

# Week#10 Labs

*Neha Agrawal*

## Table of Contents

<b>Dataprocc, Dataflow .....</b>	<b>2</b>
Dataprocc Lab #1(pi).....	2
Calculating pi .....	2
Code .....	2
Dataprocc Setup.....	2
Create compute engine cluster.....	2
Run computation.....	2
Scale cluster.....	2
Run computation again.....	2
Clean up.....	2
Dataflow lab #1(Java package popularity) .....	2
Setup .....	2
Beam code.....	2
Run pipeline locally .....	3
Dataflow lab #2(word count) .....	4
Run code locally.....	4
Setup for cloud dataflow .....	4
Service account setup .....	4
Run code using dataflow runner.....	4
Clean up.....	5
<b>CDN .....</b>	<b>6</b>
Part 1: Networks and VMs .....	6
Deployment specification.....	6
Network deployment specification.....	6
Subnetwork deployment specification .....	6
Vm deployment specification .....	6
deployment .....	6
firewall deployment specification .....	8
update deployment.....	8

latency measurements .....	8
Part 2: scaling via instance group and load balancing.....	8
Firewall rule for HTTP .....	8
Instance templates.....	8
Health check .....	8
Managed instance group (Europe-west1-mig) .....	8
Managed instance group (us-east1-mig) .....	8
Test groups .....	8
HTTP load balancer.....	9
HTTP load balancer .....	9
Test load balancer .....	9
Siege!.....	10
Clean up.....	10

## 10.1g: Dataproc, Dataflow

1. Dataproc Lab #1 ( $\pi$ )
2. Calculating  $\pi$
3. Code
4. Dataproc setup
5. Create Compute Engine cluster
6. Run computation

**For your lab notebook:**

- **How long did the job take to execute?**  
~66sec
- **Examine `output.txt` and show the estimate of  $\pi$  calculated.**  
Pi is roughly 3.1417039514170395

### 7. Scale cluster

### 8. Run computation again

- **How long did the job take to execute? How much faster did it take?**  
~11sec
- **Examine `output2.txt` and show the estimate of  $\pi$  calculated.**  
Pi is roughly 3.1415359514153596

### 9. Clean up

### 10. Dataflow Lab #1 (Java package popularity)

### 11. Setup

### 12. Beam code

**Answer the following questions for your lab notebook.**

- **Where is the input taken from by default?**

```
../javahelp/src/main/java/com/google/cloud/training/dataanalyst/javahelp/
```

- **Where does the output go by default?**

```
/tmp/output
```

- **Examine both the `getPackages()` function and the `splitPackageName()` function. What operation does the `'PackageUse()'` transform implement?**

`PackageUse()` will call `getPackages()` to get the packages and pass it as parameter to the `splitPackageName()` which will return package names.

Example:

1. `PackageUse(line, 'import')` where `line = import java.util.Scanner` is passed as a parameter;
2. `getPackages()` will strip off the "import" and ";" so `packageName = java.util.Scanner`
3. Now it will call `splitPackageName()` which will return a list containing `java`, `java.util`, `java.util.Scanner` to the calling function.

- **Look up Beam's `CombinePerKey`. What operation does the `TotalUse` operation implement?**

`CombinePerKey` accepts a function that takes a list of values as an input, and combines them for each key.

`TotalUse` corresponds to Reduce operation or Shuffle-Reduce operation

The operations in the pipeline mimic a Map-Reduce pattern, demonstrating Beam's ability to support it.

Answer the following question for your lab notebook.

- **Which operations correspond to a "Map"?**  
`PackageUse`
- **Which operation corresponds to a "Shuffle-Reduce"?**  
`TotalUse`
- **Which operation corresponds to a "Reduce"?**  
`TotalUse`

### 13. Run pipeline locally

- **Take a screenshot of its contents**

```
n (cloud-f20-neha-agrawal-agrawal)$ cat /tmp/output-00000-of-00001
[('org', 45), ('org.apache', 44), ('org.apache.beam', 44), ('org.apache.beam.sdk', 43), ('org.apache.beam.sdk.transforms', 16)]
```

- **Explain what the data in this output file corresponds to based on your understanding of the program.**

Each package name returned by the list of `splitPackageName()` is a key here and the value corresponding to it shows the total count.

## 14. Dataflow Lab #2 (Word count)

- **What are the names of the stages in the pipeline?**

Split

PairWithOne

GroupAndSum

- **Describe what each stage does.**

Split: The split will take each line and split into words as strings

PairWithOne: It will create a map of each word where it will keep its count as 1

GroupAndSum: As the name suggests, this will take the output from previous step and aggregate them for each key. Thus as an output we get the number of occurrences of each word.

## 15. Run code locally

- Use `wc` with an appropriate flag to determine the number of unique words in King Lear.

4784

```
agrawal@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-f20-neha-  
-agrawal-agrawal)$ wc -l outputs-00000-of-00001  
4784 outputs-00000-of-00001
```

- Use `sort` with appropriate flags to perform a *numeric* sort on the *key field* containing the count for each word in *descending* order. Pipe the output into `head` to show the top 3 words in King Lear and the number of times they appear

`sort -k2 -n -r outputs-00000-of-00001 | head -n 3`

```
agrawal@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-f20-neha-  
-agrawal-agrawal)$ sort -k2 -n -r outputs-00000-of-00001 | head -n 3  
the: 786  
I: 622  
and: 594
```

- Use the previous method to show the top 3 words in King Lear, case-insensitive, and the number of times they appear.

```
(env) agrawal@cloudshell:~/training-data-analyst/courses/machine_learning/deepdive/04_features/dataflow/python (cloud-f20-neha-  
-agrawal-agrawal)$ sort -k2 -n -r outputs-00000-of-00001 | head -n 3  
the: 908  
and: 738  
i: 622
```

## 16. Setup for Cloud Dataflow

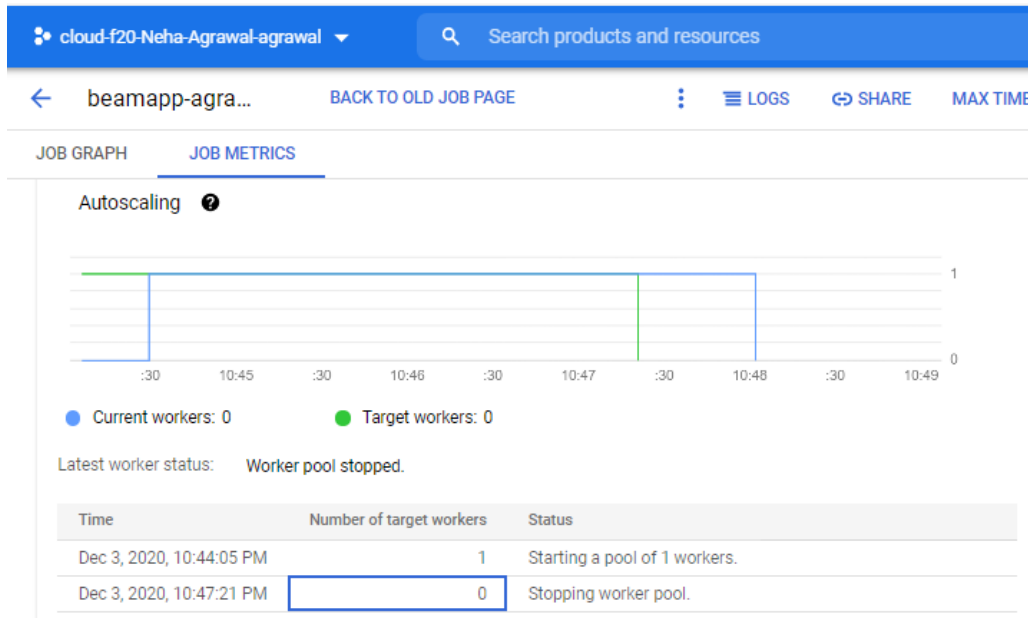
## 17. Service account setup

## 18. Run code using Dataflow runner

- The part of the job graph that has taken the longest time to complete.

Write

- The autoscaling graph showing when the worker was created and stopped.



- Examine the output directory in Cloud Storage. How many files has the final write stage in the pipeline created?

It has created 3 files in the result directory:

The screenshot shows the 'Bucket details' page for the bucket 'cloud-f20-neha-agrawal-agrawal'. The 'OBJECTS' tab is selected, showing a list of objects in the 'results' directory. The table lists three files, each named 'outp', with sizes of 15.9 KB, 15.8 KB, and 16.1 KB. All files are of type 'text/plain' and were created on Dec 3, 2020, at 10:48:05 PM, 10:48:15 PM, and 10:48:25 PM respectively. They are stored in the 'Standard' storage class and are not public.

Name	Size	Type	Created time	Storage class	Last modified	Public access	Encryption	Retention expiration date	Holds
outp	15.9 KB	text/plain	Dec 3, 2020, 10:48:05 PM	Standard	Dec 3, 2020, 10:48:05 PM	Not public	Google-managed key	—	None
outp	15.8 KB	text/plain	Dec 3, 2020, 10:48:15 PM	Standard	Dec 3, 2020, 10:48:15 PM	Not public	Google-managed key	—	None
outp	16.1 KB	text/plain	Dec 3, 2020, 10:48:25 PM	Standard	Dec 3, 2020, 10:48:25 PM	Not public	Google-managed key	—	None

## 19. Clean up

## 10.2g: CDN

1. Part 1: Networks and VMs
2. Deployment specification
3. Network deployment specification
4. Subnetwork deployment specification
5. Virtual machine deployment specification
6. Deployment

- Take a screenshot of the output to include in your lab notebook. How many networks, subnetworks, and VM instances have been created?

```
(env) agrawal@cloudshell:~/networking101 (cloud-f20-neha-agrawal-agrawal)$ gcloud deployment-manager deployments create networking101 --config networking-lab.yaml
The fingerprint of the deployment is b'XZH1cRMERG9Q2J4xyDNwHg=='
Waiting for create [operation-1607197964912-5b5bcefc84927-4be29ff6-199800a8]...done.
Create operation operation-1607197964912-5b5bcefc84927-4be29ff6-199800a8 completed successfully.
NAME          TYPE                STATE    ERRORS  INTENT
asia-east1    compute.v1.subnetwork COMPLETED []
asia1-vm      compute.v1.instance  COMPLETED []
e1-vm         compute.v1.instance  COMPLETED []
eul-vm        compute.v1.instance  COMPLETED []
europe-west1  compute.v1.subnetwork COMPLETED []
networking101 compute.v1.network    COMPLETED []
us-east1      compute.v1.subnetwork COMPLETED []
us-west1-s1   compute.v1.subnetwork COMPLETED []
us-west1-s2   compute.v1.subnetwork COMPLETED []
w1-vm         compute.v1.instance  COMPLETED []
w2-vm         compute.v1.instance  COMPLETED []
```

Networks = 1

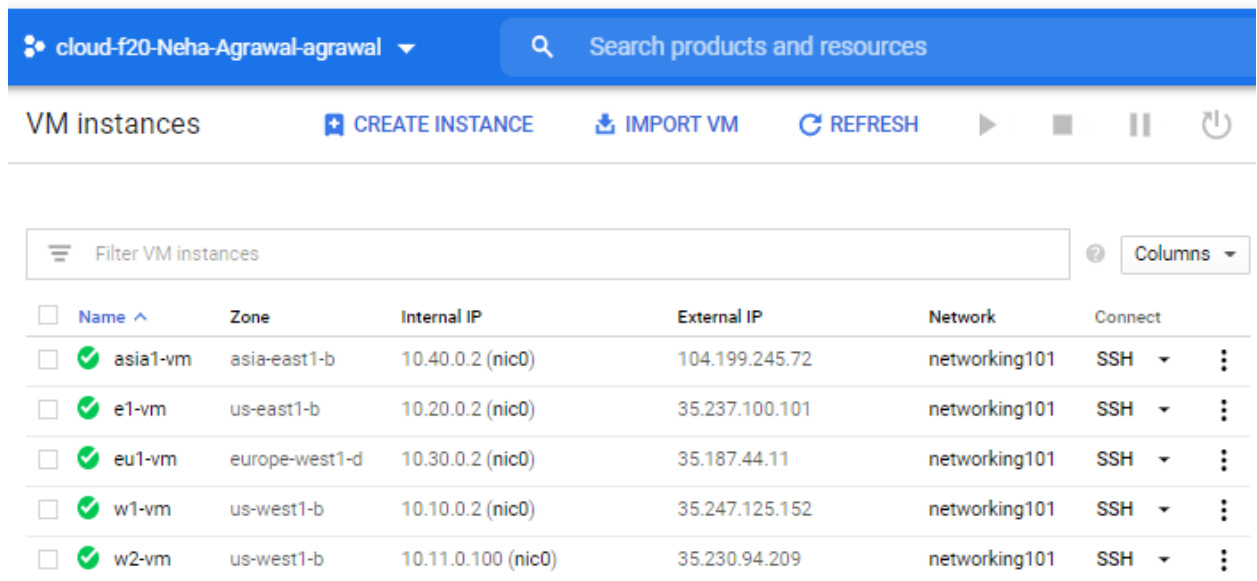
Subnetworks = 5

VM instances = 5

- Visit the web console for VPC network and show the network and the subnetworks that have been created. Validate that it has created the infrastructure in the initial figure. Note the number of firewall rules that are in the deployment.

cloud-f20-Neha-Agrawal-agrawal									
VPC networks									
Name	Region	Subnets	MTU	Mode	IP address ranges	Gateways	Firewall Rules	Global dynamic routing	Flow logs
default		24	1460	Auto			7	Off	
networking101		5	1460	Custom			0	Off	
	europe-west1	europe-west1			10.30.0.0/16	10.30.0.1			Off
	us-west1	us-west1-s1			10.10.0.0/16	10.10.0.1			Off
	us-west1	us-west1-s2			10.11.0.0/16	10.11.0.1			Off
	asia-east1	asia-east1			10.40.0.0/16	10.40.0.1			Off
	us-east1	us-east1			10.20.0.0/16	10.20.0.1			Off

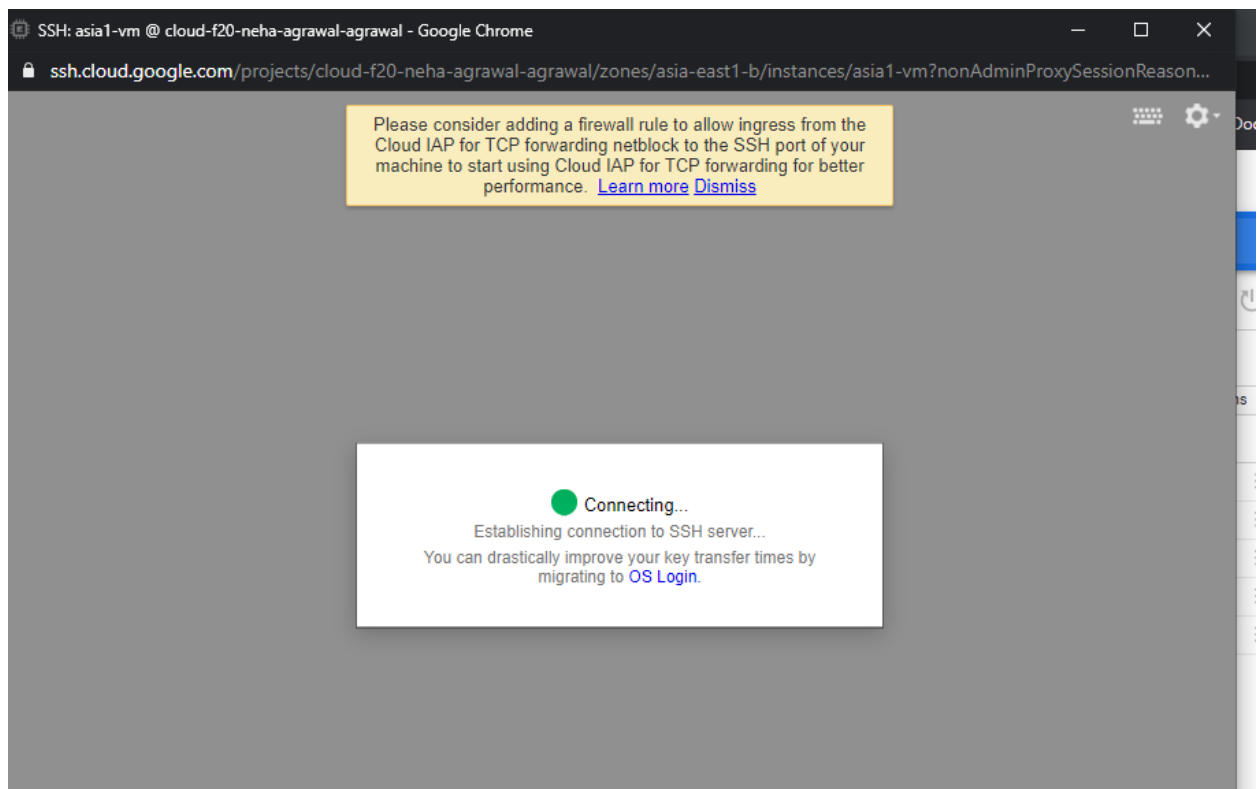
- Visit the web console for Compute Engine and show all VMs that have been created, their internal IP addresses and the subnetworks they have been instantiated on. Validate that it has created the infrastructure shown in the initial figure.



Name	Zone	Internal IP	External IP	Network	Connect
asia1-vm	asia-east1-b	10.40.0.2 (nic0)	104.199.245.72	networking101	SSH
e1-vm	us-east1-b	10.20.0.2 (nic0)	35.237.100.101	networking101	SSH
eu1-vm	europa-west1-d	10.30.0.2 (nic0)	35.187.44.11	networking101	SSH
w1-vm	us-west1-b	10.10.0.2 (nic0)	35.247.125.152	networking101	SSH
w2-vm	us-west1-b	10.11.0.100 (nic0)	35.230.94.209	networking101	SSH

- Click on the `ssh` button for one of the VMs and attempt to connect. Did it succeed? Take a screenshot and include it in your lab notebook.

It failed!





## 7. Firewall deployment specification

## 8. Update deployment

## 9. Latency measurements

Location pair	Ideal latency	Measured latency
us-west1 us-east1	37.12 ms	63.5ms
us-west1 europe-west1	80.01 ms	137ms
us-west1 asia-east1	99.59 ms	113ms
us-east1 europe-west1	67.51 ms	92.1ms
us-east1 asia-east1	131.44 ms	187ms
europe-west1 asia-east1	95.61 ms	246ms

## 10. Part 2: Scaling via Instance Groups and Load Balancing

## 11. Firewall rule for HTTP

## 12. Instance templates

## 13. Health check

## 14. Managed Instance Group (europe-west1-mig)

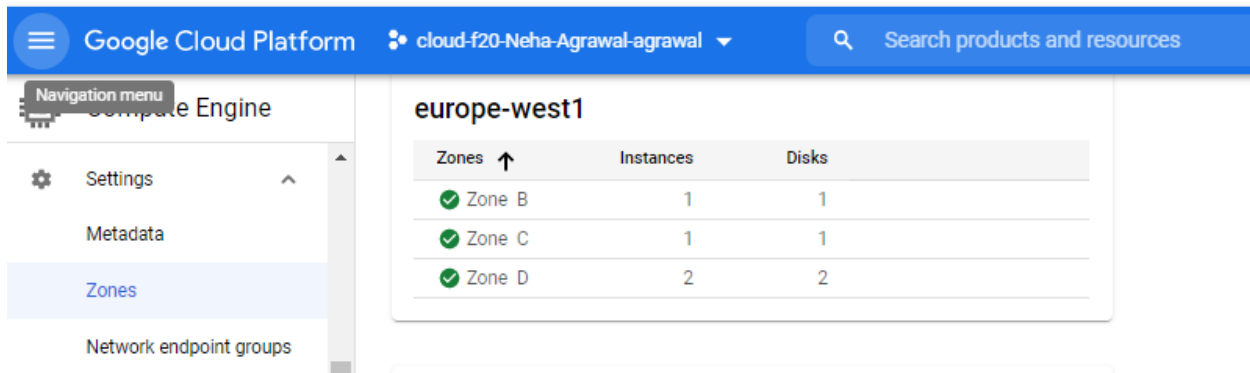
## 15. Managed Instance Group (us-east1-mig)

## 16. Test groups

- Are the instances in the same availability zone or in different ones?

*The instances are in the different availability zones. One is in zone B, other is in zone C and the third one is in Zone D*

- List all availability zones that your servers show up in for your lab notebook.



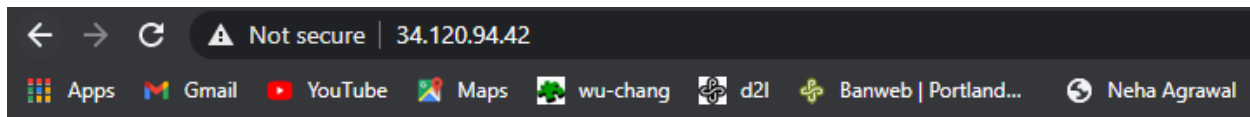
europe-west1		
Zones ↑	Instances	Disks
✓ Zone B	1	1
✓ Zone C	1	1
✓ Zone D	2	2

17. HTTP load balancer

18. HTTP load balancer

19. Test load balancer

Show a screenshot of the page that is returned. If you get an error, you may need to wait several minutes for the load balancer to finish deploying. Which region does the server handling your request reside in?



## Networking 101 Lab

### Client IP

Your IP address : 130.211.1.13

### Hostname

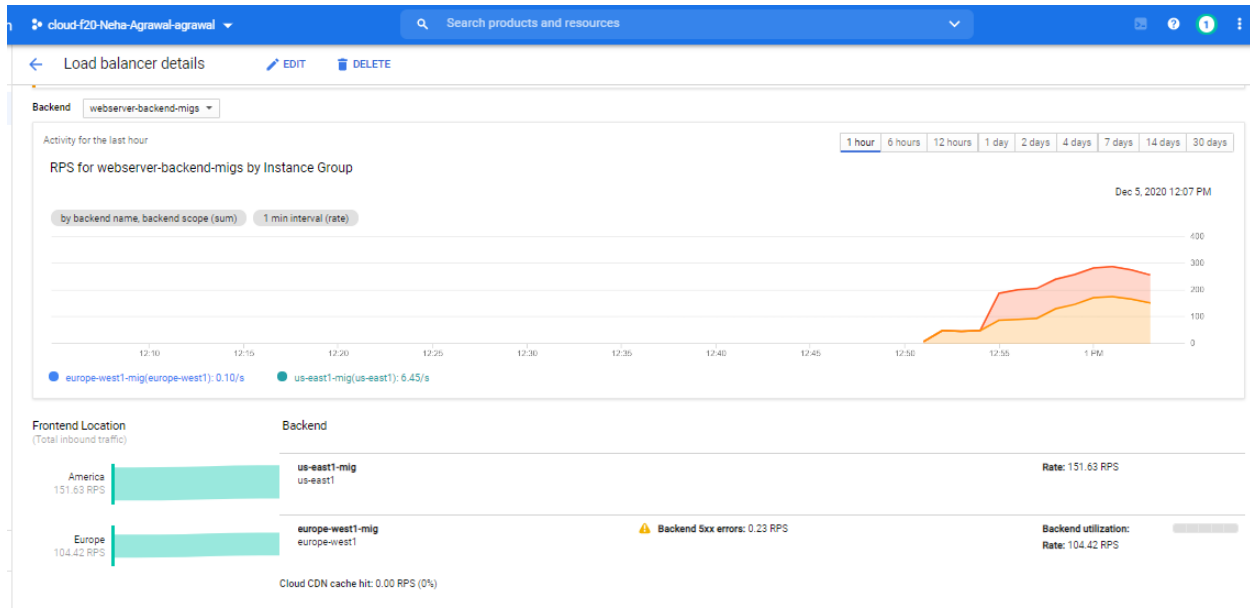
Server Hostname: us-east1-mig-0z98

### Server Location

Region and Zone: us-east1-d

*Region: us-east1*

## 20. Siege!



*I couldn't get the graph to display much on the monitoring tab till 10 mins, neither the instances scaled up. And then I hit the refresh and above is the plot I saw. So, I do not have intermittent screenshots to show all the 5 instances up.*

## 21. Clean-up