

CODTECH INTERNSHIP – Cloud Computing

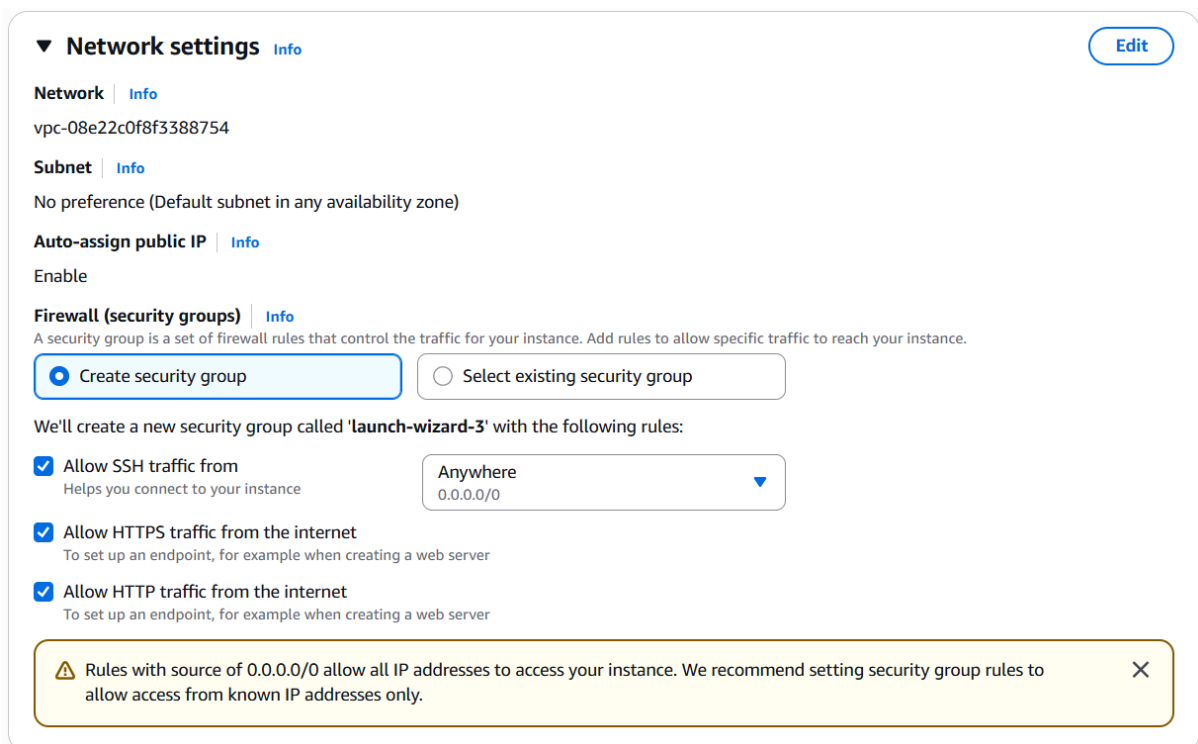
Task-3 : Multi-Cloud Architecture

DESIGN A MULTI-CLOUD ARCHITECTURE WHERE SERVICES ARE DISTRIBUTED ACROSS TWO CLOUD PROVIDERS

DELIVERABLE: A DOCUMENTATION AND DEMO SHOWCASING INTEROPERABILITY BETWEEN THE PLATFORMS

Steps to Complete Task:

Create an Instance



The screenshot shows the 'Network settings' section of an AWS console. It includes fields for 'Network' (vpc-08e22c0f8f3388754) and 'Subnet' (No preference). Under 'Auto-assign public IP', 'Enable' is selected. In the 'Firewall (security groups)' section, 'Create security group' is selected, and a new group named 'launch-wizard-3' is being created with three rules: 'Allow SSH traffic from Anywhere', 'Allow HTTPS traffic from the internet', and 'Allow HTTP traffic from the internet'. A warning message at the bottom states: 'Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.'

▼ Network settings [Info](#) [Edit](#)

Network | [Info](#)
vpc-08e22c0f8f3388754

Subnet | [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP | [Info](#)
Enable

Firewall (security groups) | [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

We'll create a new security group called 'launch-wizard-3' with the following rules:

- ☒ Allow SSH traffic from
Helps you connect to your instance Anywhere
0.0.0.0/0
- ☒ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
- ☒ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ✕

While Creating Instance, Allow all network traffic.

Connect [info](#)

Connect to an instance using the browser-based client.

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID
i-046383d0125352517 (Multi-Instance)

☒ Connect using a Public IP
Connect using a public IPv4 or IPv6 address

☐ Connect using a Private IP
Connect using a private IP address and a VPC endpoint

☒ Public IPv4 address
54.160.173.224

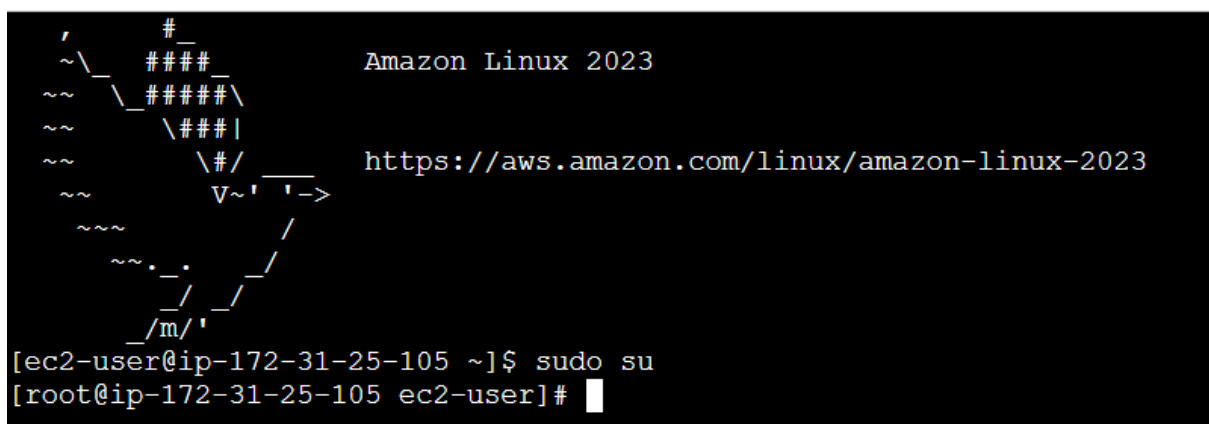
☐ IPv6 address
-

Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, ec2-user.

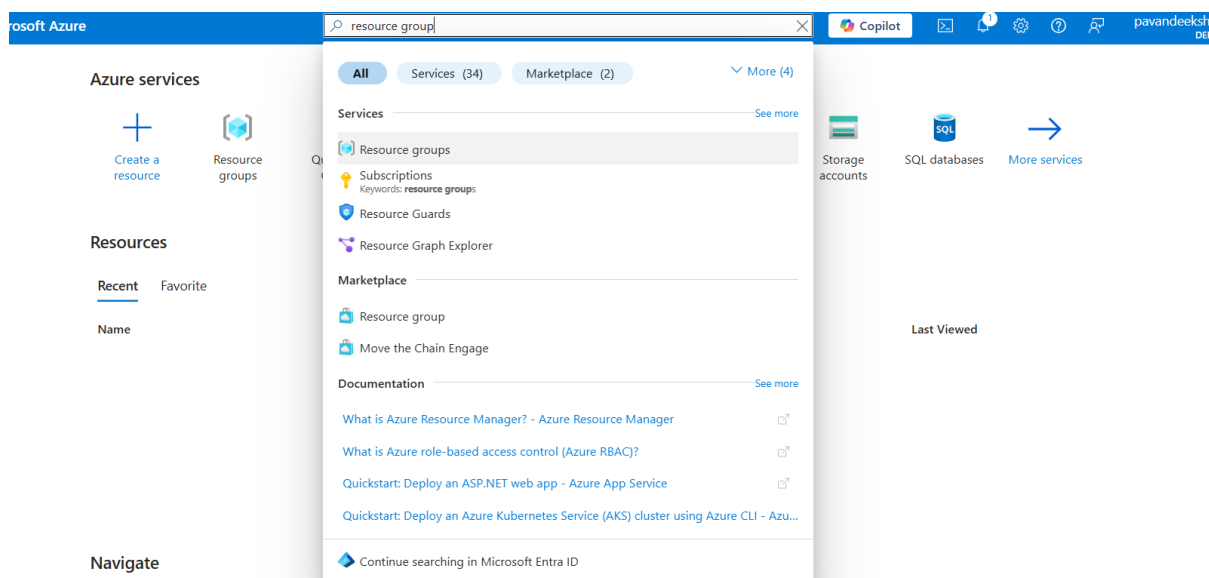
ec2-user

Note: In most cases, the default username, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Select the Instance and Click on Connect.



Connect to Linux as admin using “sudo su”



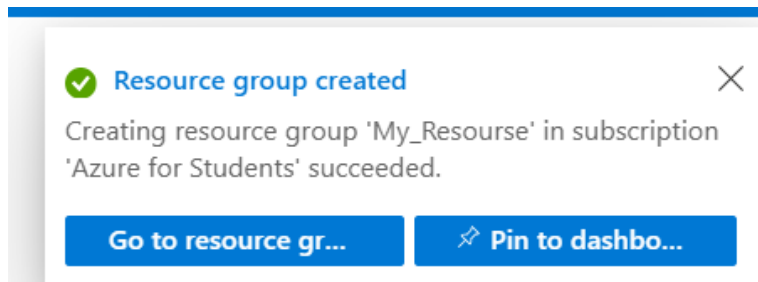
Sign in to Azure Portal and go to Home page

Search for Resource groups in Search bar and Click on it.

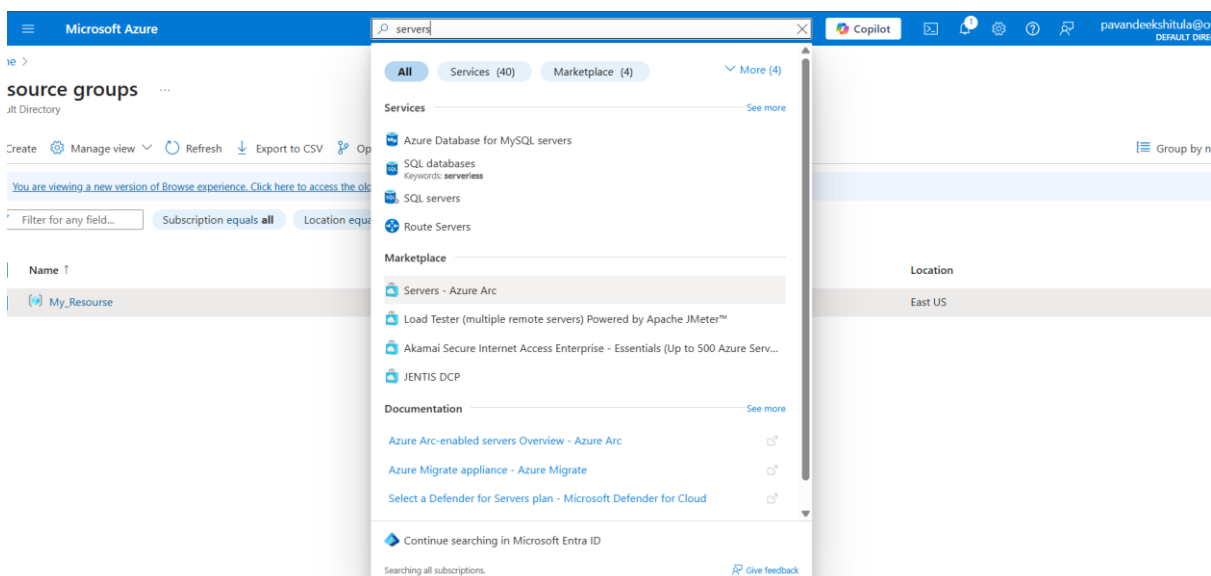
Give the Name and Regions for it.

- Name : My_Resource
- Regions : (US) East US

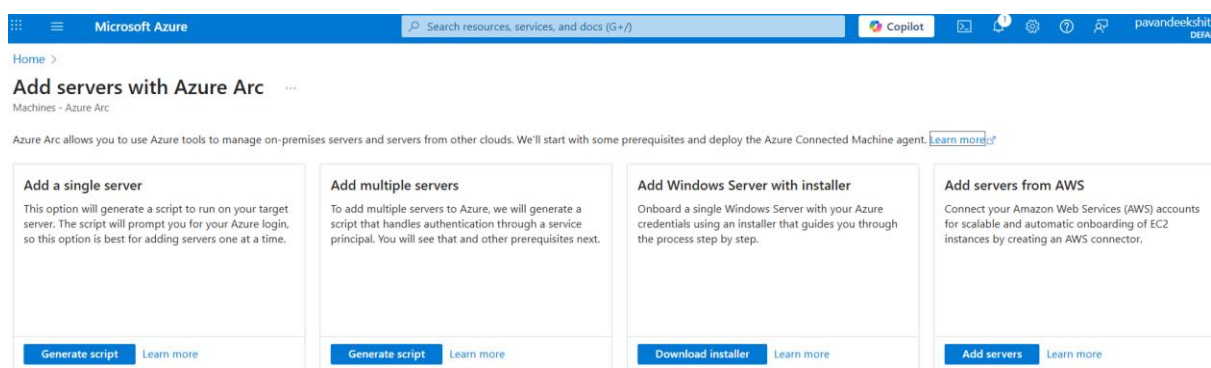
Click on Create



Group is Created



Go to Servers → Azure Arc → Click it



Go to add a Single Server → Click on Generate script.

The screenshot shows the Microsoft Azure portal interface for adding a server with Azure Arc. The top navigation bar includes the Microsoft Azure logo and a search bar. The breadcrumb trail is 'Home > Add servers with Azure Arc >'. The main heading is 'Add a server with Azure Arc'. Below the heading are tabs for 'Basics', 'Tags', and 'Download and run script'. A message states: 'Complete the fields below to connect servers on-premise and in other clouds to be managed and governed in Azure. [Learn more](#)'. The 'Project details' section instructs to 'Select the subscription and resource group where you want the server to be managed within Azure.' It contains two dropdown menus: 'Subscription * ⓘ' with 'Azure for Students' selected, and 'Resource group * ⓘ' with 'My_Resource' selected. A 'Create new' link is present below the resource group dropdown. The 'Server details' section instructs to 'Select details for the servers that you want to add. An agent package will be generated for the selected server type.' It contains two dropdown menus: 'Region * ⓘ' with '(US) East US' selected, and 'Operating system * ⓘ' with 'Linux' selected.

Give the Subscription and Resource group in Project Details.

Give the Region and OS in Server Details.

Select the Connectivity as Public endpoint.

Keep the Physical location tags as Default.

Go to Add servers with Azure Arc

[Home](#) > [Add servers with Azure Arc](#) >

Add a server with Azure Arc ...

```
1
2 export subscriptionId="d40032a9-b8a0-48ab-8cb3-bb6c8ac3dea5";
3 export resourceGroup="My_Resource";
4 export tenantId="bd38eb72-bb42-4b92-8bab-d31c38871ad8";
5 export location="eastus";
6 export authType="token";
7 export correlationId="a54210b4-3b9d-4a2c-8a00-0feb3e164a19";
8 export cloud="AzureCloud";
9
10
11 # Download the installation package
12 LINUX_INSTALL_SCRIPT="/tmp/install_linux_azcmagent.sh"
13 if [ -f "$LINUX_INSTALL_SCRIPT" ]; then rm -f "$LINUX_INSTALL_SCRIPT"; fi;
14 output=$(wget https://gbl.his.arc.azure.com/azcmagent-linux -O "$LINUX_INSTALL_SCRIPT" 2>&1);
15 if [ $? != 0 ]; then wget -qO- --method=PUT --body-data="{\"subscriptionId\": \"$subscriptionId\", \"resourceGroup\": \"$resourceGroup\", \"tenantId\": \"$tenantId\", \"location\": \"$location\", \"authType\": \"$authType\", \"operation\": \"onboarding\", \"messageType\": \"DownloadScriptFailed\", \"message\": \"$output\"}" https://gbl.his.arc.azure.com/log" &> /dev/null || true; fi;
16 echo "$output";
17
18 # Install the hybrid agent
19 bash "$LINUX_INSTALL_SCRIPT";
20 sleep 5;
21
22 # Run connect command
23 sudo azcmagent connect --resource-group "$resourceGroup" --tenant-id "$tenantId" --location "$location" --subscription-id "$subscriptionId" --cloud "$cloud" --correlation-id "$correlationId";
24
```

Copy this code → go to aws Linux console

Type the code in Linux console.

```
vi azure.sh
```

```
export subscriptionId="d40032a9-b8a0-48ab-8cb3-bb6c8ac3dea5";
export resourceGroup="My_Resource";
export tenantId="bd38eb72-bb42-4b92-8bab-d31c38871ad8";
export location="eastus";
export authType="token";
export correlationId="a54210b4-3b9d-4a2c-8a00-0feb3e164a19";
export cloud="AzureCloud";
LINUX_INSTALL_SCRIPT="/tmp/install_linux_azcmagent.sh"
if [ -f "$LINUX_INSTALL_SCRIPT" ]; then rm -f "$LINUX_INSTALL_SCRIPT"; fi;
output=$(wget https://gbl.his.arc.azure.com/azcmagent-linux -O "$LINUX_INSTALL_SCRIPT" 2>&1);
if [ $? != 0 ]; then wget -qO- --method=PUT --body-data="{\"subscriptionId\": \"$subscriptionId\", \"resourceGroup\": \"$resourceGroup\", \"tenantId\": \"$tenantId\", \"location\": \"$location\", \"correlationId\": \"$correlationId\", \"authType\": \"$authType\", \"operation\": \"onboarding\", \"messageType\": \"DownloadScriptFailed\", \"message\": \"$output\"}" https://gbl.his.arc.azure.com/log" &> /dev/null || true; fi;
echo "$output";
bash "$LINUX_INSTALL_SCRIPT";
sleep 5;
sudo azcmagent connect --resource-group "$resourceGroup" --tenant-id "$tenantId" --location "$location" --subscription-id "$subscriptionId" --cloud "$cloud" --correlation-id "$correlationId";
```

Paste the Code here and save it

After saving, it redirects to Linux console.

After returning to console, type this code

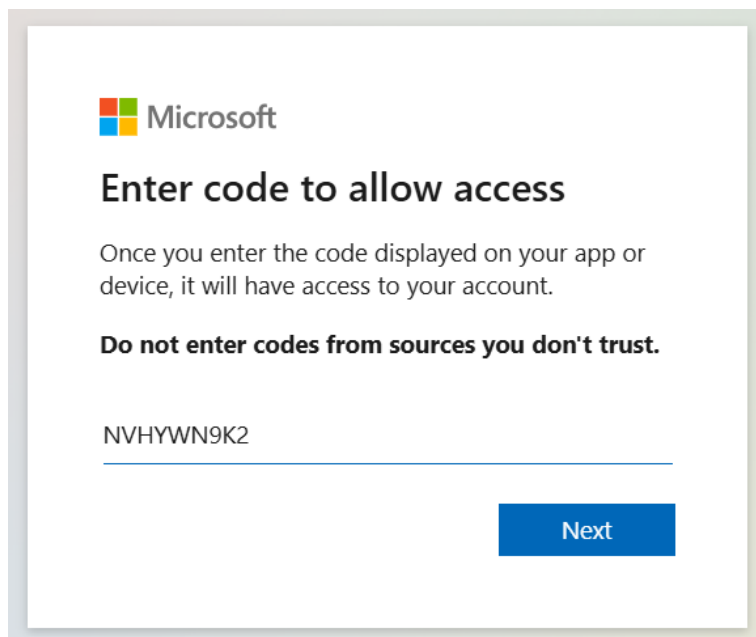
```
chmod +x azure.sh
```

```
ls
```

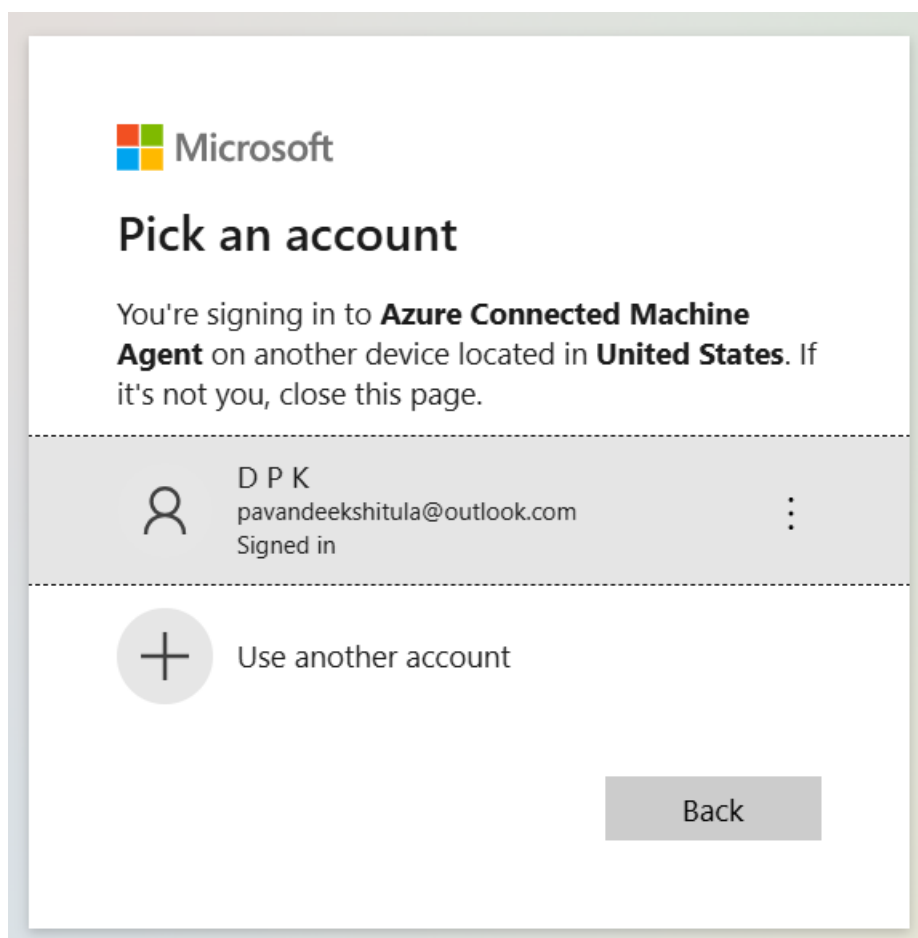
```
sh azure.sh
```

After executing the code we will get details and get https browser link.

Copy the https browser link and paste it in the browser tab.



Enter the codes from details you got.




Select the Account to Sign in.



Now the Azure Machine is connected.

After that close this → Go to home → Resource Group → Select My_Resource (Group you created).

^ Essentials

Resource group [\(move\)](#) : [My_Resource](#) 
Status : Connected
Location [\(move\)](#) : East US
Subscription [\(move\)](#) : [Azure for Students](#)
Subscription ID : d40032a9-b8a0-48ab-8cb3-bb6c8ac3dea5
Agent version : 1.54.03104.480

Computer name : ip-172-31-25-105.ec2.internal
FQDN : ip-172-31-25-105.ec2.internal
Operating system : Amazon Linux 2023.8.20250707
Operating system version : 6.1.141-165.249.amzn2023.x86_64
Cloud provider : AWS
Manufacturer : Amazon EC2
Model : t3.micro
Arc gateway : [Add Arc gateway](#)

Instance summary for i-046383d0125352517 (Multi-Instance)

Info

Connect

Instance state

Actions

Updated 1 minute ago

<div>Instance ID</div> <div>i-046383d0125352517</div>	<div>Public IPv4 address</div> <div>54.160.173.224 open address</div>	<div>Private IPv4 addresses</div> <div>172.31.25.105</div>
<div>IPv6 address</div> <div>-</div>	<div>Instance state</div> <div>Running</div>	<div>Public DNS</div> <div>ec2-54-160-173-224.compute-1.amazonaws.com open address</div>
<div>Hostname type</div> <div>IP name: ip-172-31-25-105.ec2.internal</div>	<div>Private IP DNS name (IPv4 only)</div> <div>ip-172-31-25-105.ec2.internal</div>	
<div>Answer private resource DNS name</div> <div>IPv4 (A)</div>	<div>Instance type</div> <div>t3.micro</div>	<div>Elastic IP addresses</div> <div>-</div>
<div>Auto-assigned IP address</div> <div>54.160.173.224 [Public IP]</div>	<div>VPC ID</div> <div>vpc-08e22c0f8f3388754</div>	<div>AWS Compute Optimizer finding</div> <div> Opt-in to AWS Compute Optimizer for recommendation s. Learn more </div>
<div>IAM Role</div> <div>-</div>	<div>Subnet ID</div> <div>subnet-03e0a3db4a7ba8801</div>	<div>Auto Scaling Group name</div> <div>-</div>
<div>IMDSv2</div> <div>Required</div>	<div>Instance ARN</div> <div>arn:aws:ec2:us-east-1:233720366258:instance/i-046383d0125352517</div>	<div>Managed</div> <div>false</div>

Now we can see that the computer name will be having same IP Address in Azure and Private IPv4 address in AWS.

Here, This is the Process to connect two cloud Providers.