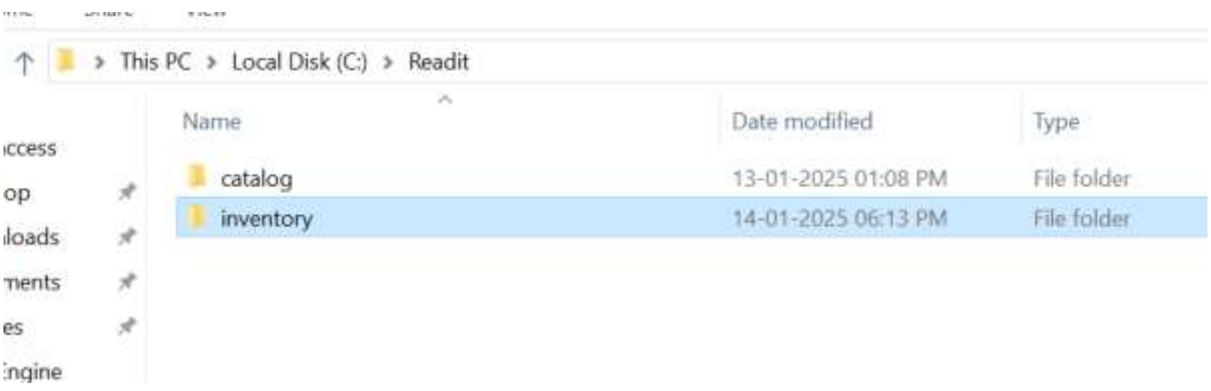
Introduction:-

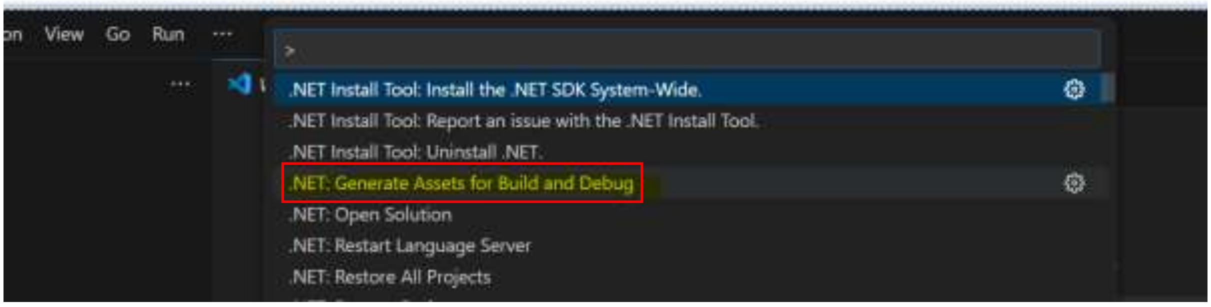
In this lab we will create a Flexible app service at GCP App Engine using code. Here we will first test the code on our local pc and then deploy it to GCP App Engine. We will test the application and also check the underlying infrastructure being used for this app.

1) Extract the "inventorybaseline" zip folder and copy the inventory folder to the ReadIt folder. You can get the inventory baseline zip folder from GitHub, so, first download it.
2) Then you need to copy the inventory folder you get from the zip file and paste it in the same folder where you have the catalog folder stored on your local machine.



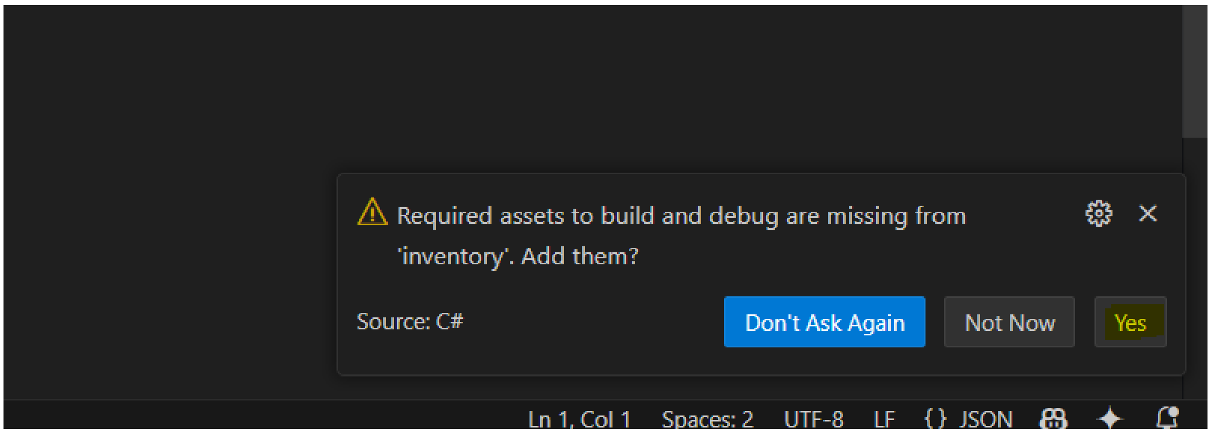
Name	Date modified	Type
catalog	13-01-2025 01:08 PM	File folder
inventory	14-01-2025 06:13 PM	File folder

3) Open this folder in VS Code, and also in VS Code go to View> Command Palette and run .Net Generate Assets...... like earlier



.NET Install Tool: Install the .NET SDK System-Wide.
.NET Install Tool: Report an issue with the .NET Install Tool.
.NET Install Tool: Uninstall .NET.
.NET: Generate Assets for Build and Debug
.NET: Open Solution
.NET: Restart Language Server
.NET: Restore All Projects

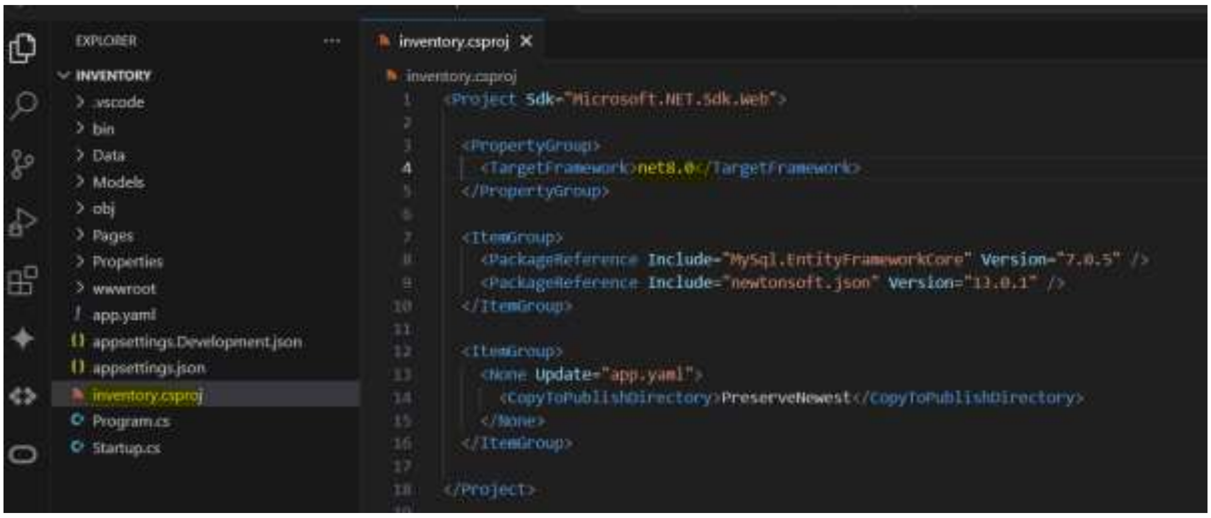
4) If see pop-up regarding debug missing, click yes



⚠ Required assets to build and debug are missing from 'inventory'. Add them?

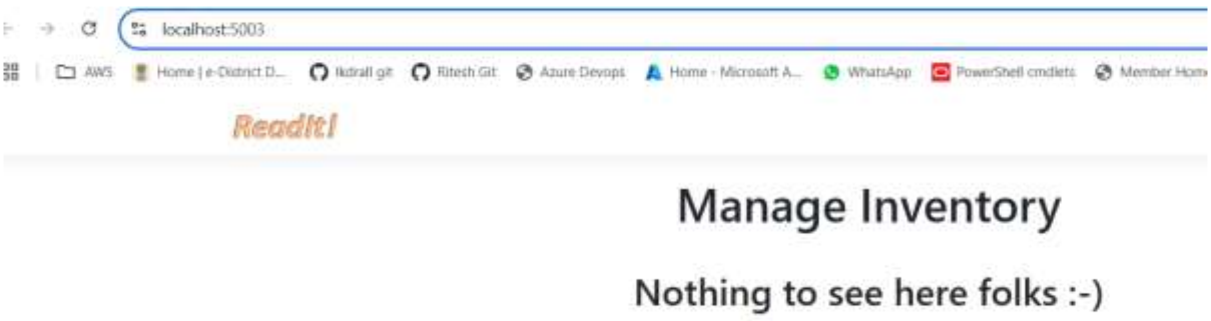
Source: C#

5) Also, make sure the target framework is net8.0 in inventory.csproj file



```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>net8.0</TargetFramework>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="MySql.EntityFrameworkCore" Version="7.0.5" />
    <PackageReference Include="newtonsoft.json" Version="13.0.1" />
  </ItemGroup>

  <ItemGroup>
    <None Update="app.yaml">
      <CopyToPublishDirectory>PreserveNewest</CopyToPublishDirectory>
    </None>
  </ItemGroup>

</Project>
```

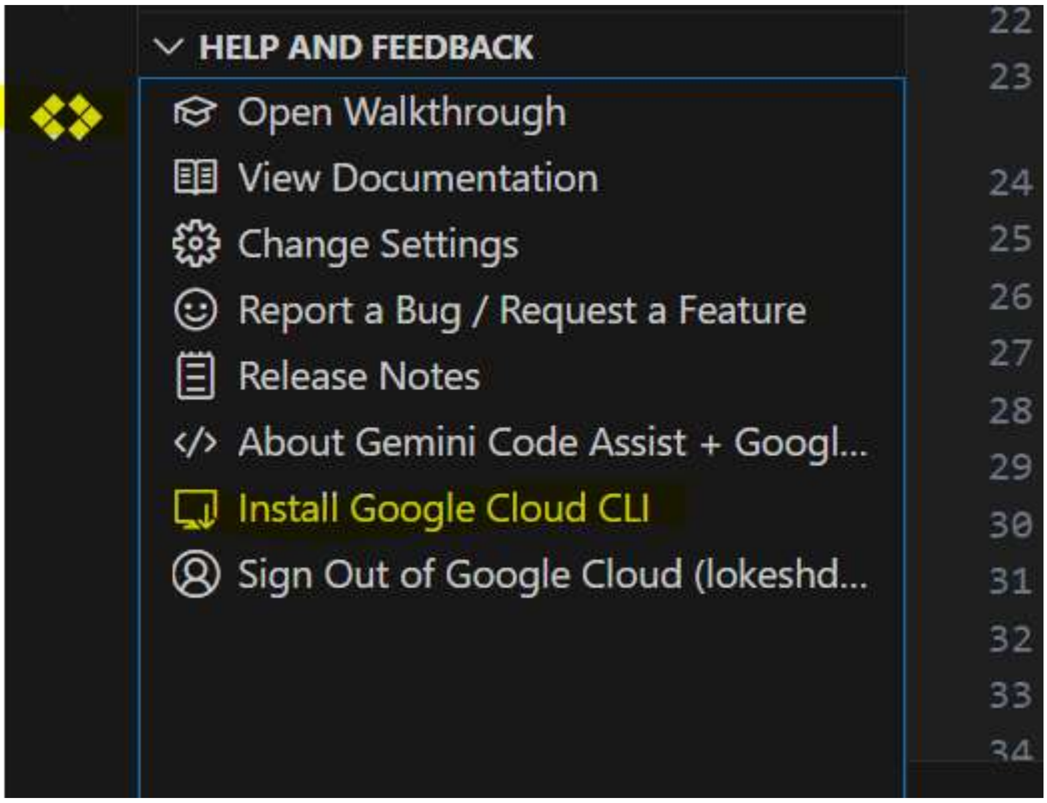
6) Press F5 to run the code, once it completes it should open the inventory webpage



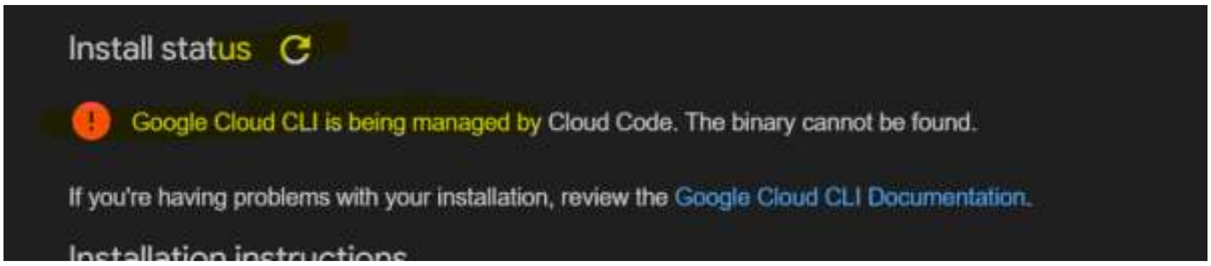
Manage Inventory

Nothing to see here folks :-)

7) In VS Code, you can optionally go to index.cshtml file under pages to see the code
8) Go to cloud code in VS Code and click install Google Cloud CLI


HELP AND FEEDBACK
Open Walkthrough
View Documentation
Change Settings
Report a Bug / Request a Feature
Release Notes
About Gemini Code Assist + Googl...
Install Google Cloud CLI
Sign Out of Google Cloud (lokeshd...

9) If you see error below

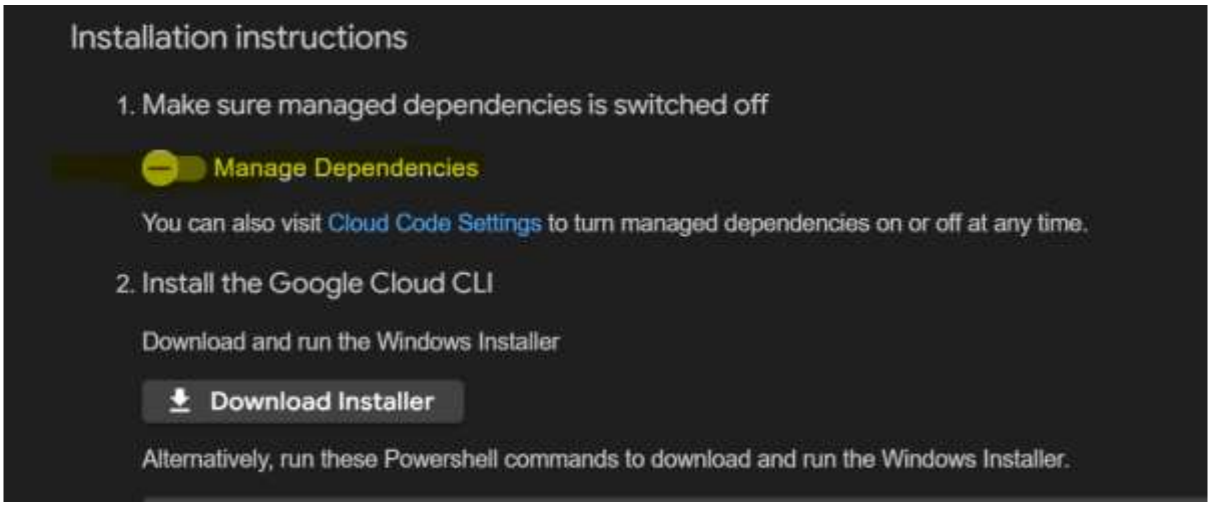

Install status

Google Cloud CLI is being managed by Cloud Code. The binary cannot be found.

If you're having problems with your installation, review the Google Cloud CLI Documentation.

Installation instructions

10) First, turn **off** manage dependencies, then click download installer


Installation instructions

1. Make sure managed dependencies is switched off

You can also visit Cloud Code Settings to turn managed dependencies on or off at any time.

2. Install the Google Cloud CLI

Download and run the Windows Installer

Alternatively, run these Powershell commands to download and run the Windows Installer.

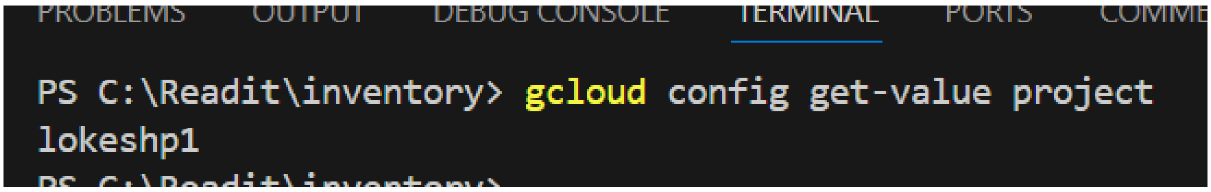
11) This will ask to download the installer from the browser, download and run it as an administrator.
Install the installer, also make it for **all users**.

12) Follow all installing instructions by keeping default, after finish when ask to configure, say **y**, select your project and say **n** when ask to set default regions

13) Restart vscode

14) (**Error**) In case see an error saying like "gcloud: File C:\Program Files (x86)\Google\Cloud SDK\google-cloud-sdk\bin\gcloud.ps1 cannot be loaded. The file C:\Program Files (x86)\Google\Cloud SDK\google-cloud-sdk\bin\gcloud.ps1 is not digitally signed…….", then open PowerShell via admin and run the below command
Set-ExecutionPolicy -ExecutionPolicy Unrestricted

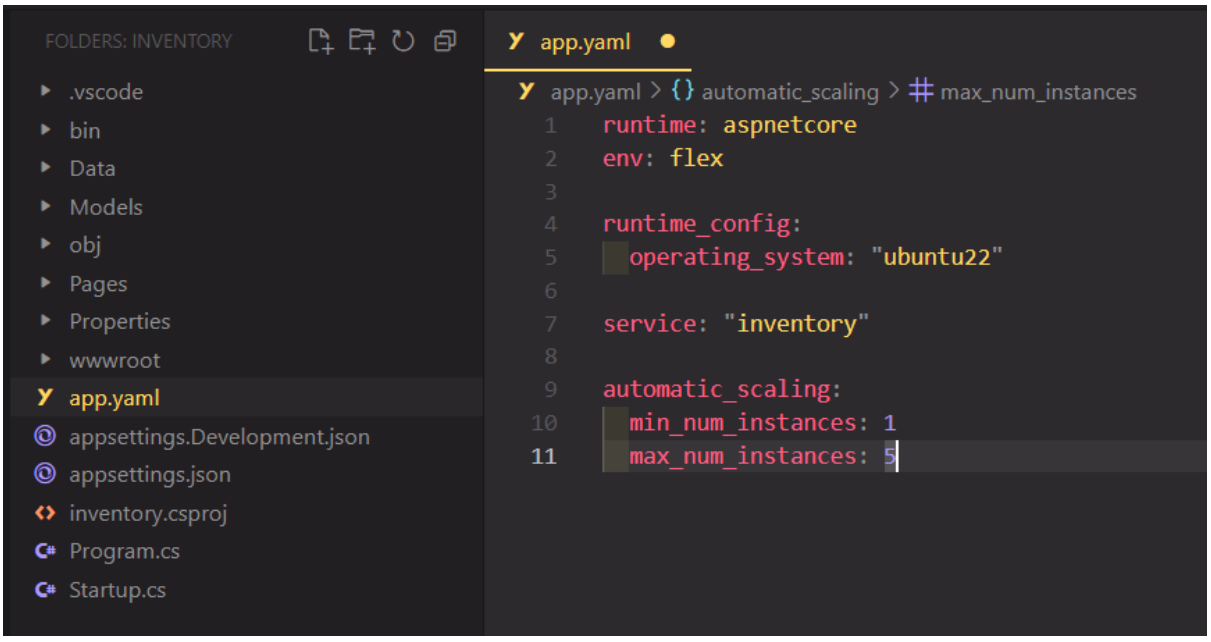
15) Run **gcloud config get-value project** to confirm your project is selected



```
PS C:\Readit\inventory> gcloud config get-value project
lokeshp1
```

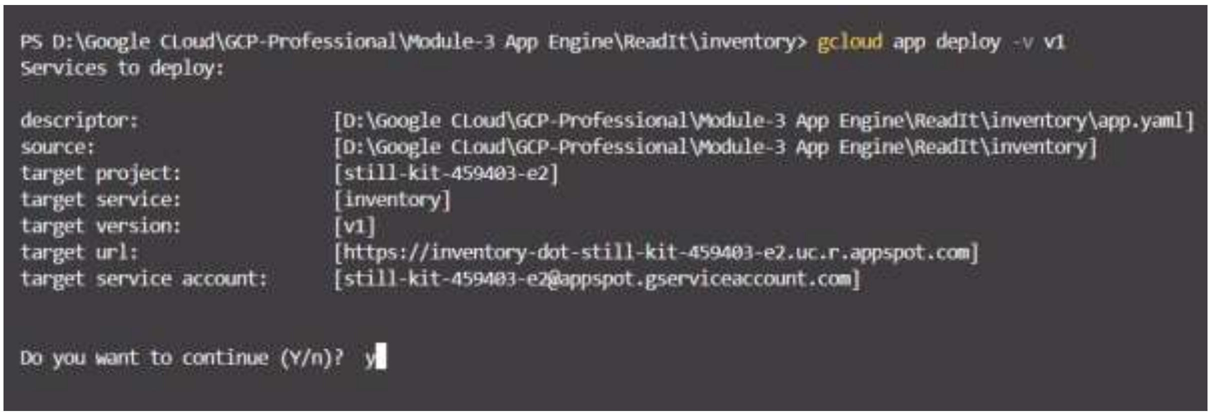
16) Now you need to create an app.yaml file inside your inventory folder and use the code given below.

```
runtime: aspnetcore
env: flex

runtime_config:
 operating_system: "ubuntu22"

service: "inventory"

automatic_scaling:
 min_num_instances: 1
 max_num_instances: 5
```



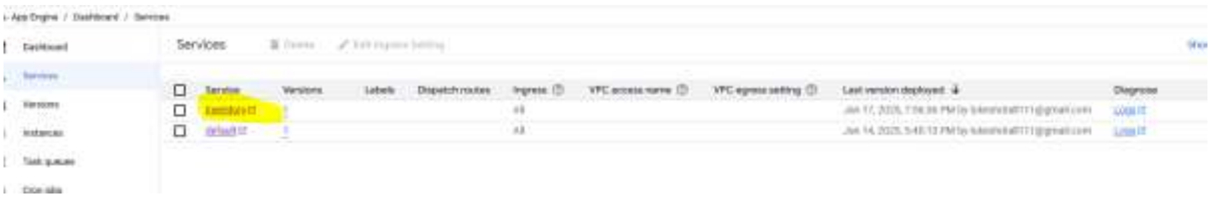
```
runtime: aspnetcore
env: flex

runtime_config:
  operating_system: "ubuntu22"

service: "inventory"

automatic_scaling:
  min_num_instances: 1
  max_num_instances: 5
```

17) Run the **gcloud app deploy -v v1** (here this name v1 can be anything, as we want to keep it for the version name) command at VSCode



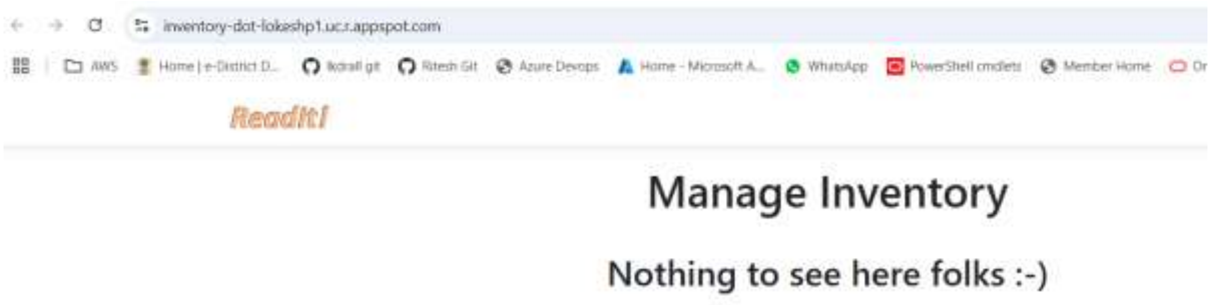
```
PS D:\Google Cloud\GCP-Professional\Module-3 App Engine\ReadIt\inventory> gcloud app deploy -v v1
Services to deploy:

descriptor:                [D:\Google Cloud\GCP-Professional\Module-3 App Engine\ReadIt\inventory\app.yaml]
source:                    [D:\Google Cloud\GCP-Professional\Module-3 App Engine\ReadIt\inventory]
target project:            [still-kit-459403-e2]
target service:            [inventory]
target version:            [v1]
target url:                [https://inventory-dot-still-kit-459403-e2.uc.r.appspot.com]
target service account:    [still-kit-459403-e2@appspot.gserviceaccount.com]


Do you want to continue (Y/n)?  y
```

18) Now go to GCP and under services, there will be a new entry named inventory (or the same name as mentioned at “service” in app.yaml file



Service	Versions	Labels	Dispatch routes	Ingress	VPC access name	VPC egress setting	Last version deployed	Diagnose
inventory	1			All			Jan 17, 2025, 7:56:38 PM by [illegible]@gmail.com	Logs
default	1			All			Jan 14, 2025, 5:40:13 PM by [illegible]@gmail.com	Logs

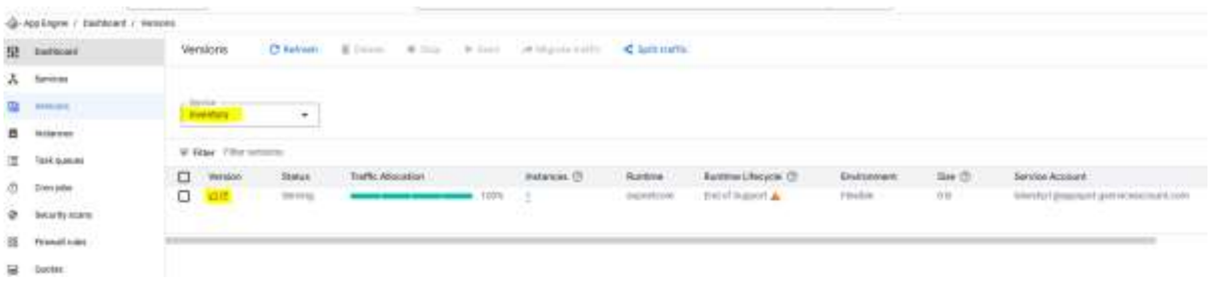
19) Now, click the “inventory” service name, and it should open the inventory page



Manage Inventory

Nothing to see here folks :-)

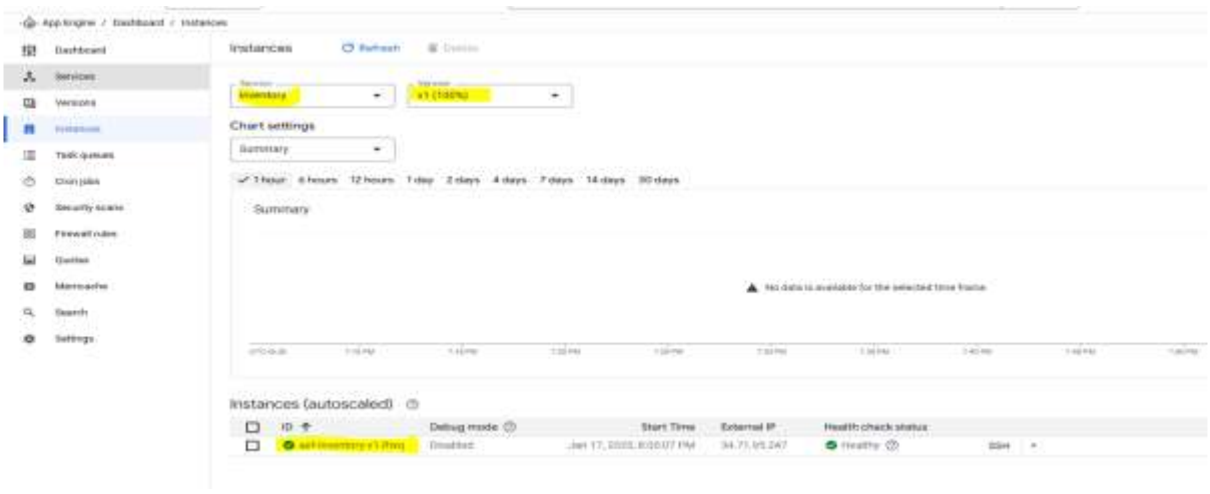
20) Go to versions, select inventory, and you will see the version name (v1) as we put in in app.yaml file while running the app deploy command, also see the environment as Flexible


Versions

Version	Status	Traffic Allocation	Instances	Runtime	Runtime Lifecycle	Environment	Size	Service Account
v1	Serving	100%	1	aspnetcore	End of Support	Flexible	0 B	[illegible]

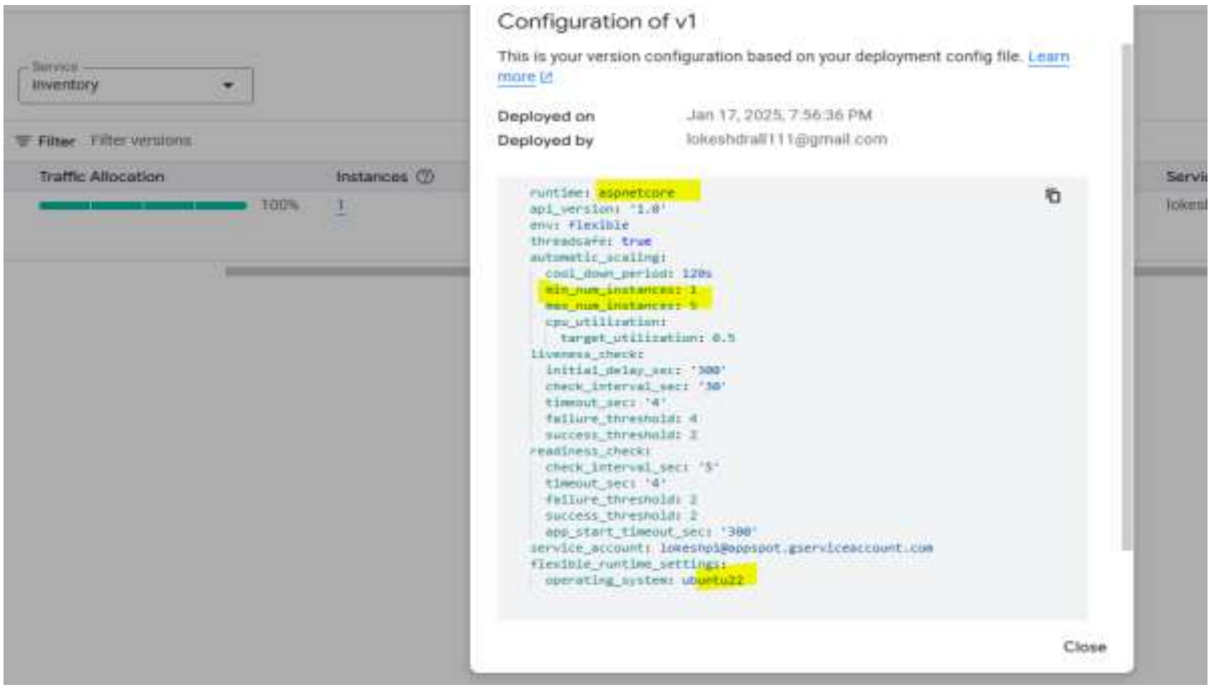
21) Click at the instance

22) This will show all details related to the infrastructure running underneath


Instances (autoscaled)

ID	Debug mode	Start Time	External IP	Health check status
aef-inventory-v1-[illegible]	Disabled	Jan 17, 2025, [illegible] PM	34.71.95.247	Healthy

23) Also verify version data, it would be similar to what we gave in app.yaml file


Configuration of v1

This is your version configuration based on your deployment config file. Learn more

Deployed on	Jan 17, 2025, 7:56:36 PM
Deployed by	lokeshdrall111@gmail.com

```
runtime: aspnetcore
api_version: '1.0'
env: flexible
threadsafe: true
automatic_scaling:
  cool_down_period: 120s
  min_num_instances: 1
  max_num_instances: 5
  cpu_utilization:
    target_utilization: 0.5
liveness_check:
  initial_delay_sec: '300'
  check_interval_sec: '30'
  timeout_sec: '4'
  failure_threshold: 4
  success_threshold: 2
readiness_check:
  check_interval_sec: '5'
  timeout_sec: '4'
  failure_threshold: 2
  success_threshold: 2
  app_start_timeout_sec: '300'
service_account: lokeshp1@appspot.gserviceaccount.com
flexible_runtime_settings:
  operating_system: ubuntu22
```