelines), and 'Custom properties' (empty). The pipeline version is set to 1.*.

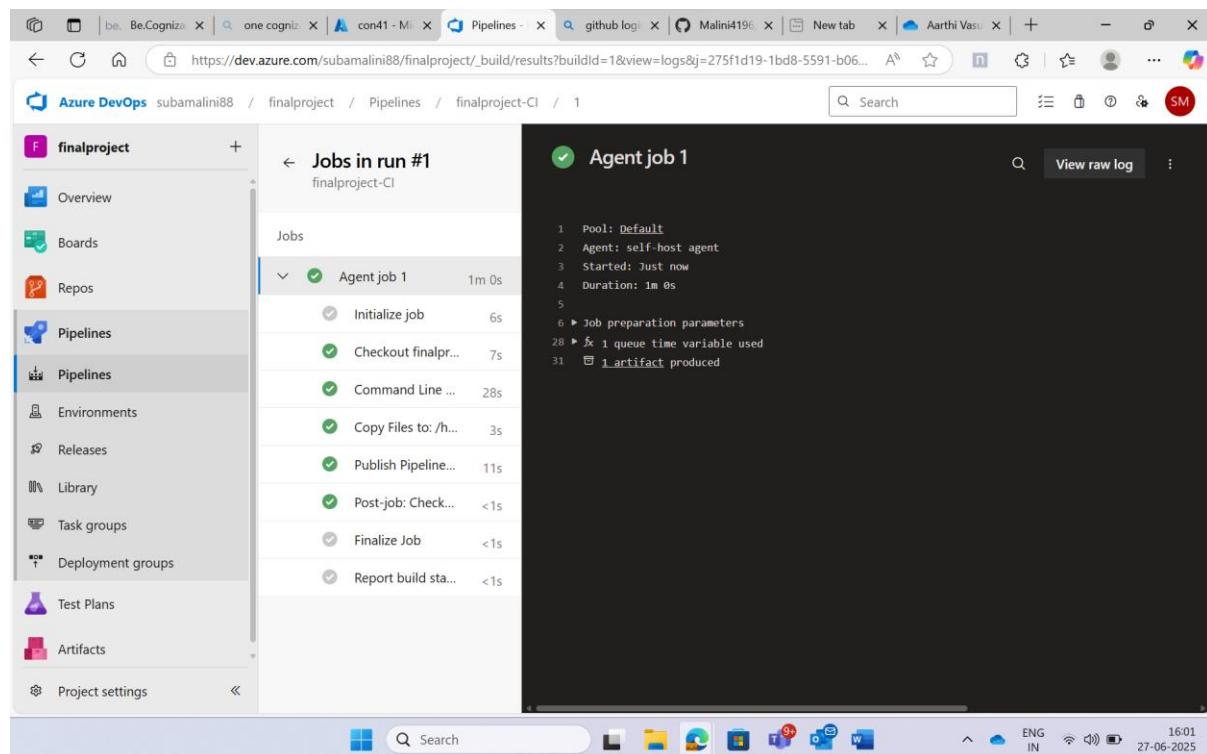
Save and Run the pipeline.

The screenshot shows the 'Run pipeline' dialog box overlaid on the Azure DevOps interface. The dialog box contains the following fields:

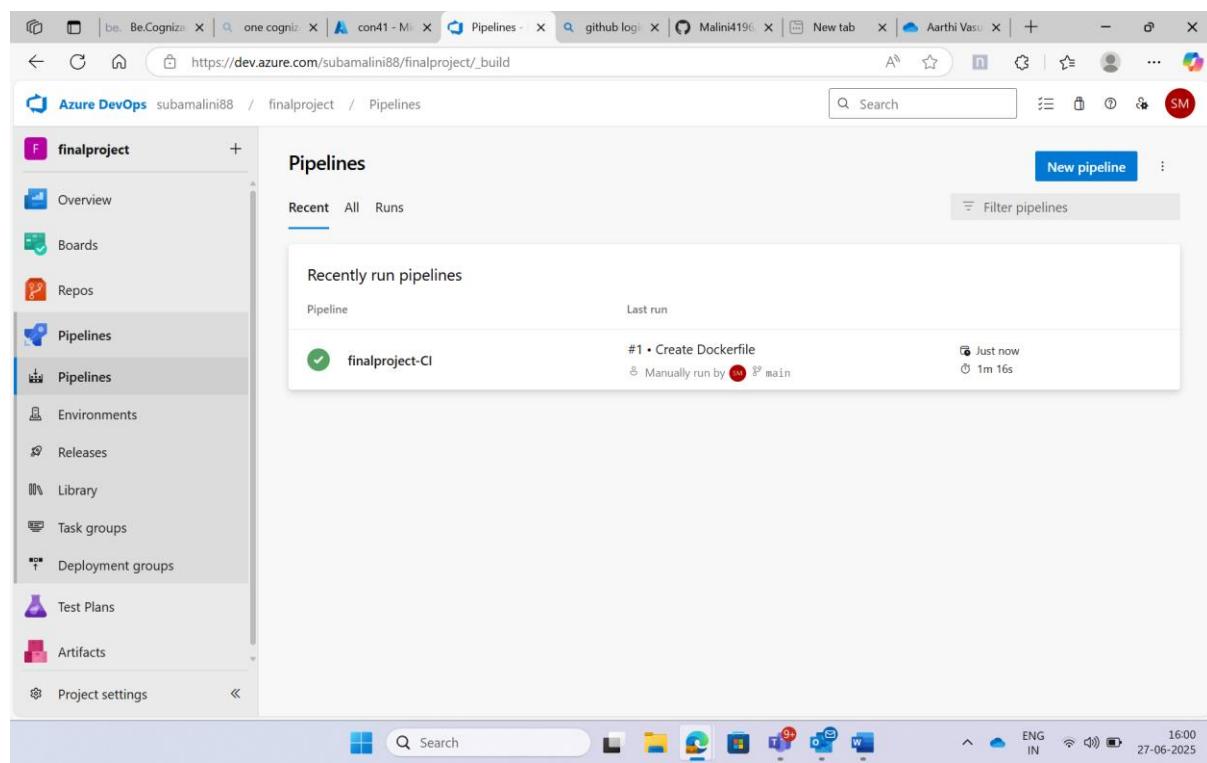
- Save comment: 'run'
- Agent pool: 'Default'
- Branch/tag: 'main'
- Advanced options:
 - Variables: '1 variable defined'
 - Demands: 'This pipeline has no defined demands'

At the bottom of the dialog box are 'Cancel' and 'Save and run' buttons. The status bar at the bottom of the screen shows the date and time as 27-06-2025 15:59.

The pipeline has been created successfully.



A screenshot of a web browser showing the Azure DevOps Pipelines interface. The URL is https://dev.azure.com/subamalini88/finalproject/_build/results?buildId=1&view=logs&j=275f1d19-1bd8-5591-b06.... The left sidebar shows the project navigation with 'Pipelines' selected. The main area displays 'Jobs in run #1' for 'finalproject-Cl'. A detailed log for 'Agent job 1' is shown, listing tasks: Initialize job (6s), Checkout finalpr... (7s), Command Line ... (28s), Copy Files to: /h... (3s), Publish Pipeline... (11s), Post-job: Check... (<1s), Finalize Job (<1s), and Report build sta... (<1s). The log indicates a duration of 1m 0s, started just now, and used 1 queue time variable. One artifact was produced.



A screenshot of a web browser showing the Azure DevOps Pipelines overview page. The URL is https://dev.azure.com/subamalini88/finalproject/_build. The left sidebar shows the project navigation with 'Pipelines' selected. The main area displays the 'Pipelines' section, showing 'Recently run pipelines'. It lists one pipeline: 'finalproject-Cl' (Last run: #1 • Create Dockerfile, Manually run by main, Just now, 1m 16s). A 'New pipeline' button is visible at the top right.

Continuous Deployment(CD) : Release pipeline

Created a release pipeline to automate the deployment of application builds to various environments, ensuring consistent and controlled releases with approval workflows and rollback options

The screenshot shows the Azure DevOps interface for a project named 'finalproject'. The left sidebar has 'Pipelines' selected under 'Releases'. The main area displays a cartoon illustration of a person launching a rocket from a blue cube. Below the illustration, the text 'No release pipelines found' is centered. A call-to-action button 'New pipeline' is visible. The browser address bar shows the URL https://dev.azure.com/subamalini88/finalproject/_release. The bottom status bar indicates the date as 27-06-2025.

In artifact select the source as finalproject-CI

The screenshot shows the 'New release pipeline' configuration page. The left sidebar has 'Artifacts' selected under 'Pipelines'. The main area shows a 'Source type' section with 'Build' selected. To the right, there are fields for 'Project' (set to 'finalproject'), 'Source (build pipeline)' (set to 'finalproject-CI'), 'Default version' (set to 'Latest'), and 'Source alias' (set to '_finalproject-CI'). A note at the bottom states: 'The artifacts published by each version will be available for deployment in release pipelines. The latest successful build of finalproject-CI published the following artifacts: Job1.' A large 'Add' button is at the bottom right. The browser address bar shows the URL https://dev.azure.com/subamalini88/finalproject/_releaseDefinition?definitionId=0&a=action-create-definition&s.... The bottom status bar indicates the date as 27-06-2025.

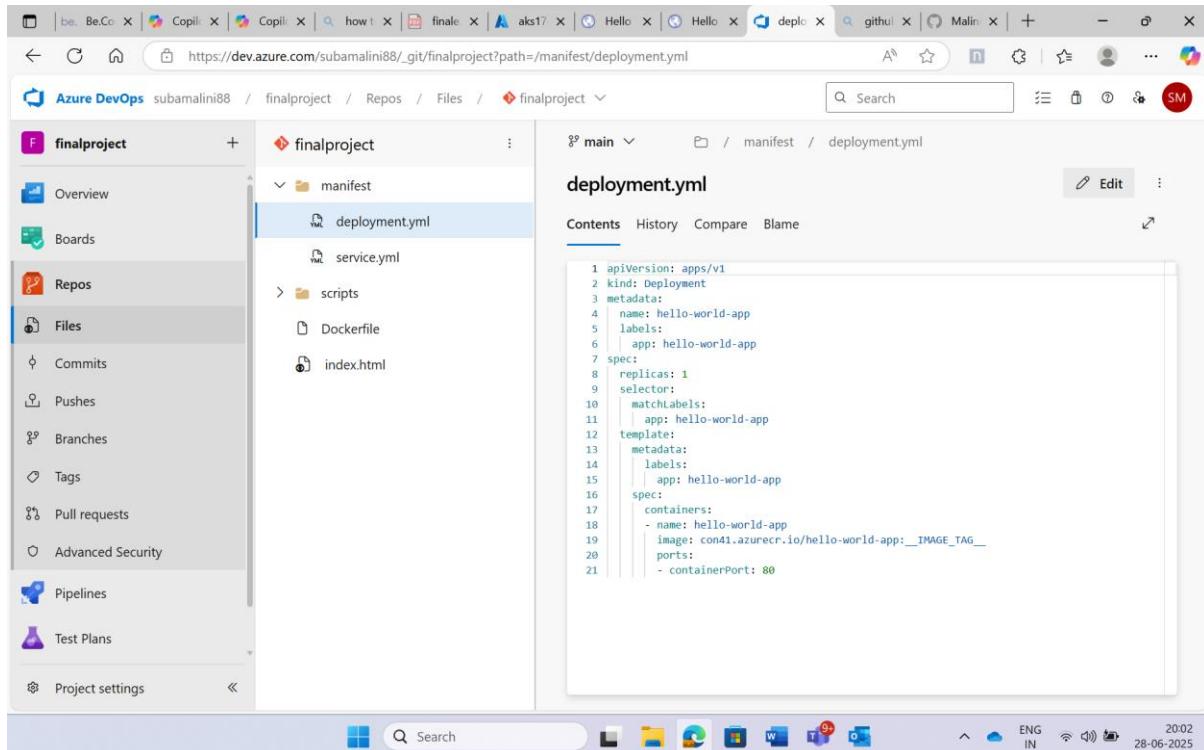
Select the stage owner name in Stage 1

The screenshot shows the Azure DevOps Pipeline Editor. On the left, there's a sidebar with 'finalproject' selected. Under 'Pipelines', 'Releases' is also selected. The main area shows a 'Stages' section with 'Stage 1' highlighted. A modal window titled 'Properties' is open over Stage 1, showing the 'Stage name' as 'Stage 1' and the 'Stage owner' as 'Suba Malini'. The bottom right corner of the screen shows the date and time as 27-06-2025.

Enable continuous deployment trigger:

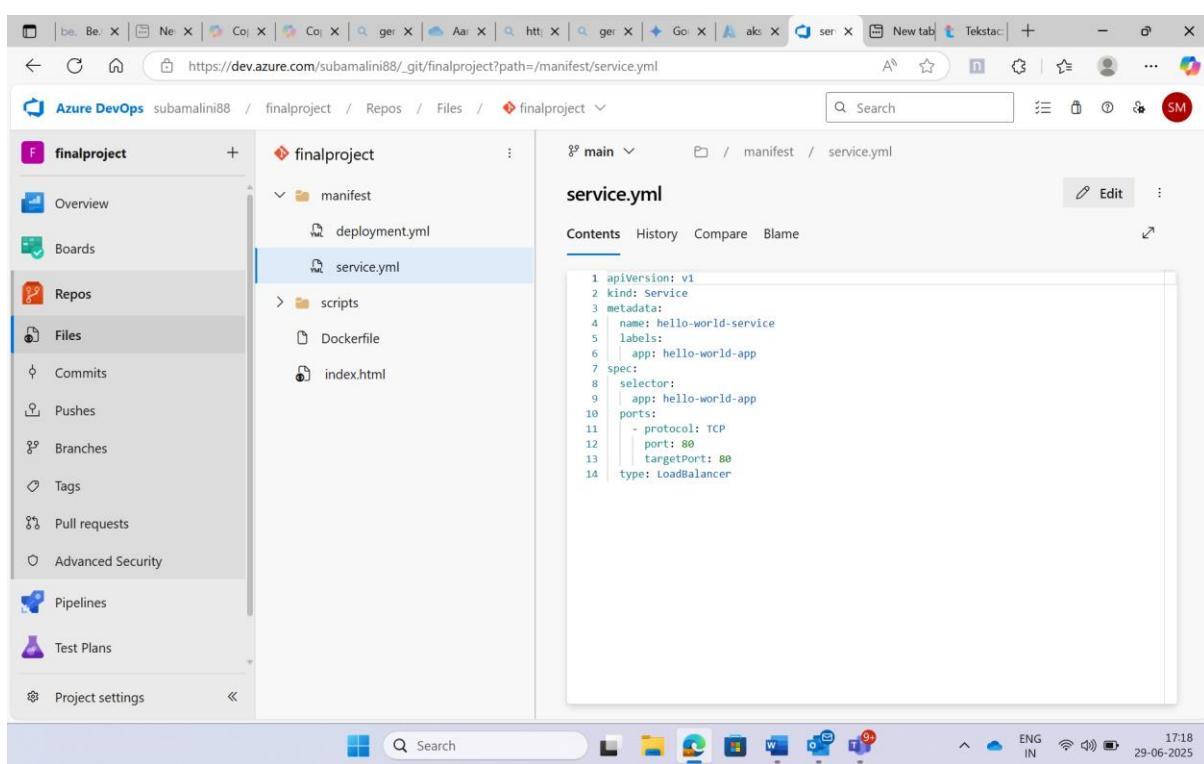
The screenshot shows the Azure DevOps Pipeline Editor. The sidebar has 'finalproject' selected under 'Pipelines'. In the main area, the 'Stages' section shows 'Stage 1' and an 'Artifacts' section with '_finalproject-Cl'. On the right, a 'Continuous deployment trigger' panel is open, showing the 'Enabled' toggle switch turned on. Below it, a 'Build branch filters' section says 'No filters added.' A 'Pull request trigger' section is also visible, with a note that it's disabled. The bottom right corner shows the date and time as 27-06-2025.

Added a manifest folder and in that add deployment.yaml and service.yaml



The screenshot shows the Azure DevOps interface for a repository named 'finalproject'. The 'Files' tab is selected in the sidebar. Inside the 'manifest' folder, there are two files: 'deployment.yaml' and 'service.yaml'. The 'deployment.yaml' file is selected and its contents are displayed in the main editor area:

```
1 apiVersion: apps/v1
2 kind: Deployment
3 metadata:
4   name: hello-world-app
5   labels:
6     app: hello-world-app
7 spec:
8   replicas: 1
9   selector:
10    matchLabels:
11      app: hello-world-app
12   template:
13     metadata:
14       labels:
15         app: hello-world-app
16     spec:
17       containers:
18         - name: hello-world-app
19           image: con41.azurecr.io/hello-world-app:__IMAGE_TAG__
20           ports:
21             - containerPort: 80
```

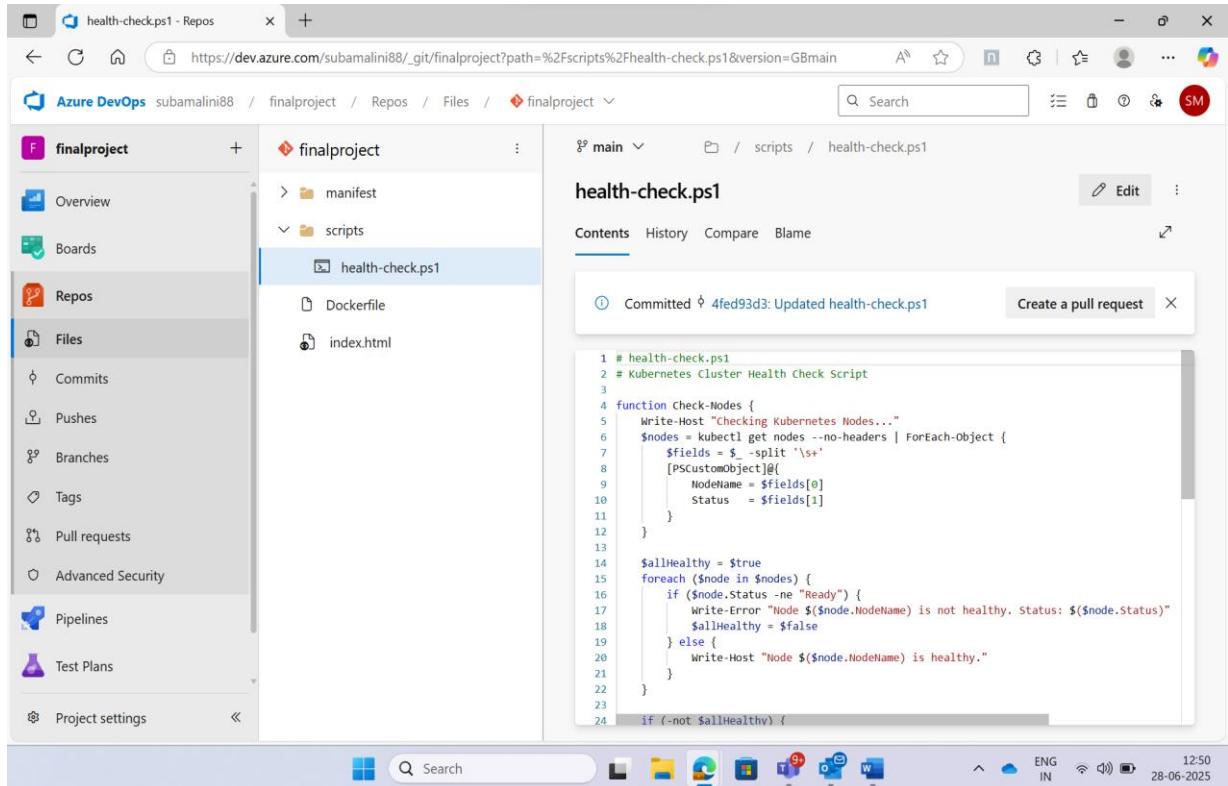


The screenshot shows the Azure DevOps interface for the same repository. The 'Files' tab is selected in the sidebar. Inside the 'manifest' folder, there are two files: 'deployment.yaml' and 'service.yaml'. The 'service.yaml' file is selected and its contents are displayed in the main editor area:

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: hello-world-service
5   labels:
6     app: hello-world-app
7 spec:
8   selector:
9     app: hello-world-app
10  ports:
11    - protocol: TCP
12      port: 80
13      targetPort: 80
14  type: LoadBalancer
```

Powershell:

Created a folder “scripts” added a file “health-check.ps1”



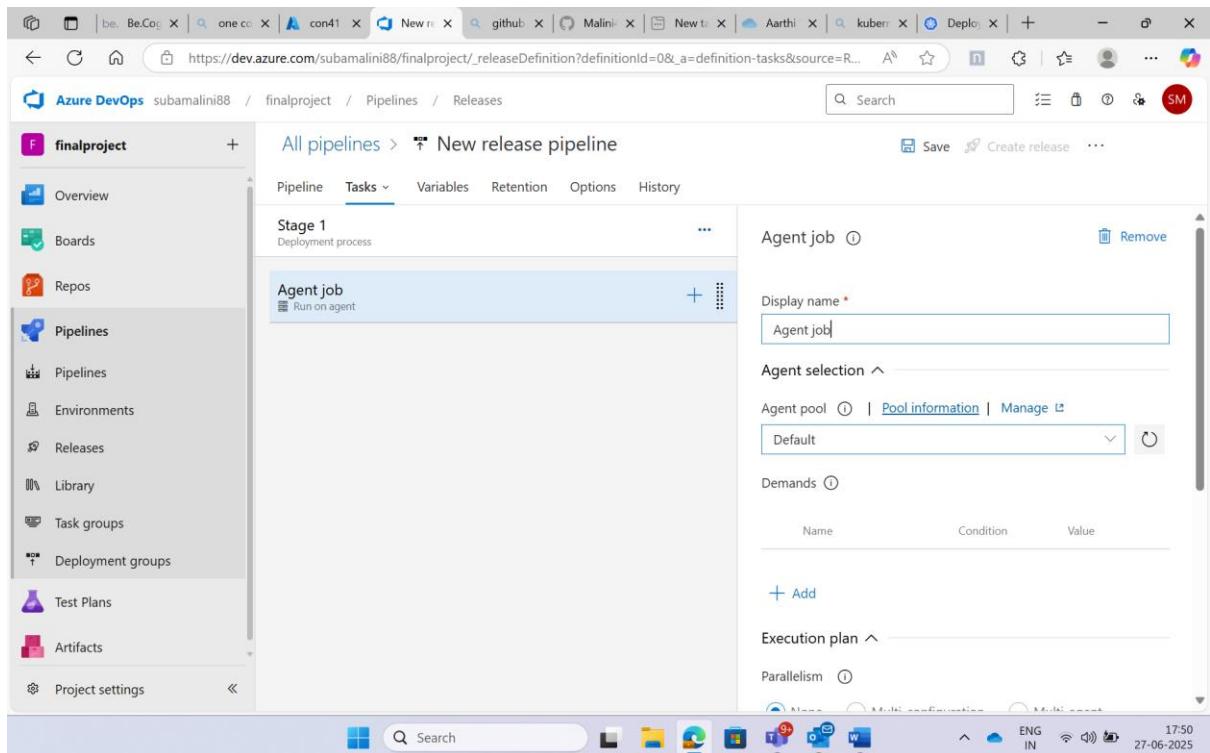
The screenshot shows the Azure DevOps interface for a repository named "finalproject". The "Files" tab is selected. Inside the "scripts" folder, there is a file named "health-check.ps1". The code editor displays the following PowerShell script:

```
# health-check.ps1
# Kubernetes Cluster Health Check Script

function Check-Nodes {
    Write-Host "Checking Kubernetes Nodes..."
    $nodes = kubectl get nodes --no-headers | ForEach-Object {
        $fields = $_ -split '\s+'
        [PSCustomObject]@{
            NodeName = $fields[0]
            Status   = $fields[1]
        }
    }
}

$allHealthy = $true
foreach ($node in $nodes) {
    if ($node.Status -ne "Ready") {
        Write-Error "Node $($node.NodeName) is not healthy. Status: $($node.Status)"
        $allHealthy = $false
    } else {
        Write-Host "Node $($node.NodeName) is healthy."
    }
}
if (-not $allHealthy) {
```

Select the Agent pool to Default in Agent job



The screenshot shows the Azure DevOps interface for a pipeline named "finalproject". The "Pipelines" tab is selected. A new release pipeline is being created. In the "Agent job" section, the "Agent selection" dropdown is set to "Default".

Replace Tokens is a task that searches for placeholder tokens in your files and replaces them with actual values from variable group:

The screenshot shows the Azure DevOps Pipelines interface for a 'finalproject' pipeline. The 'Tasks' tab is selected, and under 'Stage 1: Deployment process', there is an 'Agent job' named 'Replace tokens'. This task is currently selected. The configuration pane on the right shows the following settings:

- Task version:** 6.*
- Display name:** Replace tokens
- Root:** \$(System.DefaultWorkingDirectory)/_finalproject-CI/Job1/manifest
- Source files:** deployment.yml
- Token pattern:** custom
- Token prefix:** (empty)

Set the variable in the deployment.yml file:

The screenshot shows the Azure DevOps Pipelines interface for a 'finalproject' pipeline. The 'Variables' tab is selected. Under 'Pipeline variables', there is a table with one entry:

Name	Value
IMAGE_TAG	\$(Build.BuildId)

In Agent job add Kubernetes deploy for deployment and service. Select the cluster as Dev-AKS and namespace as default for service connection.

The screenshot shows the Azure DevOps interface for creating a new service connection. On the left, the sidebar is open with 'Pipelines' selected. In the center, under 'All pipelines > New release pipeline', there is a 'Stage 1 Deployment process' section. Within this stage, an 'Agent job Run on agent' task is listed with two 'deploy' steps. A modal window titled 'New service connection' is open on the right, prompting for configuration details. The 'Cluster' dropdown is set to 'aks1741 (DevEnvironment-RG)'. The 'Namespace' dropdown is set to 'default'. The 'Service Connection Name' field contains 'serviceconnection'. The 'Save' button is visible at the bottom right of the modal.

In first deploy add deployment.yaml directory and in second deploy add service.yaml directory

The screenshot shows the Azure DevOps interface for configuring a deployment task. The left sidebar is open with 'Pipelines' selected. In the center, under 'All pipelines > New release pipeline', there is a 'Stage 1 Deployment process' section. Within this stage, an 'Agent job Run on agent' task is listed with three steps: 'deploy', 'Deploy to Kubernetes', 'deploy', 'Deploy to Kubernetes', and 'PowerShell Script'. The 'deploy' step is currently selected. On the right, a detailed configuration pane is open for this step. It shows the 'Service connection type' as 'Kubernetes Service Connection', 'Kubernetes service connection' as 'sc1', 'Namespace' as 'default', 'Strategy' as 'None', and 'Manifests' as '\$(System.DefaultWorkingDirectory)/_finalproject-Cl/Job1/manifest/deploymentyaml'. The 'Containers' field is empty. The 'Save' and 'Create release' buttons are visible at the top right of the configuration pane.

The screenshot shows the Azure DevOps Pipelines interface for a project named 'finalproject'. A new release pipeline is being created. The 'Tasks' tab is selected, showing a single stage named 'Stage 1: Deployment process'. Inside this stage, there is an 'Agent job' task labeled 'deploy' which runs on an 'agent'. This task has two sub-tasks: 'PowerShell Script' and another 'deploy' task. To the right of the tasks, configuration options are visible, including 'Service connection type' set to 'Kubernetes Service Connection' (sc41), 'Namespace' (empty), 'Strategy' (None), 'Manifests' (empty), and 'Containers' (empty). The top navigation bar includes 'Save', 'Create release', and other standard browser controls.

Added Powershell script at agent job and give the script path for health-check.ps1

This screenshot shows the same Azure DevOps interface, but the 'Tasks' tab is now active on the PowerShell task within the 'Agent job' stage. The 'PowerShell' task is configured with the following details: Task version 2.*, Display name 'PowerShell Script', Type 'File Path' (selected), and a Script Path of '\$(System.DefaultWorkingDirectory)/_finalproject-CI/Job1/scripts/health-check.ps1'. The 'View YAML' button is also visible next to the task title. The rest of the pipeline structure remains the same as in the previous screenshot.

Created a new release and the release is successful:

The screenshot shows the Azure DevOps interface for a 'finalproject' pipeline. The 'Releases' section displays a 'New release pipeline > Release-19 > Stage 1' which has 'Succeeded'. The 'Logs' tab is selected, showing the 'Agent job' step with a green checkmark and the status 'Succeeded'. The log details show a series of tasks: 'Initialize job' (succeeded), 'Download artifact - _finalproject-CI - Job1' (succeeded), 'Replace tokens' (succeeded), 'deploy' (succeeded), 'deploy' (succeeded), 'PowerShell Script' (succeeded), and 'Finalize Job' (succeeded). The total duration for the job was 40 seconds. The log output is visible at the bottom of the screen.

Because we got exit status as 0 it got deployed successfully

The screenshot shows the logs for a 'PowerShell Script' task. The log output is as follows:

```
1 [2025-06-24T09:05:21.244238Z ##[section]Starting: PowerShell script
2 2025-06-24T09:05:21.245269Z =====
3 2025-06-24T09:05:21.245298Z Task      : PowerShell
4 2025-06-24T09:05:21.245310Z Description : Run a PowerShell script on Linux, macOS, or Windows
5 2025-06-24T09:05:21.245314Z Version   : 2.247.1
6 2025-06-24T09:05:21.245344Z Author    : Microsoft Corporation
7 2025-06-24T09:05:21.245362Z Help      : https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/powershell
8 2025-06-24T09:05:21.245381Z =====
9 2025-06-24T09:05:24.273720Z Generating script.
10 2025-06-24T09:05:24.280313Z ===== Starting Command Output =====
11 2025-06-24T09:05:24.280379Z [command]/usr/bin/pwsh -NoLogo -NoProfile -NonInteractive -Command . '/home/azureuser/myagent/_work/_te
12 2025-06-24T09:05:24.750584Z Checking Kubernetes Nodes...
13 2025-06-24T09:05:24.812951Z E0624 09:05:24.812085 25492 memcache.go:265] "Unhandled Error" err="couldn't get current server API g
14 2025-06-24T09:05:24.814271Z E0624 09:05:24.813804 25492 memcache.go:265] "Unhandled Error" err="couldn't get current server API g
15 2025-06-24T09:05:24.815754Z E0624 09:05:24.815344 25492 memcache.go:265] "Unhandled Error" err="couldn't get current server API g
16 2025-06-24T09:05:24.817187Z E0624 09:05:24.816855 25492 memcache.go:265] "Unhandled Error" err="couldn't get current server API g
17 2025-06-24T09:05:24.818494Z E0624 09:05:24.818247 25492 memcache.go:265] "Unhandled Error" err="couldn't get current server API g
18 2025-06-24T09:05:24.819752Z The connection to the server localhost:8080 was refused - did you specify the right host or port?
19 2025-06-24T09:05:24.842504Z Checking Kubernetes Pods...
20 2025-06-24T09:05:24.901348Z E0624 09:05:24.900707 25496 memcache.go:265] "Unhandled Error" err="couldn't get current server API g
21 2025-06-24T09:05:24.902588Z E0624 09:05:24.902157 25496 memcache.go:265] "Unhandled Error" err="couldn't get current server API g
22 2025-06-24T09:05:24.904045Z E0624 09:05:24.903725 25496 memcache.go:265] "Unhandled Error" err="couldn't get current server API g
23 2025-06-24T09:05:24.905461Z E0624 09:05:24.905112 25496 memcache.go:265] "Unhandled Error" err="couldn't get current server API g
24 2025-06-24T09:05:24.906836Z E0624 09:05:24.906473 25496 memcache.go:265] "Unhandled Error" err="couldn't get current server API g
25 2025-06-24T09:05:24.907991Z The connection to the server localhost:8080 was refused - did you specify the right host or port?
26 2025-06-24T09:05:24.911858Z All Kubernetes components are healthy.
27 2025-06-24T09:05:24.985060Z
28 2025-06-24T09:05:24.992651Z ##[section]Finishing: PowerShell Script
29
```

Health of the pods and nodes are healthy in Azure portal:

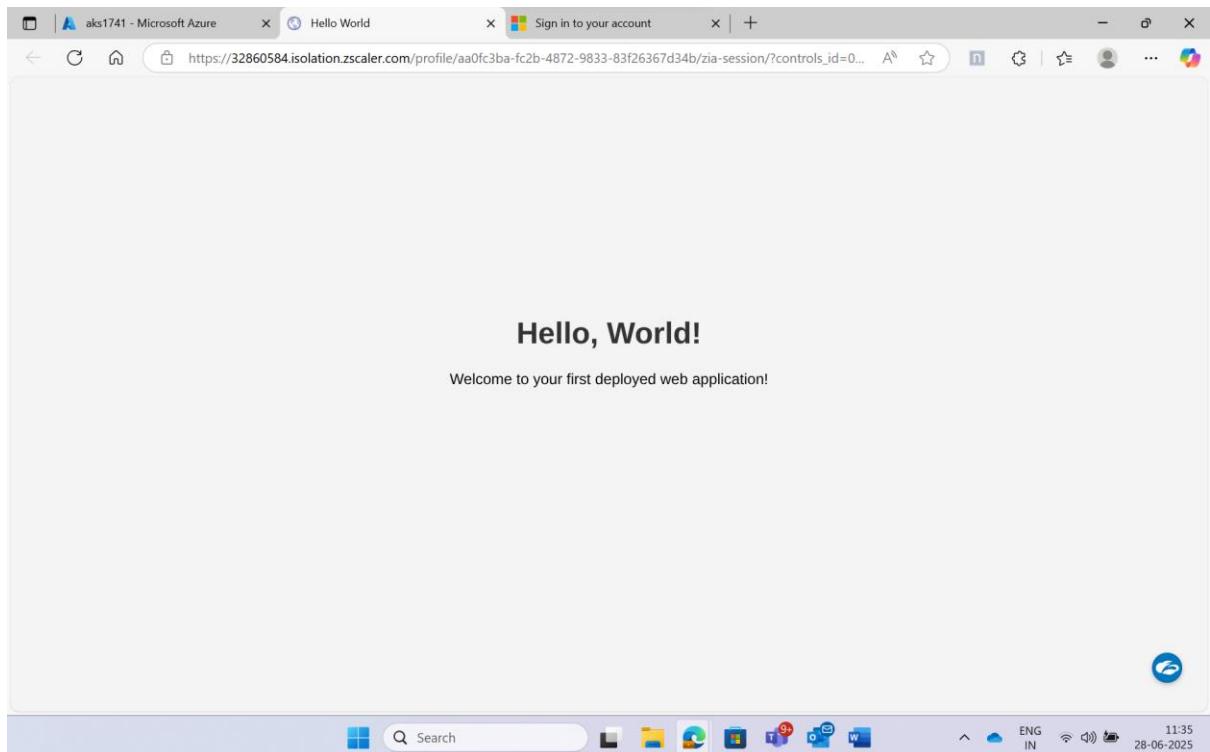
The screenshot shows the Azure portal interface for an AKS cluster named 'aks1741'. The left sidebar navigation bar includes 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Monitor', 'Diagnose and solve problems', 'Microsoft Defender for Cloud (preview)', 'Cost analysis', 'Resource visualizer', 'Kubernetes resources', 'Namespaces', 'Workloads' (which is selected), 'Services and ingresses', and 'Storage'. The main content area displays a table of workloads across all namespaces. The columns include Name, Namespace, Ready, Age, CPU, and Memory. All entries show a green checkmark in the 'Ready' column, indicating they are healthy.

Name	Namespace	Ready	Age	CPU	Memory
coredns	kube-system	2/2	1 day	0%	5%
coredns-autoscaler	kube-system	1/1	1 day	0%	4%
connectivity-agent	kube-system	2/2	1 day	0%	1%
connectivity-agent-autosc	kube-system	1/1	1 day	0%	3%
metrics-server	kube-system	2/2	1 day	0%	12%
eraser-controller-manager	kube-system	1/1	1 day	0%	2%
azure-wi-webhook-contro	kube-system	2/2	1 day	0%	5%
ama-metrics	kube-system	2/2	1 day	0.07%	1%
ama-metrics-ksm	kube-system	1/1	1 day	0%	0.81%
ama-metrics-operator-tar	kube-system	1/1	1 day	0%	0.84%
hello-world-app	default	1/1	1 day	-	-

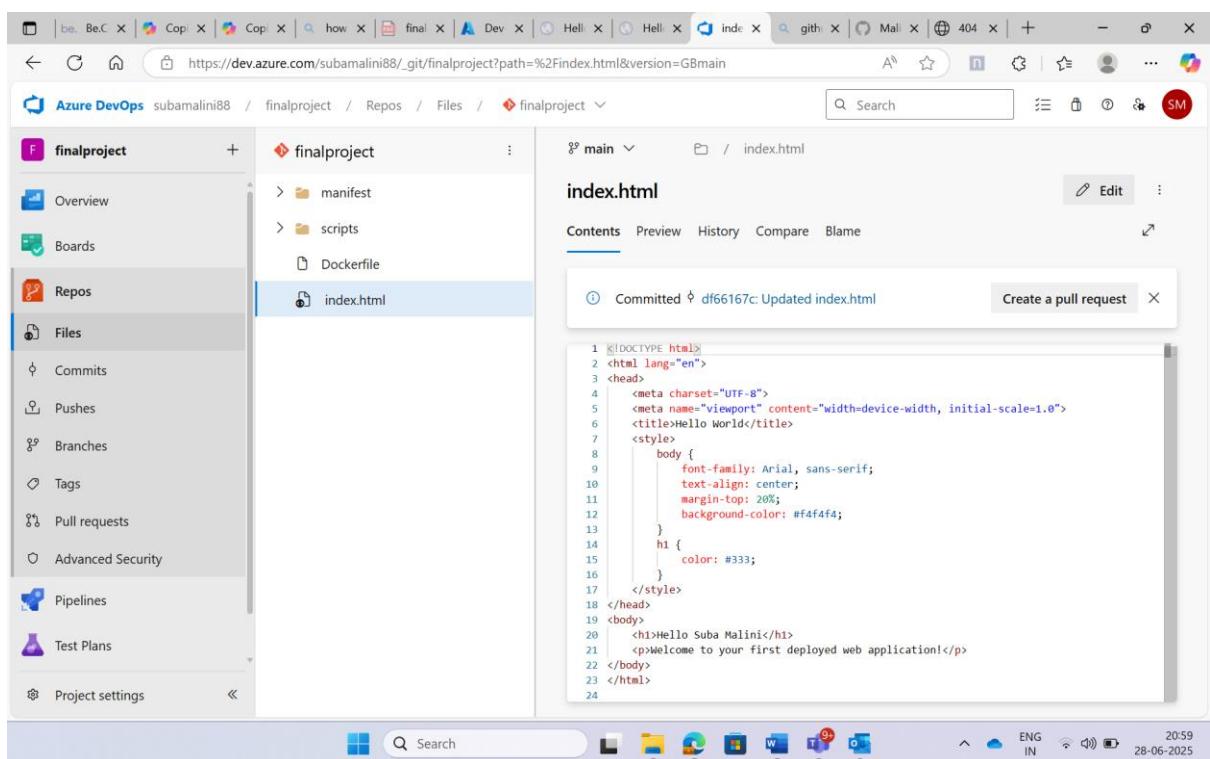
The screenshot shows the Azure portal interface for an AKS cluster named 'aks1741'. The left sidebar navigation bar includes 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', 'Monitor', 'Diagnose and solve problems', 'Microsoft Defender for Cloud (preview)', 'Cost analysis', 'Resource visualizer', 'Kubernetes resources', 'Namespaces', 'Workloads', 'Services and ingresses' (which is selected), and 'Storage'. The main content area displays a table of services across all namespaces. The columns include Name, Namespace, Status, Type, Cluster IP, External IP, and Ports. Most services are marked as 'Ok', except for 'hello-world-service' which is listed as 'LoadBalancer'.

Name	Namespace	Status	Type	Cluster IP	External IP	Ports
kubernetes	default	Ok	ClusterIP	10.0.0.1		443/TCP
kube-dns	kube-system	Ok	ClusterIP	10.0.0.10		53/UDP,53/TCP
metrics-server	kube-system	Ok	ClusterIP	10.0.48.131		443/TCP
azure-wi-webhook-web...	kube-system	Ok	ClusterIP	10.0.126.155		443/TCP
ama-metrics-ksm	kube-system	Ok	ClusterIP	10.0.67.169		8080/TCP
ama-metrics-operator-t...	kube-system	Ok	ClusterIP	10.0.211.180		80/TCP
network-observability	kube-system	Ok	ClusterIP	10.0.97.140		10093/TCP
hello-world-service	default	Ok	LoadBalancer	10.0.193.31	4.156.41.162	80:32042/TCP

The page gets executed



By making changes in the index.html :



We can visualise the changes now:

