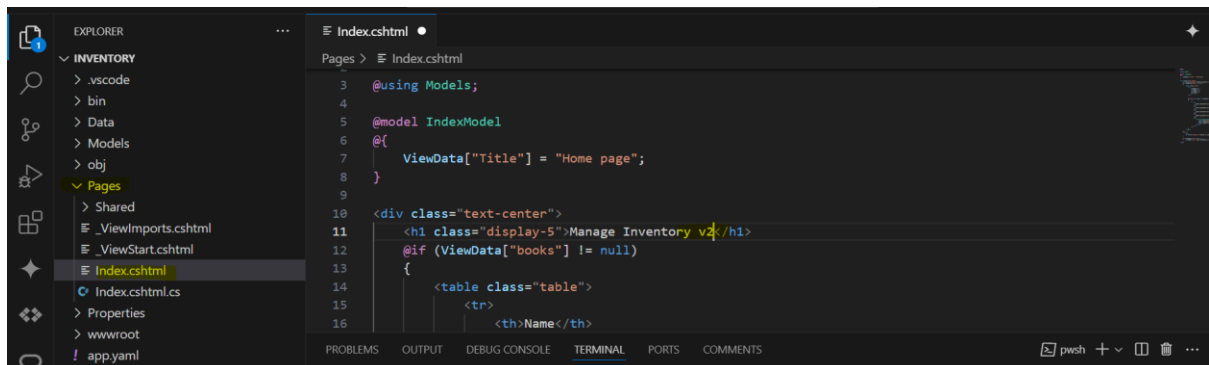


In this lab of GCP App Engine, we will create a new version of our app, and we will also test traffic splitting between versions

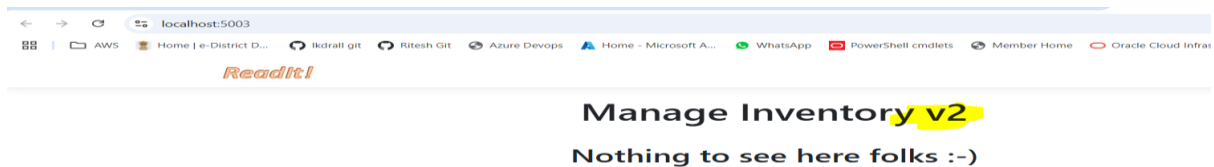
Creating Version: -

1. Open the same Inventory folder in VS Code that we used in the previous lab
2. Go to the index.cshhtml file under the pages folder and add V2 in front of inventory at line 11.



```
3 @using Models;
4
5 @model IndexModel
6 @{
7     ViewData["Title"] = "Home page";
8 }
9
10 <div class="text-center">
11     <h1 class="display-5">Manage Inventory v2</h1>
12     @if (ViewData["books"] != null)
13     {
14         <table class="table">
15             <tr>
16                 <th>Name</th>
```

3. Hit F5 to test the code locally.
4. On completion, the webpage should show inventory v2 now, which is to distinguish the two different versions of our app.



5. Go to the GCP console, and check Inventory shows only one version

App Engine / Dashboard / Services

Dashboard

Services

Versions

Instances

Task queues

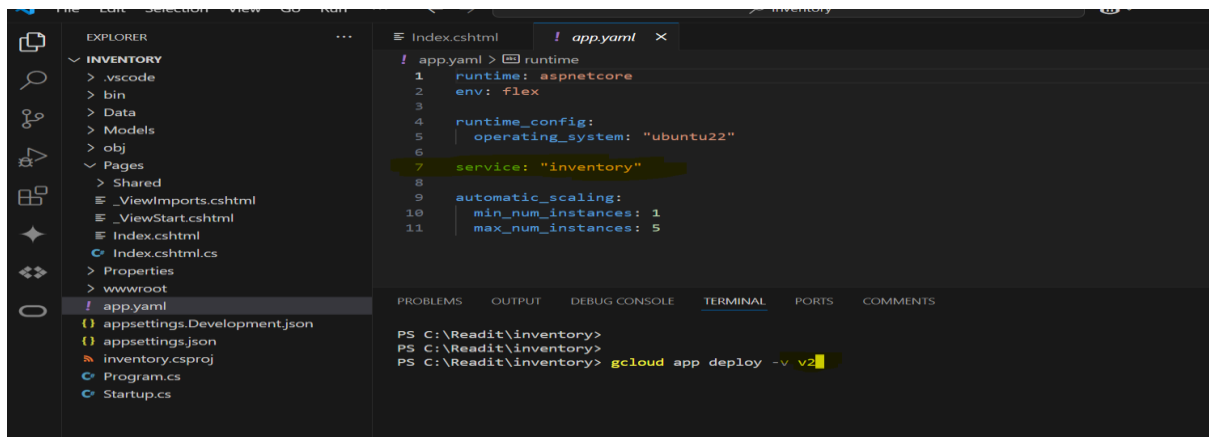
Services

Delete

Edit Ingress Setting

<input type="checkbox"/>	Service	Versions	Labels	Dispatch routes	Ingress ?	VPC access name ?
<input type="checkbox"/>	inventory	1			All	
<input type="checkbox"/>	default	1			All	

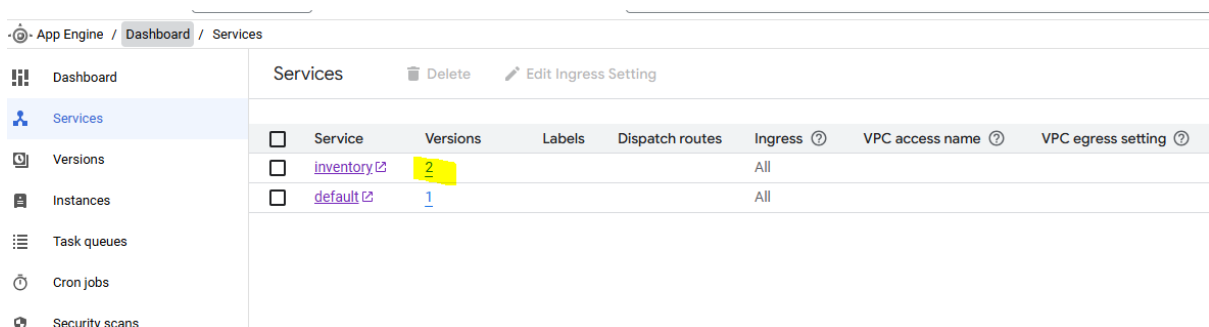
6. Run the command **gcloud app deploy -v v2**
(Make sure the service name in app.yaml is the same as we used in the previous lab, only change the version to v2)



```
! app.yaml > runtime
1 runtime: aspnetcore
2 env: flex
3
4 runtime_config:
5   operating_system: "ubuntu22"
6
7 service: "inventory"
8
9 automatic_scaling:
10   min_num_instances: 1
11   max_num_instances: 5
```

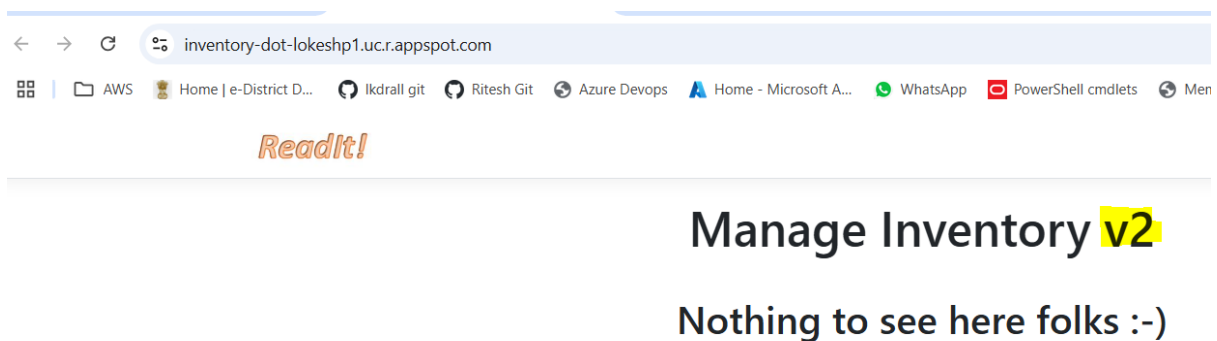
```
PS C:\Readit\inventory>
PS C:\Readit\inventory>
PS C:\Readit\inventory> gcloud app deploy -v v2
```

7. After completion, at GCP now it should now show 2 versions

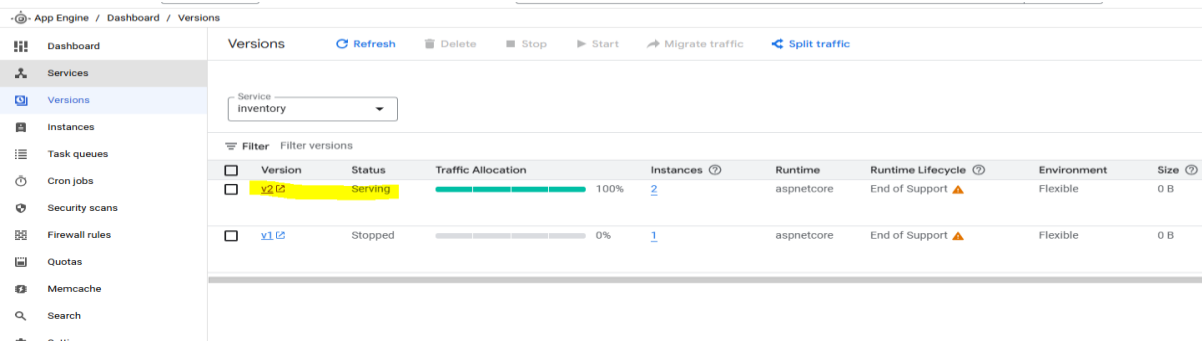


Service	Versions	Labels	Dispatch routes	Ingress	VPC access name	VPC egress setting
inventory	2			All		
default	1			All		

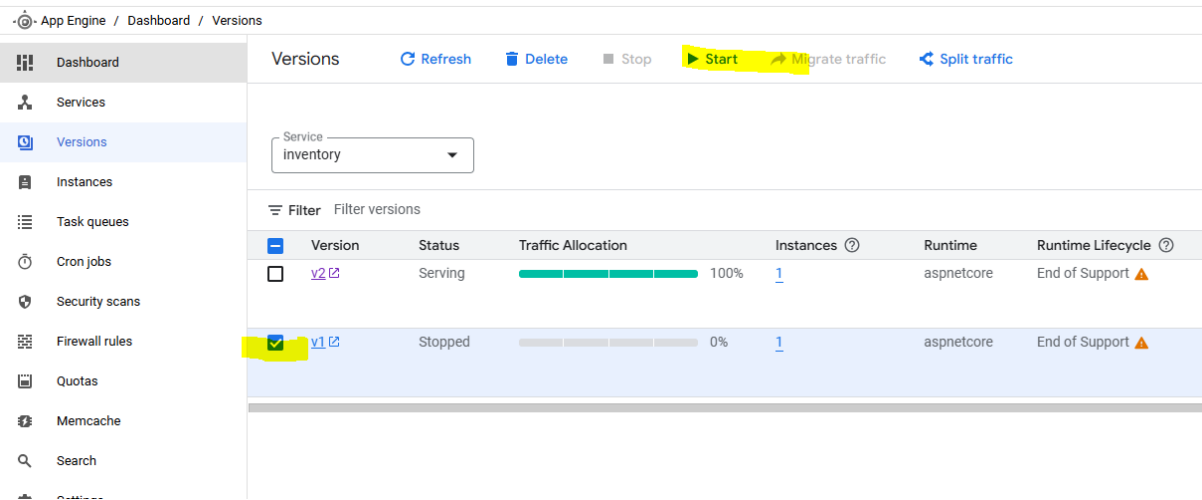
8. Click on the service name “inventory”, now it should open the v2 page. This also shows that by default, traffic is always moved to the latest version.



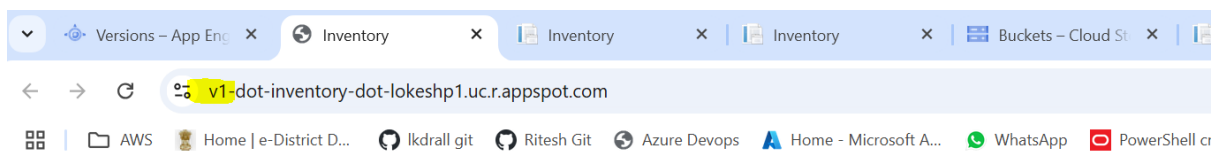
9. Go to versions, and there it will show that the status of v2 is serving and v1 is stopped



10. Select v1 and click Start to start the version for serving



11. Wait for some time. And then click v1, it will open the version 1 webpage (without v2 written after inventory). Also note that the URL also shows the v1, but not the app name like previously, which was “https://inventory-dot-lokeshp1.uc.r.appspot.com/”. This is because only the default version shows the correct app URL.



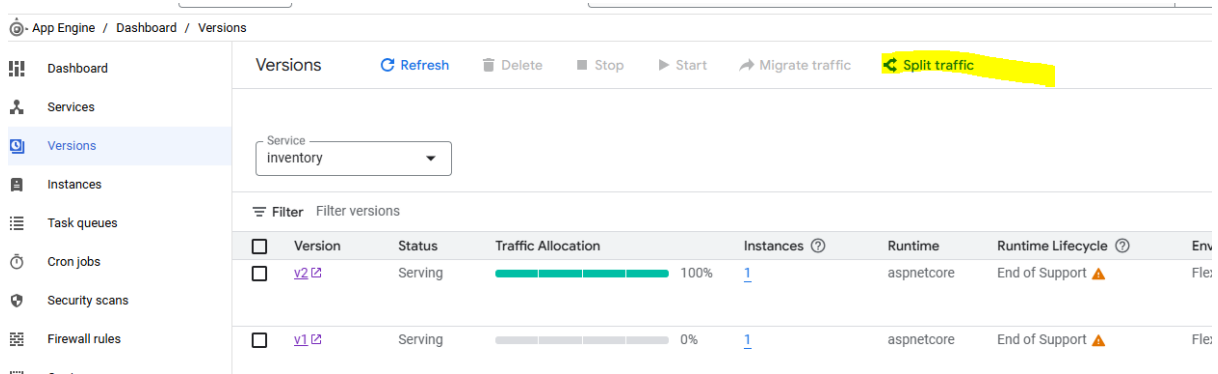
ReadIt!

Manage Inventory

Nothing to see here folks :-)

Traffic splitting:-

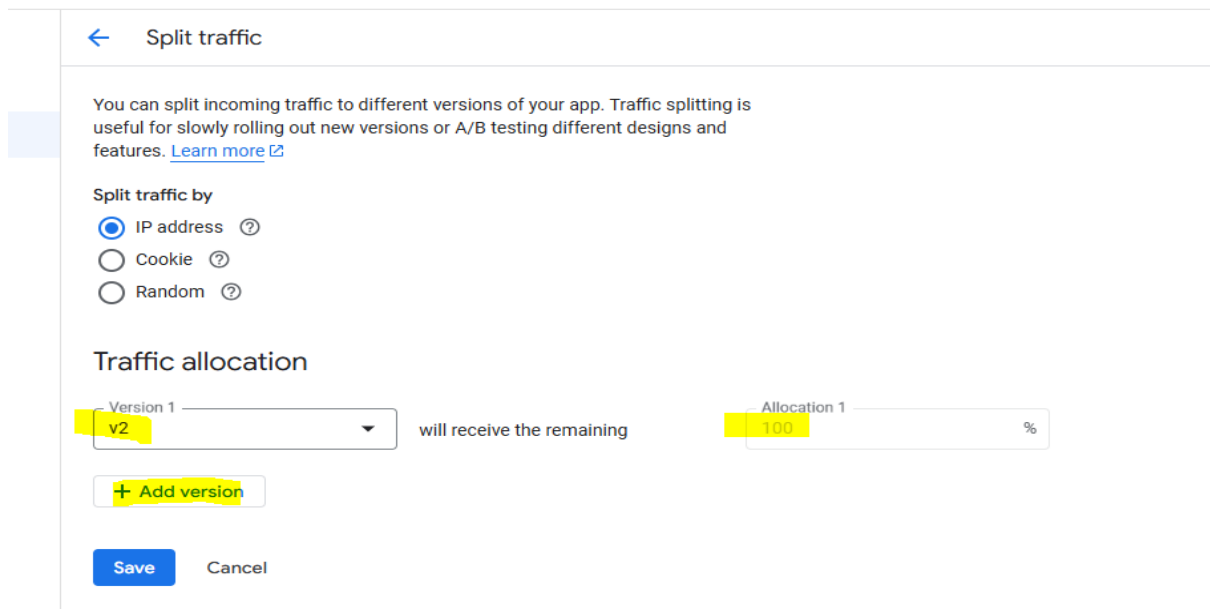
1. Click split traffic



The screenshot shows the Google Cloud App Engine 'Versions' page for a service named 'inventory'. The left sidebar contains navigation links: Dashboard, Services, Versions (selected), Instances, Task queues, Cron jobs, Security scans, and Firewall rules. The main content area has a 'Versions' header with buttons for Refresh, Delete, Stop, Start, Migrate traffic, and Split traffic (highlighted in yellow). Below the header is a 'Service' dropdown set to 'inventory' and a 'Filter' section. A table lists two versions: v2 and v1. Version v2 is 'Serving' with 100% traffic allocation and 1 instance. Version v1 is 'Serving' with 0% traffic allocation and 1 instance. Both versions use the 'aspnetcore' runtime and are at 'End of Support'.

Version	Status	Traffic Allocation	Instances	Runtime	Runtime Lifecycle	Env
v2	Serving	100%	1	aspnetcore	End of Support	Fl:
v1	Serving	0%	1	aspnetcore	End of Support	Fl:

2. It shows 100% allocation to v2, click “Add version”



The screenshot shows the 'Split traffic' configuration dialog. It includes a back arrow, a title 'Split traffic', and an explanatory text: 'You can split incoming traffic to different versions of your app. Traffic splitting is useful for slowly rolling out new versions or A/B testing different designs and features. [Learn more](#)'. Under 'Split traffic by', the 'IP address' option is selected. The 'Traffic allocation' section shows 'Version 1' set to 'v2' and 'Allocation 1' set to '100%'. A '+ Add version' button is highlighted in yellow. At the bottom are 'Save' and 'Cancel' buttons.

3. Select v1 and split to 50-50. Select “Random” split traffic for lab purposes

4. Also note, we can use an IP address (this will route traffic from a specific IP address to the same version every time and if we use Cookie, then that will save the cookie in the user's browser and will use the version accordingly in the future

← Split traffic

You can split incoming traffic to different versions of your app. Traffic splitting is useful for slowly rolling out new versions or A/B testing different designs and features. [Learn more](#)

Split traffic by

- ☐ IP address ?
- ☐ Cookie ?
- ☒ Random ?

Traffic allocation

Version 1 v2 will receive the remaining

Allocation 1 50 %

v1 50 %

[+ Add version](#)

[Save](#) [Cancel](#)

5. Now go back to the Services page and open url by clicking the service name, this will open the app page again. Try refreshing the page again and again, and it will sometimes open v1 and sometimes v2, but the URL will remain the same.

App Engine / Dashboard / Services

Dashboard	Services	Delete	Edit Ingress Setting
Services			
Versions			
Instances			
Task queues			

Service	Versions	Labels	Dispatch routes	Ingress ?	VPC access
<input type="checkbox"/> inventory	2			All	
<input type="checkbox"/> default	1			All	

inventory-dot-lokeshp1.uc.r.appspot.com

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

ReadIt!

Manage Inventory

Nothing to see here folks :-)

ReadIt!

Manage Inventory v2

Nothing to see here folks :-)

6. Once the lab is completed, make sure to delete the traffic splitting by removing v1.

App Engine / Dashboard / Versions / Traffic splitting

Dashboard

Services

Versions

Instances

Task queues

Cron jobs

Security scans

Firewall rules

Quotas

Memcache

Search

Settings

Split traffic

You can split incoming traffic to different versions of your app. Traffic splitting is useful for slowly rolling out new versions or A/B testing different designs and features. [Learn more](#)

Split traffic by

☐ IP address ?

☐ Cookie ?

☒ Random ?

Traffic allocation

Version 1

v2

will receive the remaining

Allocation 1

50

%

Delete item

Allocation 2

50

%

Delete item

+ Add version

Save

Cancel