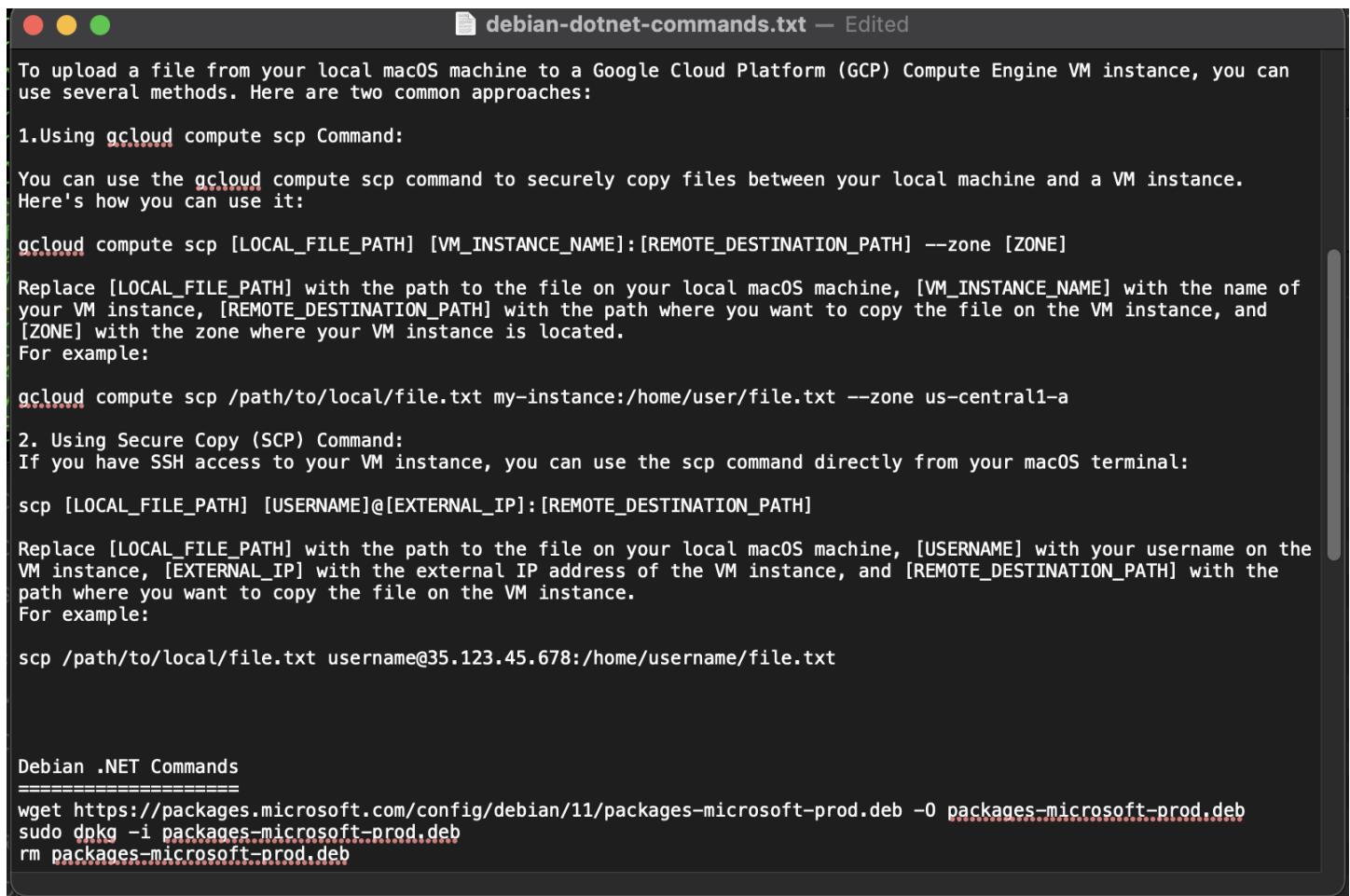


## 51. Setup catalog VM, 52. Publish the catalog app

This was given by lecturer, and i have mac so no need putty.



```
To upload a file from your local macOS machine to a Google Cloud Platform (GCP) Compute Engine VM instance, you can use several methods. Here are two common approaches:

1. Using gcloud compute scp Command:
You can use the gcloud compute scp command to securely copy files between your local machine and a VM instance. Here's how you can use it:
gcloud compute scp [LOCAL_FILE_PATH] [VM_INSTANCE_NAME]:[REMOTE_DESTINATION_PATH] --zone [ZONE]
Replace [LOCAL_FILE_PATH] with the path to the file on your local macOS machine, [VM_INSTANCE_NAME] with the name of your VM instance, [REMOTE_DESTINATION_PATH] with the path where you want to copy the file on the VM instance, and [ZONE] with the zone where your VM instance is located.
For example:
gcloud compute scp /path/to/local/file.txt my-instance:/home/user/file.txt --zone us-central1-a

2. Using Secure Copy (SCP) Command:
If you have SSH access to your VM instance, you can use the scp command directly from your macOS terminal:
scp [LOCAL_FILE_PATH] [USERNAME]@[EXTERNAL_IP]:[REMOTE_DESTINATION_PATH]
Replace [LOCAL_FILE_PATH] with the path to the file on your local macOS machine, [USERNAME] with your username on the VM instance, [EXTERNAL_IP] with the external IP address of the VM instance, and [REMOTE_DESTINATION_PATH] with the path where you want to copy the file on the VM instance.
For example:
scp /path/to/local/file.txt username@35.123.45.678:/home/username/file.txt

Debian .NET Commands
=====
wget https://packages.microsoft.com/config/debian/11/packages-microsoft-prod.deb -O packages-microsoft-prod.deb
sudo dpkg -i packages-microsoft-prod.deb
rm packages-microsoft-prod.deb
```

Follow rest of commands above txt file but ignore putty part -



you

is putty required ?



ChatGPT

No, PuTTY is not required for uploading files from your local macOS machine to a Google Cloud Platform (GCP) Compute Engine VM instance. PuTTY is primarily used for SSH connectivity and remote command-line access to Windows-based servers. However, since you're connecting from a macOS machine, you can use the built-in Terminal application to execute SSH commands and securely transfer files to your VM instance using SSH protocols like SCP (Secure Copy Protocol).

So, you can use either the `gcloud compute scp` command or the `scp` command directly from your macOS Terminal to upload files to your VM instance without the need for PuTTY.

how to upload file from local mac area to gcloud instance vm

 ChatGPT

To upload a file from your local macOS machine to a Google Cloud Platform (GCP) Compute Engine VM instance, you can use several methods. Here are two common approaches:

**1. Using `gcloud compute scp` Command:**

You can use the `gcloud compute scp` command to securely copy files between your local machine and a VM instance. Here's how you can use it:

less

 Copy code

```
gcloud compute scp [LOCAL_FILE_PATH] [VM_INSTANCE_NAME]:[REMOTE_DESTINATION_PATH]
```

Replace `[LOCAL_FILE_PATH]` with the path to the file on your local macOS machine, `[VM_INSTANCE_NAME]` with the name of your VM instance, `[REMOTE_DESTINATION_PATH]` with the path where you want to copy the file on the VM instance, and `[ZONE]` with the zone where your VM instance is located. For example:

lua

 Copy code

```
scp /path/to/local/file.txt my-instance:/home/user/file.txt --zone us-central1-a
```

**2. Using Secure Copy (SCP) Command:**

If you have SSH access to your VM instance, you can use the `scp` command directly from your macOS terminal:

less

 Copy code

```
scp [LOCAL_FILE_PATH] [USERNAME]@[EXTERNAL_IP]:[REMOTE_DESTINATION_PATH]
```

Replace `[LOCAL_FILE_PATH]` with the path to the file on your local macOS machine, `[USERNAME]` with your username on the VM instance, `[EXTERNAL_IP]` with the external IP address of the VM instance, and `[REMOTE_DESTINATION_PATH]` with the path where you want to copy the file on the VM instance. For example:

ruby

 Copy code

```
scp /path/to/local/file.txt username@35.123.45.678:/home/username/file.txt
```

Ensure that you have proper permissions and network connectivity to perform these operations.

↓ Additionally, replace placeholders with actual values specific to your environment.

PR

You

```
scp [LOCAL_FILE_PATH] [USERNAME]@[EXTERNAL_IP]:[REMOTE_DESTINATION_PATH]
```

my gcp details are -

username > rohankodavalla

external ip > 35.203.96.189

remote destination path > /home/rohankodavalla/catalog

local file path > (base) rohankodavalla@Rohans-Air publish %

ChatGPT

If your local file path includes spaces or special characters, you'll need to enclose it in quotes to ensure that it's interpreted correctly. Here's how you can use the `scp` command to upload a file from your local macOS machine to your GCP Compute Engine VM instance:

bash

 Copy code

```
scp "[LOCAL_FILE_PATH]" rohankodavalla@35.203.96.189:/home/rohankodavalla/catalog
```

Replace `"[LOCAL\_FILE\_PATH]"` with the full path to the file on your local macOS machine. For example:

bash

 Copy code

```
scp "/path/to/local/file.txt" rohankodavalla@35.203.96.189:/home/rohankodavalla/catalog
```

Ensure that you have SSH access to your GCP Compute Engine VM instance and that the remote destination path exists and is writable. Additionally, replace placeholders like `"[LOCAL\_FILE\_PATH]"` with the actual path to your file.

PR

You

/Users/rohankodavalla/Desktop/STUDY/GCP/Codes/ReadIt/catalog



ChatGPT

Great! If your file is located in the directory

`/Users/rohankodavalla/Desktop/STUDY/GCP/Codes/ReadIt/catalog` , you can use the following command to upload it to your GCP Compute Engine VM instance:

bash

Copy code

```
scp -r "/Users/rohankodavalla/Desktop/STUDY/GCP/Codes/ReadIt/catalog" rohankodavall...
```

In this command:

- ` -r` flag is used to recursively copy directories and their contents.
- ` /Users/rohankodavalla/Desktop/STUDY/GCP/Codes/ReadIt/catalog` is the path to your local directory containing the file(s) you want to upload.
- ` rohankodavalla@35.203.96.189` is your GCP Compute Engine VM instance username and IP address.
- ` /home/rohankodavalla/catalog` is the remote destination path where you want to copy the files on the VM instance.

This command will upload the entire `catalog` directory and its contents to the specified directory on your VM instance. Make sure that the destination directory exists and that you have the necessary permissions to write to it.

PR You

so in my vscode terminal im currently here -

```
(base) rohankodavalla@Rohans-Air catalog % dotnet publish -o publish --configuration release
MSBuild version 17.3.2+561848881 for .NET
Determining projects to restore...
All projects are up-to-date for restore.
catalog ->
/Users/rohankodavalla/Desktop/STUDY/GCP/Codes/ReadIt/catalog/bin/Release/net6.0/catalog.d
catalog -> /Users/rohankodavalla/Desktop/STUDY/GCP/Codes/ReadIt/catalog/publish/
(base) rohankodavalla@Rohans-Air catalog % cd publish
(base) rohankodavalla@Rohans-Air publish %
```

ChatGPT

Great, it looks like you've successfully published your .NET application to the `publish` directory.

Now, you can use the `scp` command to upload this directory to your GCP Compute Engine VM instance.

Since you're already in the `publish` directory in your terminal, you can run the `scp` command from here. Here's the command you can use:

bash

 Copy code

```
scp -r . rohankodavalla@35.203.96.189:/home/rohankodavalla/catalog
```

This command will recursively copy all files and directories from the current directory (`publish`) to the remote directory `/home/rohankodavalla/catalog` on your GCP Compute Engine VM instance.

Please make sure to replace `35.203.96.189` with the external IP address of your VM instance and ensure that you have SSH access and necessary permissions on the VM instance.



You

```
base) rohankodavalla@Rohans-Air publish % scp -r .
rohankodavalla@35.203.96.189:/home/rohankodavalla/catalog
```

```
The authenticity of host '35.203.96.189 (35.203.96.189)' can't be established.
ED25519 key fingerprint is SHA256:2K2h4AqL1VVq9Qsv/IuYlurfuHXCr6w+MPzwxv5mxok.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.203.96.189' (ED25519) to the list of known hosts.
rohankodavalla@35.203.96.189: Permission denied (publickey).
scp: Connection closed
(base) rohankodavalla@Rohans-Air publish %
```



ChatGPT

It seems that there was an issue with connecting to your GCP Compute Engine VM instance due to a permission denied error related to public key authentication.

Here are a few steps you can take to resolve this issue:

- 1. Check SSH Key:** Ensure that you have the correct SSH key configured for authentication with your GCP Compute Engine VM instance. Make sure that the public key associated with your SSH key pair is added to the `~/.ssh/authorized\_keys` file on your VM instance.
- 2. Verify SSH Access:** Confirm that you have SSH access to your VM instance by attempting to connect to it using SSH from your terminal:

```
bash
```

Copy code

```
ssh rohankodavalla@35.203.96.189
```

If you can successfully connect to your VM instance using SSH, it means your SSH key authentication is working correctly.

- 3. Check Firewall Rules:** Ensure that the firewall rules for your VM instance allow inbound SSH traffic (port 22) from your local machine. You can check and update firewall rules in the GCP Console or using the `gcloud` command-line tool.
- 4. Check User Permissions:** Verify that the user `rohankodavalla` has the necessary permissions to write to the `/home/rohankodavalla/catalog` directory on your VM instance.

If you continue to experience issues, double-check your SSH key configuration, firewall rules, and user permissions on the VM instance. If the issue persists, feel free to provide more details, and I'll do my best to assist you further.

PR

## You

Check SSH Key: Ensure that you have the correct SSH key configured for authentication with your GCP Compute Engine VM instance. Make sure that the public key associated with your SSH key pair is added to the `~/.ssh/authorized\_keys` file on your VM instance.

how to add it in my vm

PR

## ChatGPT

To add your SSH public key to the `~/.ssh/authorized\_keys` file on your GCP Compute Engine VM instance, you can follow these steps:

- 1. Locate Your SSH Public Key:** If you haven't already generated an SSH key pair, you'll need to do so. You can generate an SSH key pair using the `ssh-keygen` command on your local machine.

For example:

bash

 Copy code

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

This command will generate a new SSH key pair with the specified email address.

- 2. Copy Your SSH Public Key:** Once you've generated your SSH key pair, you'll need to copy the contents of the public key file (typically `~/.ssh/id\_rsa.pub`) to your clipboard.
- 3. SSH into Your VM Instance:** Use SSH to connect to your VM instance. You can do this from your terminal using the following command:

bash

 Copy code

```
ssh rohankodavalla@35.203.96.189
```

Replace `35.203.96.189` with the external IP address of your VM instance.

- 4. Add Your Public Key to `~/.ssh/authorized\_keys`:** Once connected to your VM instance, you can use a text editor like `nano` or `vim` to edit the `~/.ssh/authorized\_keys` file. If the file doesn't exist, you can create it.

bash

 Copy code

```
nano ~/.ssh/authorized_keys
```

Paste the contents of your SSH public key into this file, then save and exit the text editor.

- 5. Set Permissions:** Ensure that the permissions on the `~/.ssh/authorized\_keys` file are set correctly. You can set the permissions to `600` using the following command:

bash

 Copy code

```
bash
```

 Copy code

```
chmod 600 ~/.ssh/authorized_keys
```

This ensures that only the owner of the file can read or write to it.

**6. Exit SSH:** Once you've added your SSH public key to the `~/.ssh/authorized\_keys` file and set the correct permissions, you can exit the SSH session by typing:

```
bash
```

 Copy code

```
exit
```

Your SSH public key should now be added to the `~/.ssh/authorized\_keys` file on your GCP Compute Engine VM instance, allowing you to authenticate using SSH key pairs. You should now be able to use the `scp` command or SSH to connect to your VM instance without encountering permission denied errors.

PR

## You

can i add it on ui of google cloud > instance



## ChatGPT

Yes, you can add your SSH public key to your GCP Compute Engine VM instance using the Google Cloud Console. Here's how:

1. **Go to the VM Instances Page:** Open the [Google Cloud Console](#) in your web browser and navigate to the "VM Instances" page.
2. **Select Your VM Instance:** Click on the name of the VM instance to which you want to add the SSH public key.
3. **Edit Instance:** Click the "Edit" button at the top of the page to enter edit mode for the VM instance.
4. **Add SSH Keys:** Scroll down to the "SSH Keys" section. Here, you can add or remove SSH keys for connecting to the VM instance.
5. **Add Your SSH Public Key:** Click on "Add Item" and paste the contents of your SSH public key (`gcpkey.pub`) into the text field. You can provide a description if desired.
6. **Save Changes:** Click "Save" to save your changes and exit edit mode.

After adding your SSH public key through the Google Cloud Console, you should be able to SSH into your VM instance using your private key without having to enter a password.

Keep in mind that adding SSH keys through the Google Cloud Console only adds them to the `~/.ssh/authorized_keys` file for the specified user on the VM instance. It's essentially the same as manually adding the keys to the file via SSH.

PR You

tell me what exactly is my content to be added based on my generated -

```
(base) rohankodavalla@Rohans-Air publish % ssh-keygen -t rsa -b 4096 -C  
"rohankodavalla@gmail.com"
```

Generating public/private rsa key pair.

Enter file in which to save the key (/Users/rohankodavalla/.ssh/id\_rsa): gcpkey

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in gcpkey

Your public key has been saved in gcpkey.pub

The key fingerprint is:

```
SHA256:FjcRCcOEUmmppmAqlsAuYZyCquC0nmWqe5Sb4tIP/Go rohankodavalla@gmail.com
```

The key's randomart image is:

```
+---[RSA 4096]---
```

```
|= o..*+.oo |
```

```
|=*..=.... |
```

```
|X..*.o. |
```

```
|Bo + o. |
```

```
|*.. S |
```

```
|.ooo. . |
```

```
|..=o o |
```

```
|.+*.E. |
```

```
|=* .oo.. |
```

```
+---[SHA256]---
```

```
(base) rohankodavalla@Rohans-Air publish %
```

< 2 / 2 > ⌂



## ChatGPT

The content you need to add to your `authorized\_keys` file on your VM instance is the public key generated by `ssh-keygen`. In your case, the public key is stored in the file `gcpkey.pub`.

You can view the contents of your public key file (`gcpkey.pub`) using a text editor or by printing it directly from your terminal. Here's how you can do it:

```
bash
```

Copy code

```
cat gcpkey.pub
```

This command will display the contents of your public key in the terminal. It will look something like this:

```
css
```

Copy code

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQACQCz...  
...C9r5rohankodavalla@gmail.com
```

Copy the entire content of the `gcpkey.pub` file, starting with `ssh-rsa` and ending with your email address, into the `authorized\_keys` file on your VM instance.

After adding the public key to the `authorized\_keys` file, remember to set the correct permissions as mentioned earlier:

```
bash
```

Copy code

```
chmod 700 ~/.ssh  
chmod 600 ~/.ssh/authorized_keys
```

This ensures that only the owner of the files can read or write to them.

After adding the public key and setting the correct permissions, you should be able to SSH into your VM instance using your private key without having to enter a password.





You

```
(base) Rohans-Air:ssh rohankodavalla$ ls  
authorized_keys  id_rsa.pub  known_hosts.old  
id_rsa          known_hosts  ssh-key-2023-04-13.key  
(base) Rohans-Air:ssh rohankodavalla$
```



ChatGPT

It seems that the `gcpkey.pub` file is not present in your `~/.ssh` directory. This could be because you specified a different filename or location when generating the SSH key pair.

If you're unable to locate the `gcpkey.pub` file, you can regenerate your SSH key pair using the `ssh-keygen` command, ensuring that you specify the desired filename and location.

Here's how you can generate a new SSH key pair with the filename `gcpkey`:

bash

Copy code

```
ssh-keygen -t rsa -b 4096 -C "rohankodavalla@gmail.com" -f ~/.ssh/gcpkey
```

This command will generate a new RSA key pair with a 4096-bit key length, and the private key will be saved as `~/.ssh/gcpkey` and the public key as `~/.ssh/gcpkey.pub`.

Once the key pair is generated successfully, you can use the `cat` command to view the contents of the `gcpkey.pub` file and proceed to add it to the `authorized\_keys` file on your GCP Compute Engine VM instance.

```
● (base) rohankodavalla@Rohans-Air publish % ssh-keygen -t rsa -b 4096 -C "rohankodavalla@gmail.com" -f ~/.ssh/gcpkey  
Generating public/private rsa key pair.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /Users/rohankodavalla/.ssh/gcpkey  
Your public key has been saved in /Users/rohankodavalla/.ssh/gcpkey.pub  
The key fingerprint is:  
SHA256:AbVRCrApzh2z87dIb543uqh8pNk1rsITeL88l7Msgqo rohankodavalla@gmail.com  
The key's randomart image is:  
+---[RSA 4096]---+  
| ..0.0.. |  
| o o + |  
| . = + |  
| o o + . |  
| o.+ S |  
| . oo. o |  
| + Boo.o |  
| ..Bo***oo |  
| E. o**0X* . |  
+---[SHA256]---+  
○ (base) rohankodavalla@Rohans-Air publish %
```

```
(base) Rohans-Air:~ ssh rohankodavalla$ cat gckey.pub
cat: gckey.pub: No such file or directory
(base) Rohans-Air:~ ssh rohankodavalla$ cat gcpkey.pub
cat: gcpkey.pub: No such file or directory
(base) Rohans-Air:~ ssh rohankodavalla$ ls
authorized_keys      id_rsa.pub      known_hosts.old
id_rsa               known_hosts    ssh-key-2023-04-13.key
(base) Rohans-Air:~ ssh rohankodavalla$ ls
authorized_keys      id_rsa         known_hosts.old
gcpkey               id_rsa.pub     ssh-key-2023-04-13.key
gcpkey.pub           known_hosts
(base) Rohans-Air:~ ssh rohankodavalla$ cat ~/.ssh/gcpkey.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQACQC2MdgpaiWF/tIKOTz+4JY9iTbRPMvJqXdcxK/Y2CsHzuQPafIxffF9RtAbmwqJLU2/KcffrLDmaTc0+D1Q199VKxXhVLfXwgN81oBXCBQ
cIhhDP8Vb3JPJy7DMigBTPJaBXcOYC7wRPyvtOuA2gRdQ8sU/fS4taKJHyg3CbyKSbPrVV/Ccyj7wPFcKqiHJnkhNPfAjKpLtxn+q7JO/2GXBkzRWFKYDM977eSrCiZXtdmuzDaldLR
+rdfHnOmQsjlyTjVnuMTmE17AG/I/jb+Vo7EefXdPgnpgncGPjfvptra4UggzquPWIS1067/zu0jltDj64dajZM0mZK1H3B0IAj4YlJMpfIAeE7fd+GbJ5y5fzw18xoEkBL2aHfXYrUsQV
+xhbomhif0kQD85S863vfoENA00jq+jANU4EvvizW7muaEzRKt+kVJDOLuXaJN1W9Ko7B7rRqhmdkuH+KDq3zurUsfQWJRVAu866By7D9Yq+yHA3jL+ndUuQ9/vUw0fkal3qyRUbmcYKLKH
A5FUd5i/7dP4gS0w83iwF705SWjQA8mA8TkXT+0cIR0kSvElBhK13qgxEu1As+LHezcbsTpSfioyp2dDUHR9+//LU12y0Jk39+9i0ozArL74InjW6WU/JW0iF5r8Lajkmk+hh0JNlgqw1tFI
ow== rohankodavalla@gmail.com
(base) Rohans-Air:~ ssh rohankodavalla$
```

## NOW ADD IT TO VM IN GCP SSH KEY

The screenshot shows the Google Cloud Platform Catalog VM interface. At the top, there are tabs for 'catalog-vm', 'EDIT', 'RESET', 'CREATE MACHINE IMAGE', 'CREATE SIMILAR', and 'OPERATIONS'. Below these tabs, there are four main sections: 'DETAILS', 'OBSERVABILITY', 'OS INFO', and 'SCREENSHOT'. Under 'DETAILS', there is a section for 'Integrity Monitoring' which is set to 'On'. In the 'SSH Keys' section, there is a table with two rows. The first row has 'Username' as 'rohankodavalla' and 'Key' as 'ecdsa-sha2-nistp256...'. The second row has 'Username' as 'rohankodavalla' and 'Key' as 'ssh-rsa...'. A third row is partially visible, starting with 'rohankodavalla' and containing a very long, complex public key string. At the bottom of the page, there is a section for 'Block project-wide SSH keys' which is set to 'Off'.

Username	Key
rohankodavalla	ecdsa-sha2-nistp256...
rohankodavalla	ssh-rsa...
rohankodavalla	ssh-rsa AAAAAB3NzaC1yc2EAAAQABAAQACQC2MdgpaiWF/tIKOTz+4JY9iTbRPMvJqXdcxK/Y2CsHzuQPafIxffF9RtAbmwqJLU2/KcffrLDmaTc0+D1Q199VKxXhVLfXwgN81oBXCBQ cIhhDP8Vb3JPJy7DMigBTPJaBXcOYC7wRPyvtOuA2gRdQ8sU/fS4taKJHyg3CbyKSbPrVV/Ccyj7wPFcKqiHJnkhNPfAjKpLtxn+q7JO/2GXBkzRWFKYDM977eSrCiZXtdmuzDaldLR +rdfHnOmQsjlyTjVnuMTmE17AG/I/jb+Vo7EefXdPgnpgncGPjfvptra4UggzquPWIS1067/zu0jltDj64dajZM0mZK1H3B0IAj4YlJMpfIAeE7fd+GbJ5y5fzw18xoEkBL2aHfXYrUsQV +xhbomhif0kQD85S863vfoENA00jq+jANU4EvvizW7muaEzRKt+kVJDOLuXaJN1W9Ko7B7rRqhmdkuH+KDq3zurUsfQWJRVAu866By7D9Yq+yHA3jL+ndUuQ9/vUw0fkal3qyRUbmcYKLKH A5FUd5i/7dP4gS0w83iwF705SWjQA8mA8TkXT+0cIR0kSvElBhK13qgxEu1As+LHezcbsTpSfioyp2dDUHR9+//LU12y0Jk39+9i0ozArL74InjW6WU/JW0iF5r8Lajkmk+hh0JNlgqw1tFI ow== rohankodavalla@gmail.com

## NOW WE ESTABLISHED SSH TO THE VM FROM LOCAL MAC (VS CODE TERMINAL)

Now we have to copy files in publish to the vm , so use -

```
(base) rohankodavalla@Rohans-Air publish % scp -i ~/.ssh/gcpkey -r * rohankodavalla@35.203.96.189:/home/rohankodavalla/catalog
BouncyCastle.Crypto.dll                                         100% 3241KB   1.4MB/s  00:02
Google.Protobuf.dll                                         100% 400KB   1.3MB/s  00:00
K4os.Compression.LZ4.Streams.dll                           100% 79KB    1.2MB/s  00:00
K4os.Compression.LZ4.dll                                     100% 66KB    1.2MB/s  00:00
K4os.Hash.xxHash.dll                                       100% 13KB   809.0KB/s  00:00
Microsoft.Data.SqlClient.dll                            100% 350KB   1.3MB/s  00:00
Microsoft.EntityFrameworkCore.Abstractions.dll           100% 34KB    1.1MB/s  00:00
Microsoft.EntityFrameworkCore.InMemory.dll             100% 200KB   1.3MB/s  00:00
Microsoft.EntityFrameworkCore.Relational.dll          100% 1800KB  1.4MB/s  00:01
Microsoft.EntityFrameworkCore.SqlServer.dll            100% 442KB   1.3MB/s  00:00
Microsoft.EntityFrameworkCore.dll                         100% 2034KB  1.4MB/s  00:01
Microsoft.Extensions.Caching.Abstractions.dll          100% 31KB    1.1MB/s  00:00
Microsoft.Extensions.Caching.Memory.dll                100% 43KB    1.1MB/s  00:00
Microsoft.Extensions.Configuration.Abstractions.dll    100% 29KB    1.1MB/s  00:00
Microsoft.Extensions.DependencyInjection.Abstractions.dll 100% 44KB    1.1MB/s  00:00
Microsoft.Extensions.DependencyInjection.dll              100% 83KB    1.3MB/s  00:00
```

To copy only the files and directories inside the `publish` directory, you should navigate to the `publish` directory first and then run the `scp` command. Here's the corrected command:

bash

 Copy code

```
scp -i ~/.ssh/gcpkey -r * rohankodavalla@35.203.96.189:/home/rohankodavalla/catalog
```

This command will copy all the contents of the `publish` directory (`\*`) to the destination directory on your VM instance. Make sure to execute this command from the directory containing the `publish` directory.

Now verify on vm instance -

```
rohankodavalla@catalog-vm:~/catalog$ ls
BouncyCastle.Crypto.dll
Google.Protobuf.dll
K4os.Compression.LZ4.Streams.dll
K4os.Compression.LZ4.dll
K4os.Hash.xxHash.dll
Microsoft.Data.SqlClient.dll
Microsoft.EntityFrameworkCore.Abstractions.dll
Microsoft.EntityFrameworkCore.InMemory.dll
Microsoft.EntityFrameworkCore.Relational.dll
Microsoft.EntityFrameworkCore.SqlServer.dll
Microsoft.EntityFrameworkCore.dll
Microsoft.Extensions.Caching.Abstractions.dll
Microsoft.Extensions.Caching.Memory.dll
Microsoft.Extensions.Configuration.Abstractions.dll
Microsoft.Extensions.DependencyInjection.Abstractions.dll
Microsoft.Extensions.DependencyInjection.dll
Microsoft.Extensions.Logging.Abstractions.dll
Microsoft.Extensions.Logging.dll
Microsoft.Extensions.Options.dll

Microsoft.Extensions.Primitives.dll
Microsoft.Identity.Client.dll
Microsoft.IdentityModel.JsonWebTokens.dll
Microsoft.IdentityModel.Logging.dll
Microsoft.IdentityModel.Protocols.OpenIdConnect.dll
Microsoft.IdentityModel.Protocols.dll
Microsoft.IdentityModel.Tokens.dll
Microsoft.Win32.SystemEvents.dll
MySql.Data.dll
MySql.EntityFrameworkCore.dll
ServiceStack.Common.dll
ServiceStack.Interfaces.dll
ServiceStack.Redis.dll
ServiceStack.Text.dll
System.Configuration.ConfigurationManager.dll
System.Diagnostics.DiagnosticSource.dll
System.Drawing.Common.dll
System.IO.Pipelines.dll
System.IdentityModel.Tokens.Jwt.dll

System.Runtime.Caching.dll
System.Security.Cryptography.ProtectedData.dll
System.Security.Permissions.dll
System.Text_ENCODINGS.Web.dll
System.Text.Json.dll
System.Windows.Extensions.dll
ZstdSharp.dll
appsettings.Development.json
appsettings.json
catalog
catalog.deps.json
catalog.dll
catalog.pdb
catalog.runtimeconfig.json
gcpkey
gcpkey.pub
runtimes
web.config
wwwroot
```

Success!

Now to run the app one last thing to do is - Set Production Environment Variable and run the app, refer .txt for putty steps.

```

rohankodavalla@catalog-vm:~/catalog$ export ASPNETCORE_ENVIRONMENT=Production
rohankodavalla@catalog-vm:~/catalog$ dotnet catalog.dll
Environment: Production
Running in production mode...
warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
      No XML encryptor configured. Key {f04348ab-1a56-463b-941c-59af93e33e9} may be persisted to storage 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/rohankodavalla/catalog

```

It is success, listening on > Now listening on: [http://\[::\]:8080](http://[::]:8080)

But we still cannot access the app, since this port is blocked by firewall

\* Change firewall rule to open the 8080 port \*

The screenshot shows the Google Cloud VPC networks interface. The left sidebar is titled "VPC network" and lists several options: IP addresses, Internal ranges, Bring your own IP, Firewall, Routes, VPC network peering, Shared VPC, Serverless VPC access, and Packet mirroring. The main area is titled "VPC networks" and shows a table with one row for the "default" VPC network. The table columns are: Name (sorted by name), Subnets, MTU, Mode, IPv6 ULA range, Gateways, Firewall rules, and Global d. The "default" row has values: 42, 1460, Auto, and Off. A modal window titled "Get started with real-time analytics" is open, encouraging users to use Network Intelligence Center for monitoring and troubleshooting. It lists several features with green checkmarks: Visualize your network resources, Diagnose and prevent connectivity issues, View packet loss and latency metrics, and Keep your firewall rules strict and efficient. Buttons for "TRY NOW" and "REMIND ME LATER" are at the bottom of the modal.

Name	Subnets	MTU	Mode	IPv6 ULA range	Gateways	Firewall rules	Global d
<a href="#">default</a>	42	1460	Auto			6	Off

# Duplicate tab > VPC Networks > firewall > click default-allow-http

The screenshot shows the Google Cloud Firewall policies interface. On the left, a sidebar lists various security components: Network Security, Secure Web Proxy, Cloud Armor, Adaptive Protection, Cloud Armor Service Tier, Cloud IDS, IDS Dashboard, IDS Endpoints, IDS Threats, Cloud NGFW, Dashboard, Firewall policies (which is selected and highlighted in blue), Threats, and Firewall endpoints. The main area is titled 'Firewall policies' and contains sections for 'CREATE FIREWALL POLICY' and 'CREATE FIREWALL RULE'. A note states that traffic from outside the network is blocked by default. Below this, a note mentions App Engine firewalls. A warning message indicates that SMTP port 25 is disallowed. The main table lists six firewall rules:

Name	Type	Targets	Filters	Protocols / ports	Action	Priority
default-allow-http	Ingress	http-server	IP ranges:	tcp:80	Allow	1
default-allow-https	Ingress	https-	IP ranges:	tcp:443	Allow	1
default-allow-icmp	Ingress	Apply to all	IP ranges:	icmp	Allow	65
default-allow-internal	Ingress	Apply to all	IP ranges:	tcp:0-65535 udp:0-65535 icmp	Allow	65
default-allow-rdo	Ingress	Apply to all	IP ranges:	tcp:3389	Allow	65



Free trial status: \$409.68 credit and 90 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

Google Cloud My First Project vpc Search

Network Security

Secure Web Proxy

Cloud Armor

Cloud Armor policies

Adaptive Protection

Cloud Armor Service Tier

Cloud IDS

IDS Dashboard

IDS Endpoints

IDS Threats

Cloud NGFW

Dashboard

Firewall policies

Threats

Firewall endpoints

Common components

Source IPv4 ranges \* 0.0.0.0/0

Second source filter None

Destination filter None

Protocols and ports

Allow all

Specified protocols and ports

TCP

Ports 80

E.g. 20, 50-60

UDP

Ports E.g. all

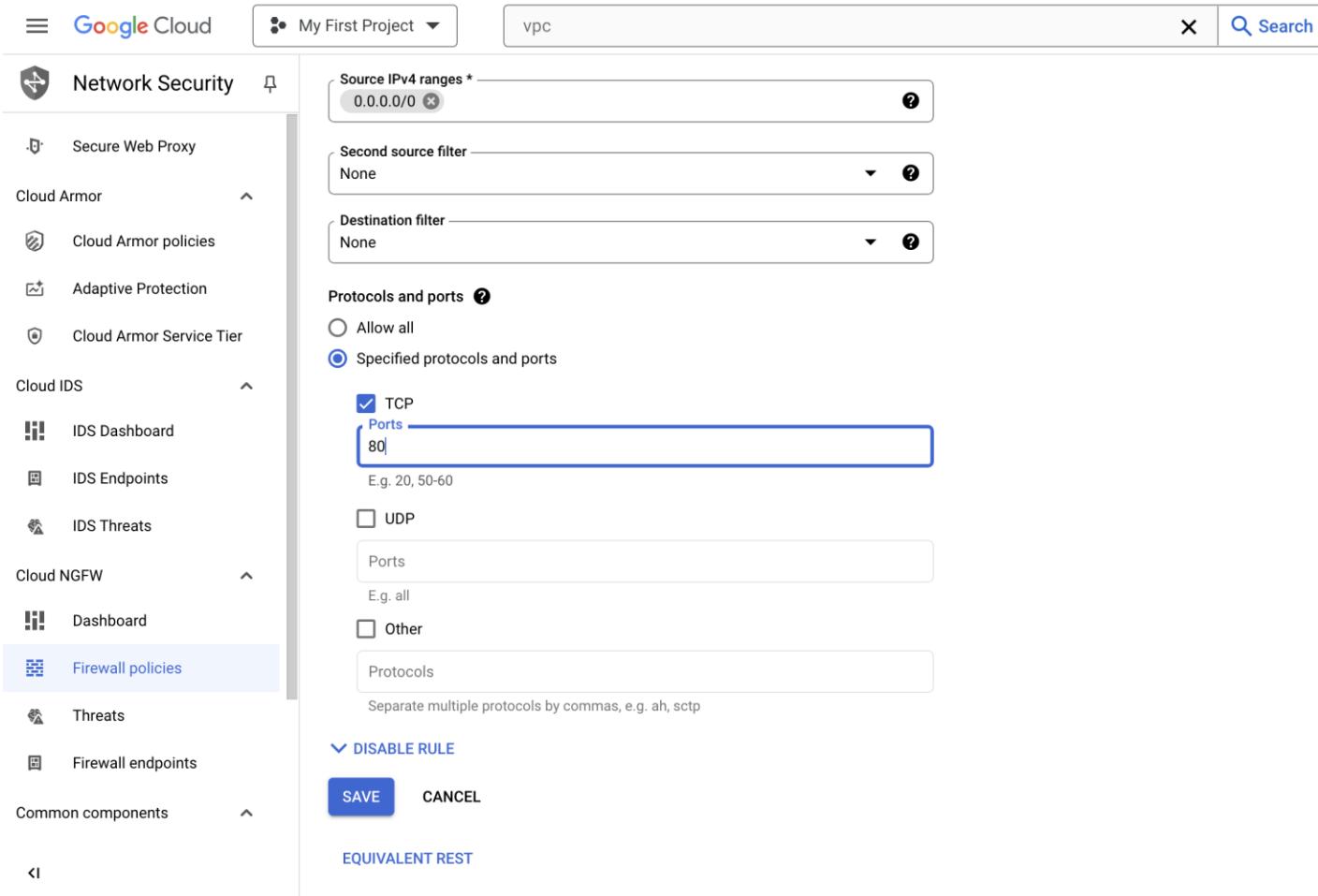
Other

Protocols Separate multiple protocols by commas, e.g. ah, sctp

DISABLE RULE

SAVE CANCEL

EQUIVALENT REST



edit > change port to 8080 and save

Google Cloud My First Project vpc Search

Network Security

Secure Web Proxy

Cloud Armor

Cloud Armor policies

Adaptive Protection

Cloud Armor Service Tier

Cloud IDS

IDS Dashboard

IDS Endpoints

IDS Threats

Cloud NGFW

Dashboard

Firewall policies

Threats

Firewall endpoints

Common components

Source filters IP ranges 0.0.0.0/0

Protocols and ports tcp:8080

Enforcement Enabled

Insights None

Hit count monitoring

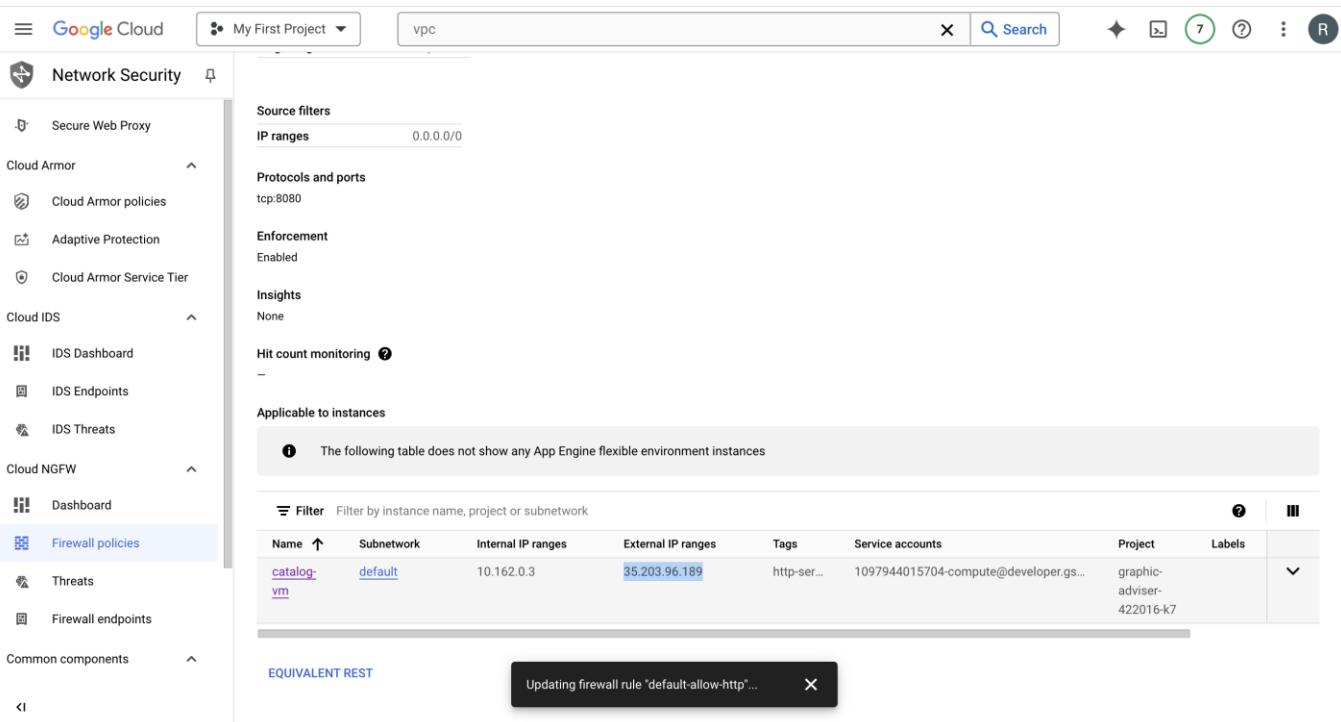
Applicable to instances

The following table does not show any App Engine flexible environment instances

Name	Subnetwork	Internal IP ranges	External IP ranges	Tags	Service accounts	Project	Labels
catalog-vm	default	10.162.0.3	35.203.96.189	http-ser...	1097944015704-compute@developer.g...	graphic-adviser-422016-k7	

EQUIVALENT REST

Updating firewall rule "default-allow-http"...



Copy the external ip newly generated and open new tab , add port 8080 and enter

The screenshot shows a browser window with the following details:

- Address Bar:** Shows the URL `35.203.96.189:8080`.
- Tab Bar:** Contains several tabs, including "Blind 75 LeetC", "35.203.96.189:8080", and "35.203.96.189:8080 - Google Search".
- Content Area:**
  - Header:** "ReadIt!" logo, "Home" link, "Weather" link.
  - Cart Icon:** Shows 0 items.
  - Section Title:** "Our Books:"
  - Table Headers:** "Name", "Author", "Pages", "Price".
  - Buttons:** "Load Books to DB" button.
- Page Footer:** © 2024 - ReadIt!

Finally !

```

Microsoft.EntityFrameworkCore.InMemory.dll           Microsoft.Win32.SystemEvents.dll
Microsoft.EntityFrameworkCore.Relational.dll        MySql.Data.dll
Microsoft.EntityFrameworkCore.SqlServer.dll         MySql.EntityFrameworkCore.dll
Microsoft.EntityFrameworkCore.dll                  ServiceStack.Common.dll
Microsoft.Extensions.Caching.Abstractions.dll    ServiceStack.Interfaces.dll
Microsoft.Extensions.Caching.Memory.dll          ServiceStack.Redis.dll
Microsoft.Extensions.Configuration.Abstractions.dll ServiceStack.Text.dll
Microsoft.Extensions.DependencyInjection.Abstractions.dll System.Configuration.ConfigurationManager.dll
Microsoft.Extensions.DependencyInjection.dll         System.Diagnostics.DiagnosticSource.dll
Microsoft.Extensions.Logging.Abstractions.dll     System.Drawing.Common.dll
Microsoft.Extensions.Logging.dll                 System.IO.Pipelines.dll
Microsoft.Extensions.Options.dll                  System.IdentityModel.Tokens.Jwt.dll

appsettings.Development.json
appsettings.json
catalog
catalog.deps.json
catalog.dll
catalog.pdb
catalog.runtimeconfig.json
gcpkey
gcpkey.pub
runtimes
web.config
wwwroot

System.Runtime.Caching.dll
System.Security.Cryptography.ProtectedData.dll
System.Security.Permissions.dll
System.Text.Encoding.Web.dll
System.Text.Json.dll
System.Windows.Extensions.dll
ZstdSharp.dll
appsettings.Development.json
appsettings.json
catalog
catalog.deps.json
catalog.dll
catalog.pdb
catalog.runtimeconfig.json
gcpkey
gcpkey.pub
runtimes
web.config
wwwroot

rohankodavalla@catalog-vm:~/catalog$ rm -rf *
rohankodavalla@catalog-vm:~/catalog$ ls
rohankodavalla@catalog-vm:~/catalog$ ls
BouncyCastle.Crypto.dll
Google.Protobuf.dll
K4os.Compression.LZ4.Streams.dll
K4os.Compression.LZ4.dll
K4os.Hash.xxHash.dll
Microsoft.Data.SqlClient.dll
Microsoft.EntityFrameworkCore.Abstractions.dll
Microsoft.EntityFrameworkCore.InMemory.dll
Microsoft.EntityFrameworkCore.Relational.dll
Microsoft.EntityFrameworkCore.SqlServer.dll
Microsoft.EntityFrameworkCore.dll
Microsoft.Extensions.Caching.Abstractions.dll
Microsoft.Extensions.Caching.Memory.dll
Microsoft.Extensions.Configuration.Abstractions.dll
Microsoft.Extensions.DependencyInjection.Abstractions.dll
Microsoft.Extensions.DependencyInjection.dll
Microsoft.Extensions.Logging.Abstractions.dll
Microsoft.Extensions.Logging.dll
Microsoft.Extensions.Options.dll
rohankodavalla@catalog-vm:~/catalog$ export ASPNETCORE_ENVIRONMENT=Production
rohankodavalla@catalog-vm:~/catalog$ dotnet catalog.dll
Environment: Production
Running in production mode...
warn: Microsoft.AspNetCore.DataProtection.KeyManagement.XmlKeyManager[35]
No XML encryptor configured. Key (f04348ab-1a56-463b-941c-59afdf93e3e9) may be persisted to storage 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/rohankodavalla/catalog
Loaded 4 books

```

But this is possible only because it is running on vm , but we want to access website even if this session is closed.

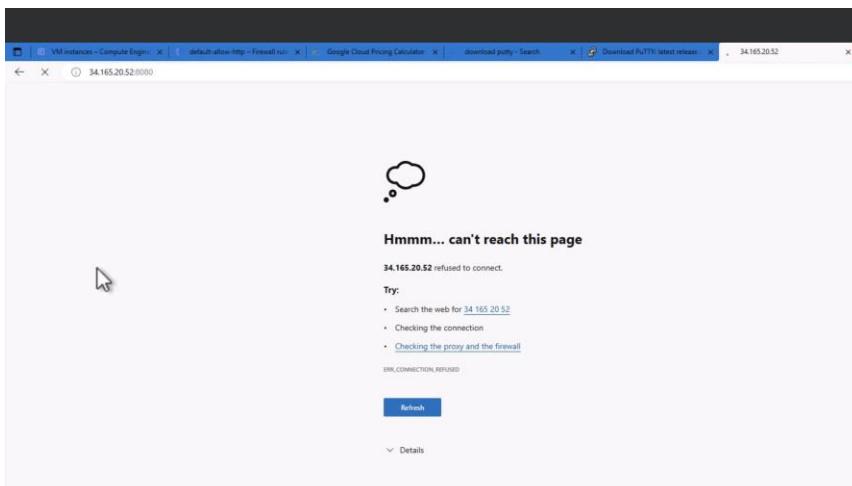
```

Microsoft.EntityFrameworkCore.InMemory.dll           Microsoft.Win32.SystemEvents.dll
Microsoft.EntityFrameworkCore.Relational.dll        MySql.Data.dll
Microsoft.EntityFrameworkCore.SqlServer.dll         MySql.EntityFrameworkCore.dll
Microsoft.EntityFrameworkCore.dll                  ServiceStack.Common.dll
Microsoft.Extensions.Caching.Abstractions.dll    ServiceStack.Interfaces.dll
Microsoft.Extensions.Caching.Memory.dll          ServiceStack.Redis.dll
Microsoft.Extensions.Configuration.Abstractions.dll ServiceStack.Text.dll
Microsoft.Extensions.DependencyInjection.Abstractions.dll System.Configuration.ConfigurationManager.dll

appsettings.Development.json
appsettings.json
catalog
catalog.deps.json
catalog.dll
catalog.pdb
catalog.runtimeconfig.json
gcpkey

```

if closed tab



he used putty so error , but

for me it worked well .

The screenshot shows a web browser window with four tabs open: 'VM instances - Compute Eng', 'default-allow-http - Firewall', 'Troubleshoot .NET Core Proj...', and 'ReadIt! - catalog'. The 'ReadIt! - catalog' tab is active, displaying a page with the title 'ReadIt!' and a 'Home' link. Below the title is a shopping cart icon with '(0)' next to it. The main content is titled 'Our Books:' and lists four books:

Name	Author	Pages	Price	Action
Rama II	Arthur C. Clark	281	44.23	Add to shoppe
Exhalation	Ted Chiang	556	50.99	Add to shoppe
Traffic Secrets	Russell Brunson	306	18.97	Add to shoppe
Clean Code	Robert Martin	464	87	Add to shoppe

At the bottom right of the table is a button labeled 'Load Books to DB'.

So in general terms we need a startup script that starts this as soon as vm is up :

So go to instances > vm > edit > automation > startup script (from .txt)

```
debian-dotnet-commands.txt
=====
mkdir catalog
cd catalog
pwd
* copy the folder path *

XXXXXXXXXXXXXXXXXXXX-- NOW USE SCP CMD ABOVE (refer pdf) AND DONT FOLLOW BELOW -----XXXXXXX
Copy files from local computer to the VM using SCP
=====
* From the VS Code terminal *
cd publish
scp -r -i c:\catalog-ssh.ppk * {USERNAME}@{IP_ADDRESS}:~/home/{USERNAME}/catalog
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Do below -
Set Production Environment Variable and run the app
=====
* In PuTTY: * not putty , in vm cmd line do this -
export ASPNETCORE_ENVIRONMENT=Production
dotnet catalog.dll

* Change firewall rule to open the 8080 port *

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

Compute Engine

Virtual machines

- VM instances
- Instance templates
- Sole-tenant nodes
- Machine images
- TPUs
- Committed use discounts
- Reservations
- Migrate to Virtual Machin...

Storage

- Disks
- Storage Pools
- Snapshots
- Marketplace

← Edit catalog-vm instance

**Metadata**

You can set custom metadata for an instance or project outside of the server-defined metadata. 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

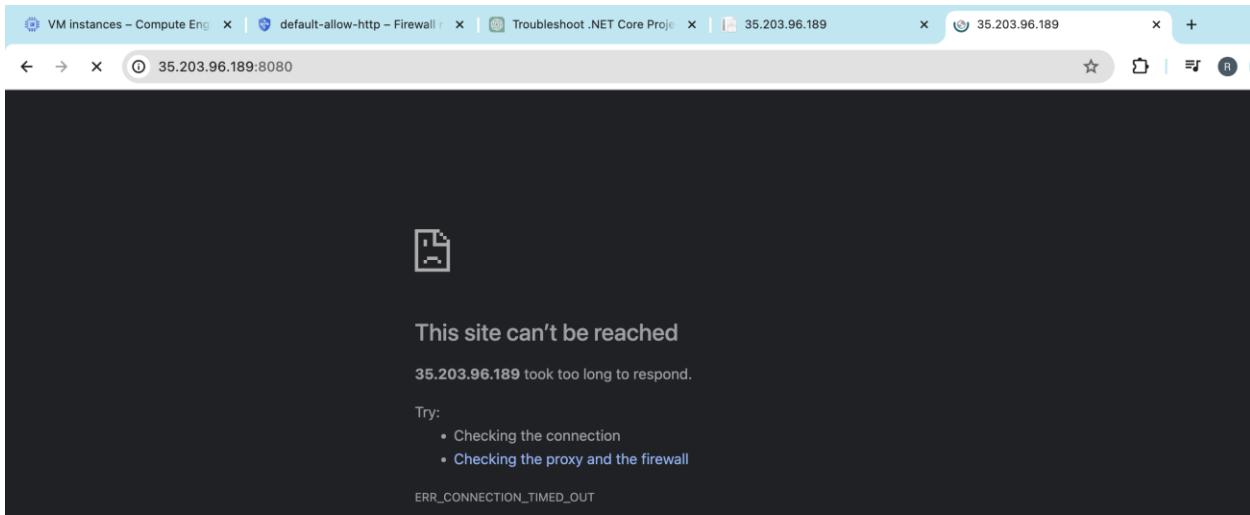
You can choose to specify a startup script that will run when your instance boots up or restarts. Startup scripts can be used to install software and updates, and to ensure that services are running within the virtual machine. [Learn more](#)

**Sole tenancy**

Provision VMs on a sole-tenant group or on a specific node within the group, within your project or projects shared with you. [Learn more](#)

Selected machine type is not compatible with Sole-tenancy.

Now stop the instance and see if website works .



After stopping vm, it takes forever to load , so now start and see

The screenshot shows the Google Cloud Compute Engine interface. In the top navigation bar, there are tabs for 'VM instances', 'CREATE INSTANCE', 'IMPORT VM', and 'REFRESH'. Below the navigation bar, there are three tabs: 'INSTANCES' (selected), 'OBSERVABILITY', and 'INSTANCE SCHEDULES'. The main area displays a table of VM instances. One instance, 'catalog-vm', is selected and highlighted with a blue border. The table columns include 'Status', 'Name', 'Zone', 'Recommendations', 'In use by', 'Internal IP', 'External IP', and 'Connect'. The 'External IP' column shows '35.203.8.236 (nic0)'. To the right of the table are buttons for 'SSH' and a more options menu. Below the table, a section titled 'Related actions' lists several links: 'Explore Backup and DR', 'View billing report', 'Monitor VMs', 'Explore VM logs', 'Set up firewall rules', and 'Patch management'.

You will get new external ip (not the one you used earlier , so use that ip:8080 and see )

The screenshot shows a web browser window with multiple tabs open. The active tab is 'ReadIt! - catalog' with the URL '35.203.8.236:8080'. The page content includes a header with 'ReadIt!', 'Home', and 'Weather' links, a shopping cart icon, and a section titled 'Our Books:'. Below this, there is a table with columns 'Name', 'Author', 'Pages', and 'Price', and a button labeled 'Load Books to DB'.

Voila !

10:09 7. Compute Engine 52. Publish the Catalog app

We now have a cloud VM instance running our catalog app.  
We don't need any special configuration for that.  
We don't need to connect to the VM instance for that to happen.  
And we completed successfully our first step in running the ReadIt app in GCP.