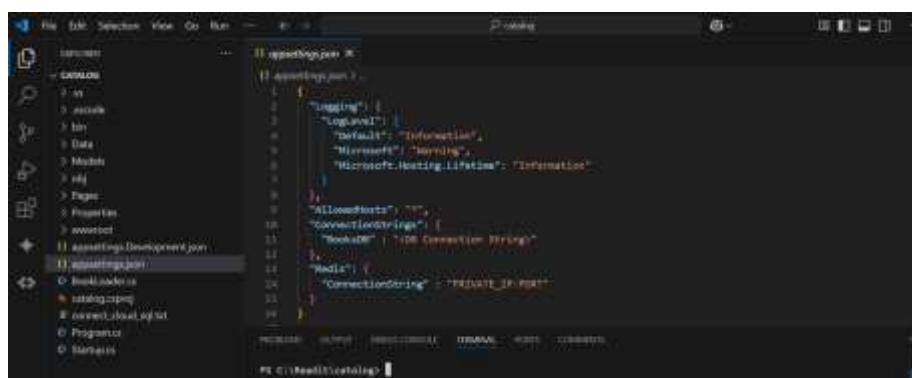
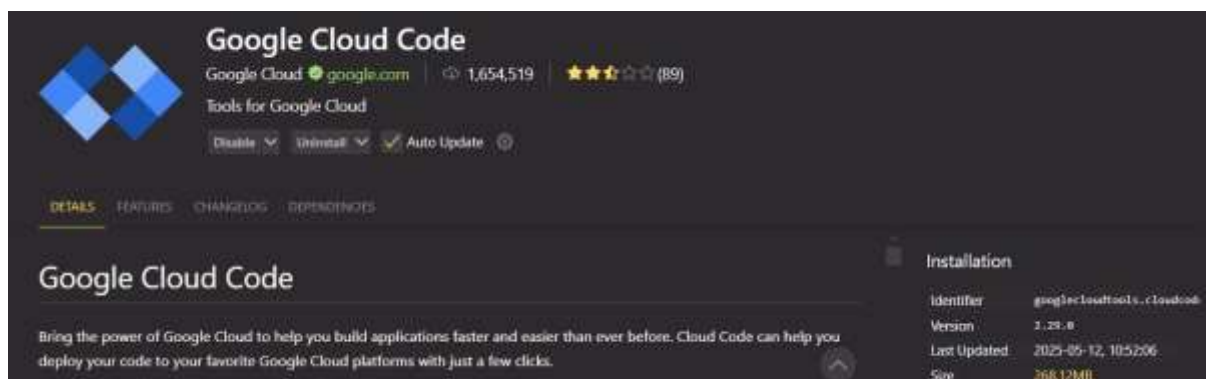
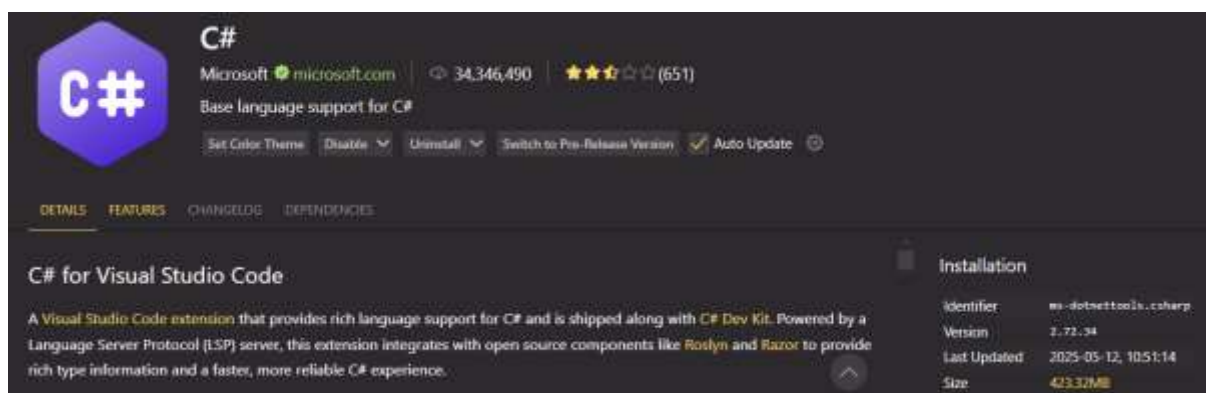
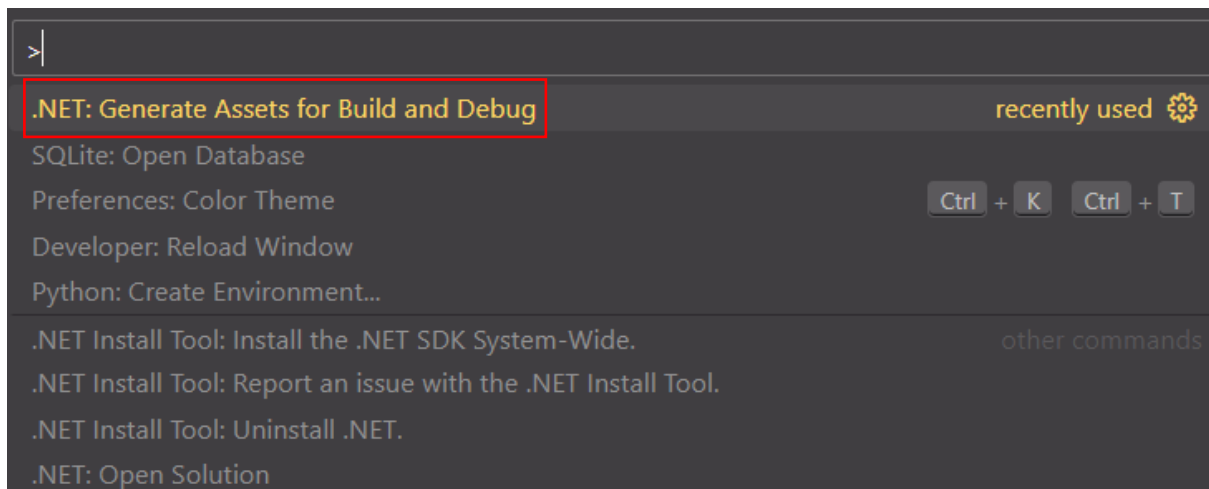


In this Lab, we will publish our ReadIt app to the cat1 VM. We will also see options to log in to a VM using PuTTY and SSH key.

- 1) Start the cat1 VM we created earlier, the VM metadata lab on the GCP console, or if you have deleted the VM, then just create a new one based on the same configurations.
- 2) Open VS Code and open the catalog folder there. (NOTE: You will find a zip file for the catalog folder on GitHub, download it and unzip it)
- 3) After unzipping the catalog folder, you need to open it in VS Code, then inside the code editor, you need to install two extensions. First, it is for C#, and the other is for Google cloud code.



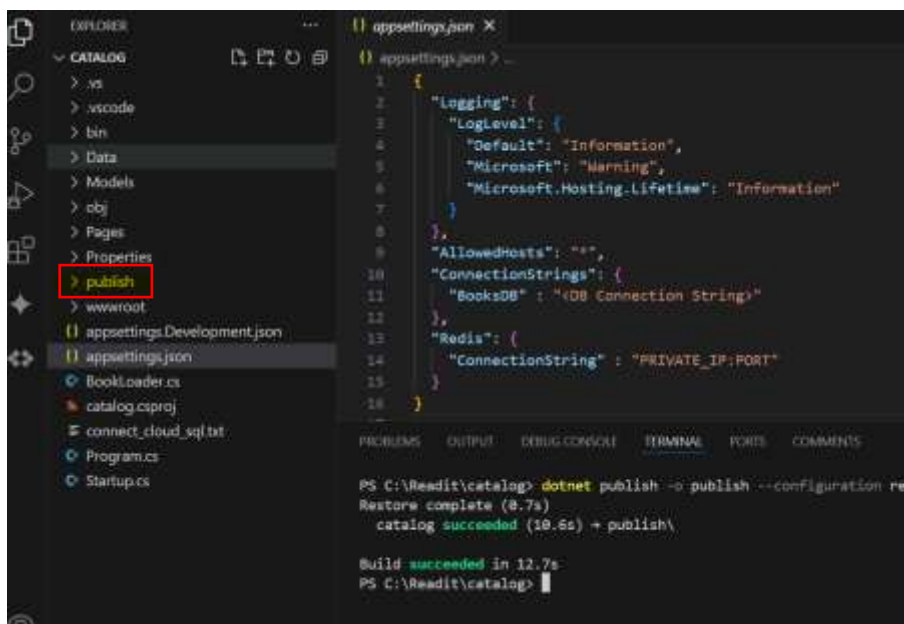
- 4) Now in VS Code, you need to open the command palette then choose this option for .NET.



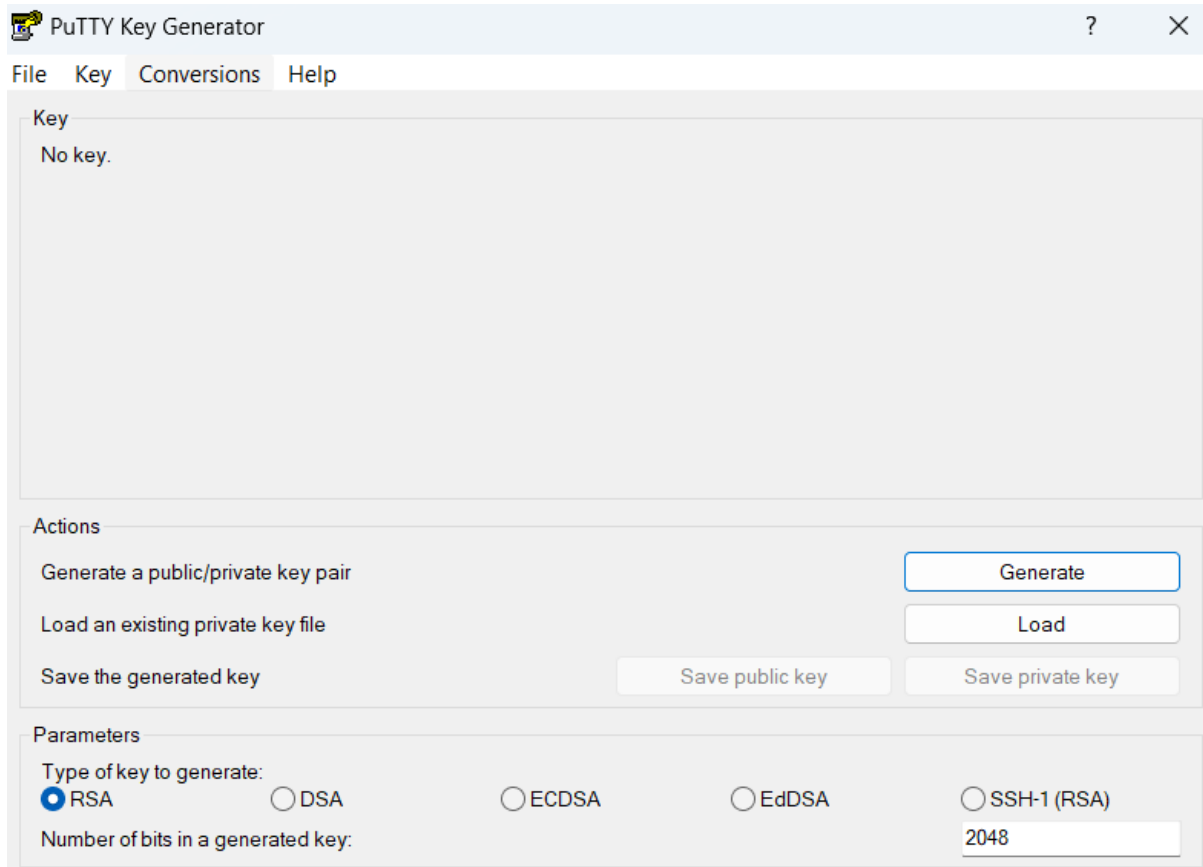
- 5) Go to Terminal>New terminal and run the below command

dotnet publish -o publish --configuration release

Once completed, you must be able to see the new folder “Publish”



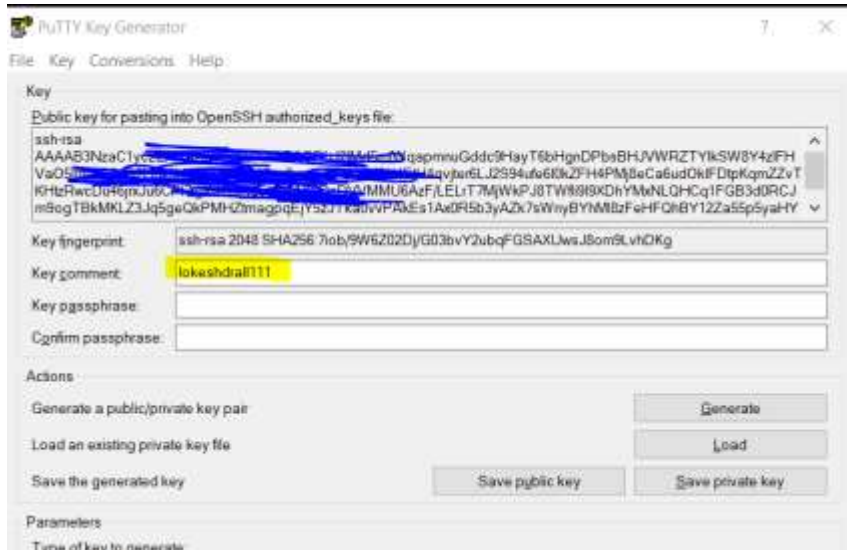
- 6) Download and install the PuTTY tool <https://www.putty.org/>
- 7) Open Puttygen (you can also download it from putty.org) and click on generate to create a key, and keep the mouse moving until it completes generating the key



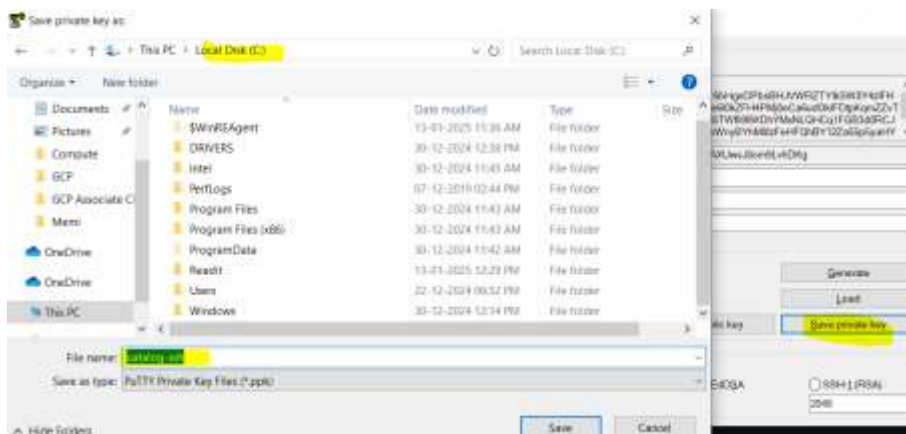
8) Now, go to the GCP console and copy your username (without email suffix)



9) Go to PuttyGen, replace "key comment" with your username



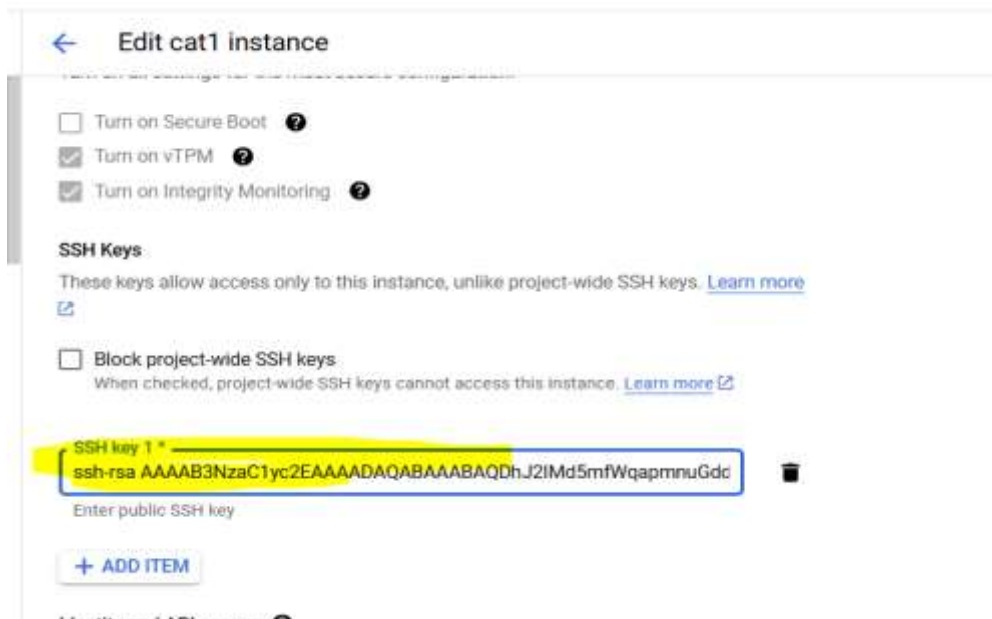
10) Click Save private key and save it to the C drive or any drive of your choice with the name “catalog-ssh” (If you see any errors saving it to C drive, then first save it somewhere else and then copy it to C drive)



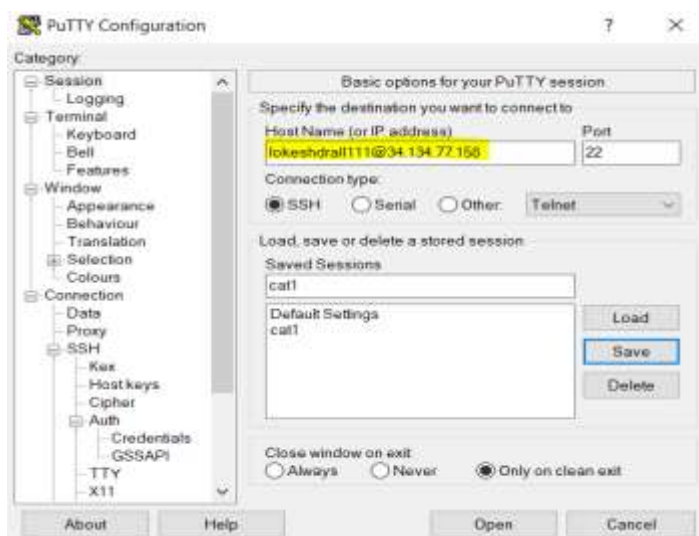
11) Copy the public key from Puttygen



12) Go to the GCP console, open the cat1 VM, click edit, scroll down to SSH Keys, click add item, and paste the key



- 13) Connect to cat1 VM via putty, open putty, enter your username@externalip as hostname, add the private key location under **SSH> Auth> Credentials**, also optionally can save it for future use, and click open



- 14) Run the below commands one by one on the cat1 VM (Also, you will get a text file for the same commands, you can use any of the options):-

- wget <https://packages.microsoft.com/config/debian/11/packages-microsoft-prod.deb> -O packages-microsoft-prod.deb**
- sudo dpkg -i packages-microsoft-prod.deb**
- sudo dpkg -i packages-microsoft-prod.deb**
- sudo apt-get update && **
sudo apt-get install -y aspnetcore-runtime-8.0

```

lokeshdrall111@cat1:~$ wget https://packages.microsoft.com/config/debian/11/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
--2025-01-13 08:07:41-- https://packages.microsoft.com/config/debian/11/packages-microsoft-prod.deb
Resolving packages.microsoft.com [packages.microsoft.com]... 13.107.246.38, 2620:1ecbdf::35
Connecting to packages.microsoft.com [packages.microsoft.com]13.107.246.38:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3134 (3.1K) [application/octet-stream]
Saving to: 'packages-microsoft-prod.deb'

packages-microsoft-prod.deb 100%[=====>] 3.05K --.-KB/s in 0s

2025-01-13 08:07:41 (35.0 MB/s) - 'packages-microsoft-prod.deb' saved [3134/3134]

lokeshdrall111@cat1:~$ sudo dpkg --i packages-microsoft-prod.deb
Selecting previously unselected package packages-microsoft-prod.
(Reading database ... 63033 files and directories currently installed.)
Preparing to unpack packages-microsoft-prod.deb ...
Unpacking packages-microsoft-prod (1.0-debian11.1) ...
Setting up packages-microsoft-prod (1.0-debian11.1) ...
lokeshdrall111@cat1:~$ rm packages-microsoft-prod.deb
lokeshdrall111@cat1:~$

```

```

Unpacking packages-microsoft-prod (1.0-debian11.1) ...
Setting up packages-microsoft-prod (1.0-debian11.1) ...
lokeshdrall111@cat1:~$ rm packages-microsoft-prod.deb
lokeshdrall111@cat1:~$
lokeshdrall111@cat1:~$
lokeshdrall111@cat1:~$ sudo apt-get update && \
sudo apt-get install -y aspnetcore-runtime-6.0

```

15) To ensure dotnet is installed run below, it should give dotnet details

dotnet --info

```

lokeshdrall111@cat1:~$ dotnet --info
global.json file:
  Not found

Host:
  Version: 6.0.36
  Architecture: x64
  Commit: f1dd57165b

.NET SDKs installed:
  No SDKs were found.

.NET runtimes installed:
  Microsoft.AspNetCore.App 6.0.36 [/usr/share/dotnet/shared/Microsoft.AspNetCore.App]
  Microsoft.NETCore.App 6.0.36 [/usr/share/dotnet/shared/Microsoft.NETCore.App]

Download .NET:
  https://aka.ms/dotnet-download

Learn about .NET Runtimes and SDKs:
  https://aka.ms/dotnet/runtimes-sdk-info
lokeshdrall111@cat1:~$

```

16) Run below commands so that we can create a folder in our VM, and in this folder, we will upload the data from the publish folder.

- a. **mkdir catalog**
- b. **cd catalog**
- c. **pwd**

17) Copy the whole path “/home/lokeshdrall111/catalog”

```

lokeshdrall111@cat1:~$ mkdir catalog
lokeshdrall111@cat1:~$ cd catalog
lokeshdrall111@cat1:~/catalog$ pwd
/home/lokeshdrall111/catalog
lokeshdrall111@cat1:~/catalog$
lokeshdrall111@cat1:~/catalog$
lokeshdrall111@cat1:~/catalog$

```

C. DV

18) Go to VS Code, go to the publish folder by typing **cd publish**


```

PS C:\Readit\catalog> dotnet publish -o publish --configuration release
Restore complete (0.7s)
catalog succeeded (10.6s) → publish\

Build succeeded in 12.7s
PS C:\Readit\catalog> cd publish
PS C:\Readit\catalog\publish>

```

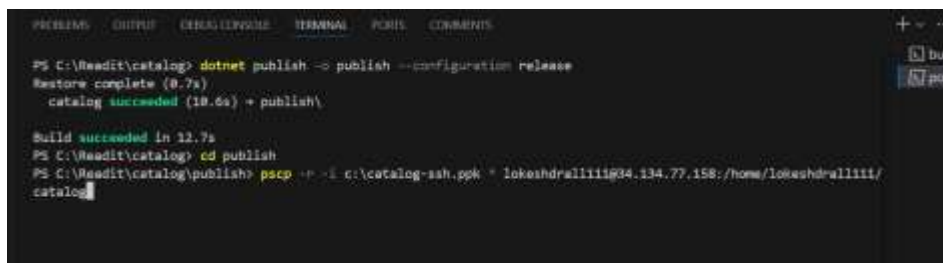
19) Now to upload our code files to VM, run the below command in VS (Note: if you see an error like pscp not recognized... then simply close and open VS Code again)

Note: The PSCP command is used for transferring files securely between a local Windows computer and a remote host using the PuTTY Secure Copy Protocol (SCP).

```

pscp -r -i c:\catalog-ssh.ppk *
{USERNAME}@{IP_ADDRESS}:/home/{USERNAME}/catalog

```



```

PREVIEWS OUTPUT DETAILEDVIEW TERMINAL FILES COMMENTS
PS C:\Readit\catalog> dotnet publish -o publish --configuration release
Restore complete (0.7s)
catalog succeeded (10.6s) → publish\

Build succeeded in 12.7s
PS C:\Readit\catalog> cd publish
PS C:\Readit\catalog\publish> pscp -r -i c:\catalog-ssh.ppk * lokeshdrall111@34.134.77.158:/home/lokeshdrall111/catalog

```

20) Run below command in VM (putty)

export ASPNETCORE_ENVIRONMENT=Production

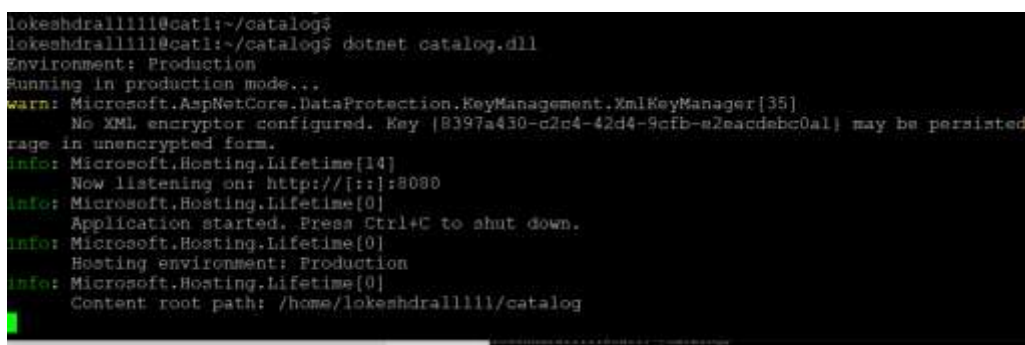


```

lokeshdrall111@cat1:~/catalog$
lokeshdrall111@cat1:~/catalog$ export ASPNETCORE_ENVIRONMENT=Production
lokeshdrall111@cat1:~/catalog$

```

21) Run dotnet **catalog.dll** it should show app running

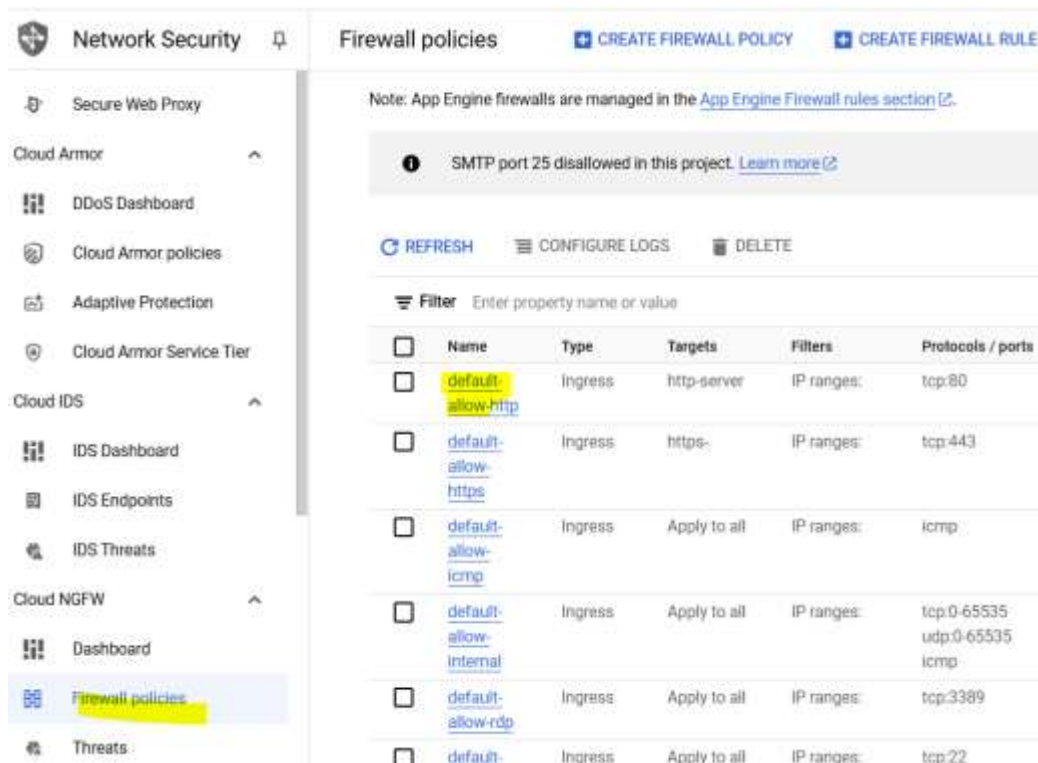


```

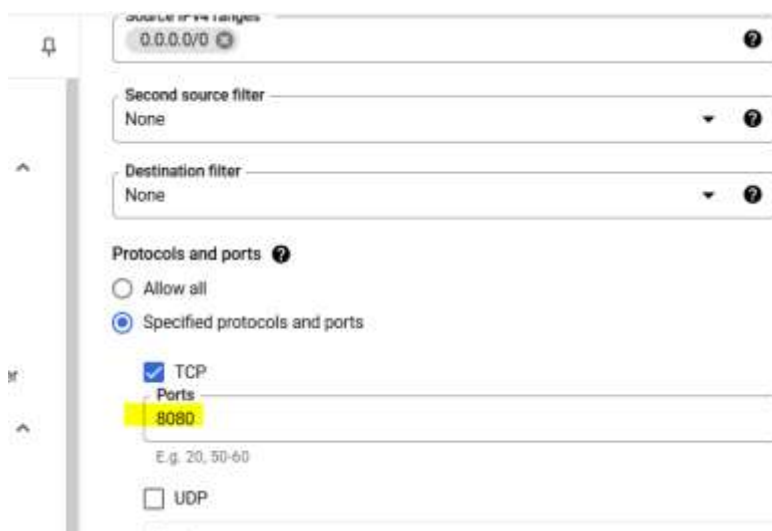
lokeshdrall111@cat1:~/catalog$
lokeshdrall111@cat1:~/catalog$ dotnet catalog.dll
Environment: Production
Running in production mode...
warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
      No XML encryptor configured. Key [8397a430-c2c4-42d4-9cfb-e2eacdebc0a1] may be persisted
      in unencrypted form.
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://[::]:8080
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /home/lokeshdrall111/catalog

```

22) Now we need to open port 8080 at GCP firewall to access this via web, for that go to GCP, search VPC network and open default-allow-http rule.

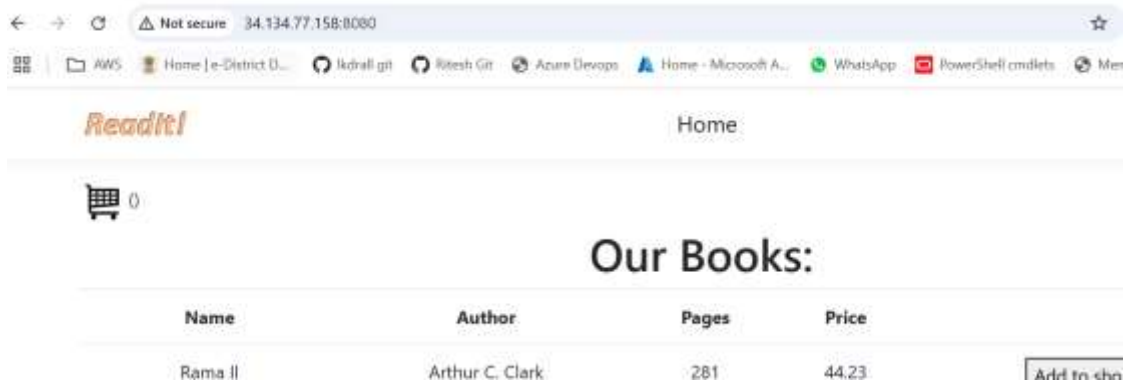


23) Click edit, and change the port number to 8080 (initially it was 80), then click save

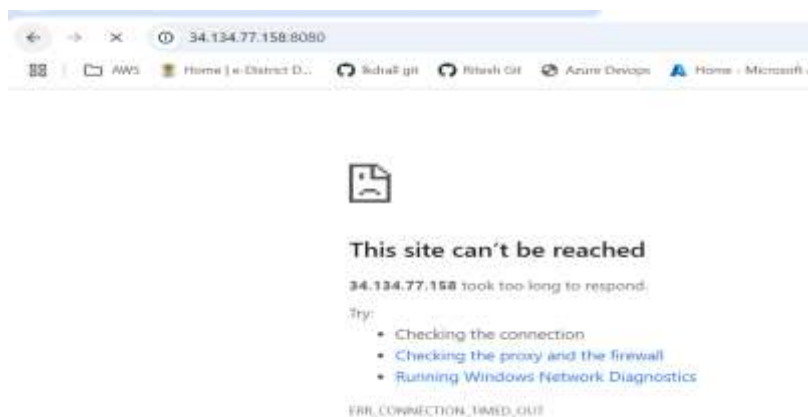


24) Copy the external IP of the VM, paste it in a new tab of the browser, and add port 8080, 34.134.77.158:8080

And open it, this should show the readit app.



25) Now if we close the putty screen, and then reload page then it will not work



26) Now go back to the GCP console, open VM and click edit, scroll down to Automation and paste the below script there, make sure to update the username, click save.

```
#!/bin/bash
```

```
export ASPNETCORE_ENVIRONMENT=Production
```

```
cd /home/{USERNAME}/catalog
```

```
dotnet catalog.dll
```

gine [←](#) Edit cat1 instance

This is useful for passing in arbitrary values to your project or instance that can be queried by your code on the instance. [Learn more](#)

Key 1 *	Value 1
env	test

[+ ADD ITEM](#)

Automation

Startup script

```
#!/bin/bash
export ASPNETCORE_ENVIRONMENT=Production
cd /home/lokeshdrall111/catalog
dotnet catalog.dll
```

You can choose to specify a startup script that will run when your instance boots up or restarts. Startup scripts can be used to install software and updates, and to ensure that services are running within the virtual machine. [Learn more](#)

27) Now stop and start the VM after starting, try accessing web app again, it should work now. (After restart external Ip may change, so make sure to check that)

→ [Not secure](#) 34.134.70.142:8080

AWS Home | e-District D... lkdrall git Ritesh Git Azure Devops Home - Microsoft A... WhatsApp

ReadIt! Home

 0

Our Books:

Name	Author	Pages
Load Books to DB		