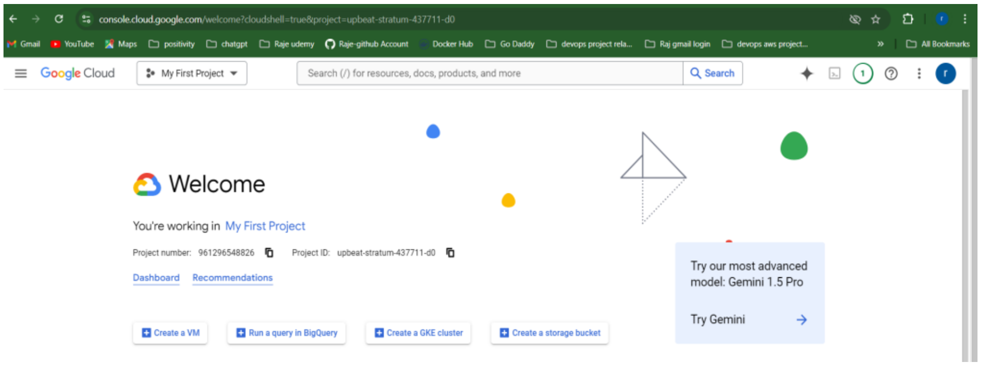
## Graded Assignment on MERN application using Kubernetes

Step 1: Set Up Google Cloud Shell Environment: -

- Go to the GCP console: https://console.cloud.google.com



Welcome

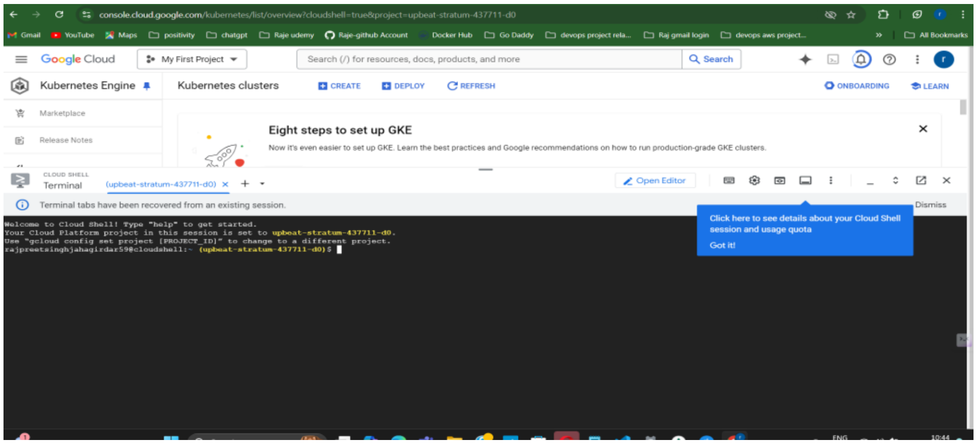
You're working in My First Project

Project number: 961296548826 Project ID: upbeat-stratum-437711-d0

Try our most advanced model: Gemini 1.5 Pro

Try Gemini

- Click on the terminal icon (Google Cloud Shell) at the top-right corner:-



Eight steps to set up GKE

Now it's even easier to set up GKE. Learn the best practices and Google recommendations on how to run production-grade GKE clusters.

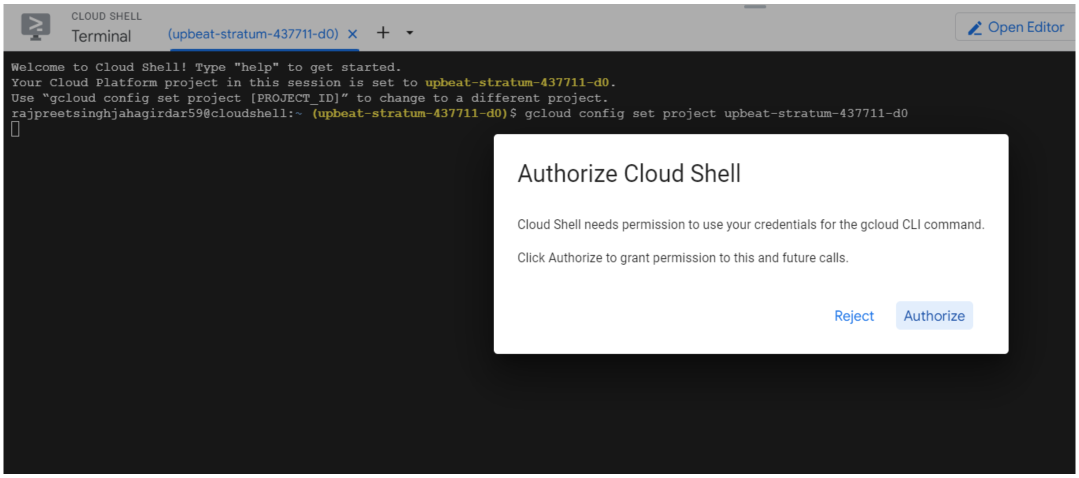
Terminal tabs have been recovered from an existing session.

Click here to see details about your Cloud Shell session and usage quota

```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to upbeat-stratum-437711-d0.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
rajpreetsinghjahagirdar59@cloudshell:~ (upbeat-stratum-437711-d0)$
```

**2. Set the GCP project:-**

gcloud config set project [PROJECT_ID]

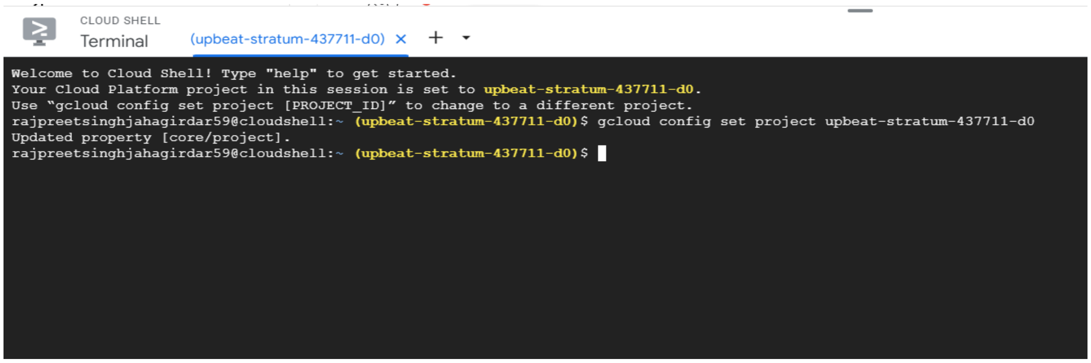


```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to upbeat-stratum-437711-d0.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
rajpreetsinghjahagirdar59@cloudshell:~ (upbeat-stratum-437711-d0)$ gcloud config set project upbeat-stratum-437711-d0
```

Authorize Cloud Shell

Cloud Shell needs permission to use your credentials for the gcloud CLI command.

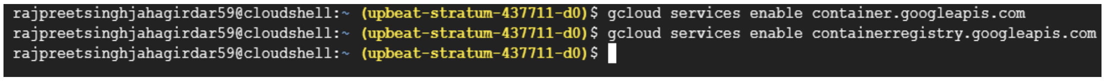
Click Authorize to grant permission to this and future calls.



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to upbeat-stratum-437711-d0.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
rajpreetsinghjahagirdar59@cloudshell:~ (upbeat-stratum-437711-d0)$ gcloud config set project upbeat-stratum-437711-d0
Updated property [core/project].
rajpreetsinghjahagirdar59@cloudshell:~ (upbeat-stratum-437711-d0)$
```

3. **Enable required APIs**: Enable the Google Kubernetes Engine (GKE) and Google Container Registry (GCR) APIs:
gcloud services enable container.googleapis.com

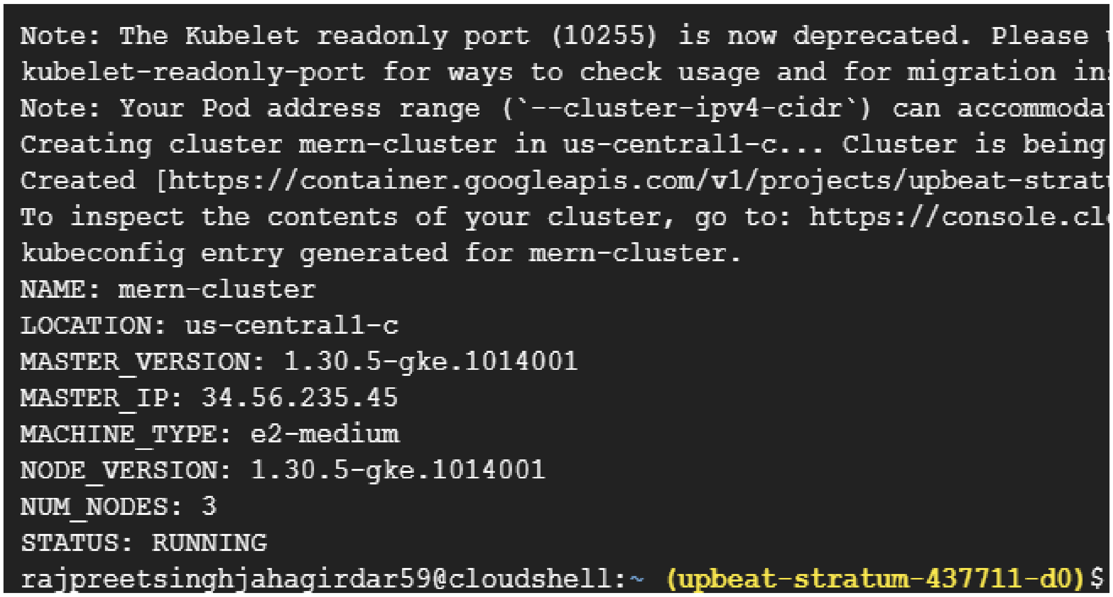
gcloud services enable containerregistry.googleapis.com



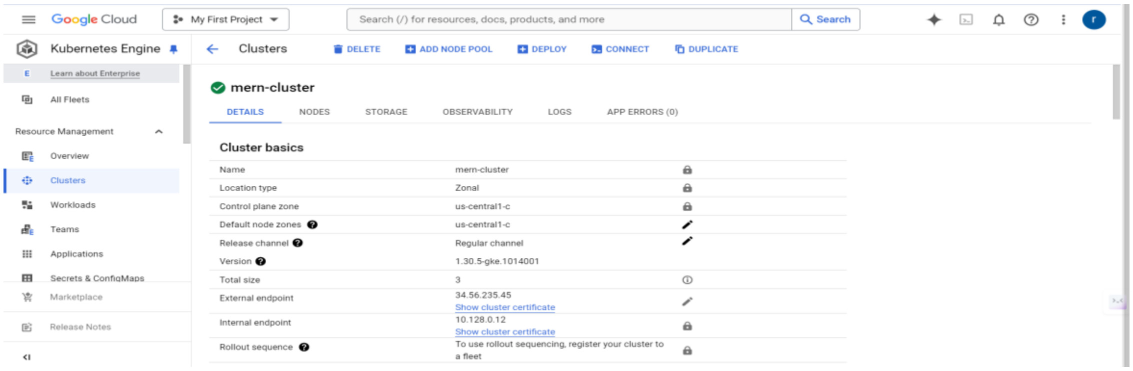
```
rajpreetsinghjahagirdar59@cloudshell:~ (upbeat-stratum-437711-d0)$ gcloud services enable container.googleapis.com
rajpreetsinghjahagirdar59@cloudshell:~ (upbeat-stratum-437711-d0)$ gcloud services enable containerregistry.googleapis.com
rajpreetsinghjahagirdar59@cloudshell:~ (upbeat-stratum-437711-d0)$
```

Step 2: Create a GKE Cluster:-

**Create the GKE cluster**: Run the following command in Cloud Shell to create a 3-node GKE cluster in the specified zone: -
gcloud container clusters create mern-cluster --zone us-central1-a --num-nodes=3



```
Note: The Kubelet readonly port (10255) is now deprecated. Please 
kubelet-readonly-port for ways to check usage and for migration in
Note: Your Pod address range (`--cluster-ipv4-cidr`) can accommoda
Creating cluster mern-cluster in us-central1-c... Cluster is being
Created [https://container.googleapis.com/v1/projects/upbeat-strat
To inspect the contents of your cluster, go to: https://console.cl
kubeconfig entry generated for mern-cluster.
NAME: mern-cluster
LOCATION: us-central1-c
MASTER_VERSION: 1.30.5-gke.1014001
MASTER_IP: 34.56.235.45
MACHINE_TYPE: e2-medium
NODE_VERSION: 1.30.5-gke.1014001
NUM_NODES: 3
STATUS: RUNNING
rajpreetsinghjahagirdar59@cloudshell:~ (upbeat-stratum-437711-d0)$
```

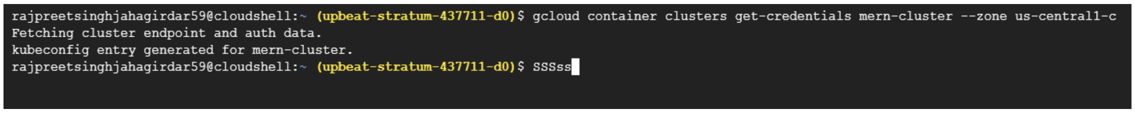


Cluster basics

| | |
|---|---|
| Name | mern-cluster |
| Location type | Zonal |
| Control plane zone | us-central1-c |
| Default node zones | us-central1-c |
| Release channel | Regular channel |
| Version | 1.30.5-gke.1014001 |
| Total size | 3 |
| External endpoint | 34.56.235.45<br>Show cluster certificate |
| Internal endpoint | 10.128.0.12<br>Show cluster certificate |
| Rollout sequence | To use rollout sequencing, register your cluster to a fleet |

**2. Get cluster credentials**: Fetch the credentials for your newly created cluster: -

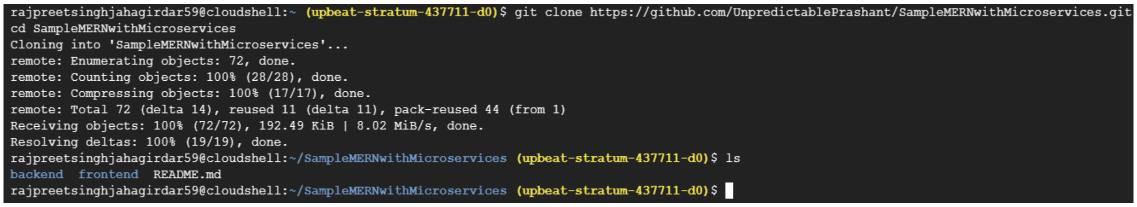
gcloud container clusters get-credentials mern-cluster --zone us-central1-c



```
rajpreetsinghjahagirdar59@cloudshell:~ (upbeat-stratum-437711-d0)$ gcloud container clusters get-credentials mern-cluster --zone us-central1-c
Fetching cluster endpoint and auth data.
kubeconfig entry generated for mern-cluster.
rajpreetsinghjahagirdar59@cloudshell:~ (upbeat-stratum-437711-d0)$ SSSss
```

Step 3: Clone the MERN Application Repository: -

**Clone the Sample MERN application**: In Cloud Shell, clone the GitHub repository: -

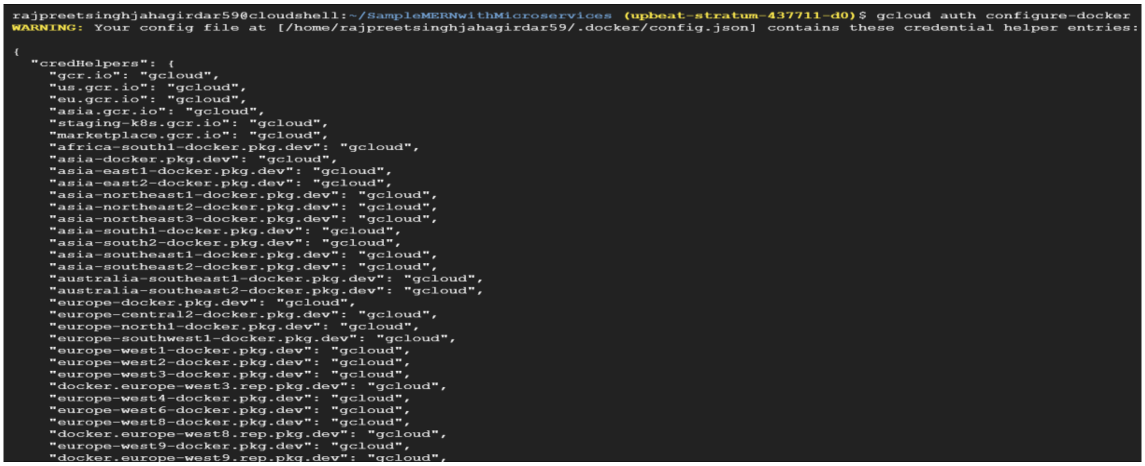


```
rajpreetsinghjahagirdar59@cloudshell:~ (upbeat-stratum-437711-d0)$ git clone https://github.com/UnpredictablePrashant/SampleMERNwithMicroservices.git
cd SampleMERNwithMicroservices
Cloning into 'SampleMERNwithMicroservices'...
remote: Enumerating objects: 72, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 72 (delta 14), reused 11 (delta 11), pack-reused 44 (from 1)
Receiving objects: 100% (72/72), 192.49 KiB | 8.02 MiB/s, done.
Resolving deltas: 100% (19/19), done.
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices (upbeat-stratum-437711-d0)$ ls
backend  frontend  README.md
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices (upbeat-stratum-437711-d0)$
```

Step 4: Build and Push Docker Images to Google Container Registry (GCR)

**Authenticate with GCR: -**

gcloud auth configure-docker



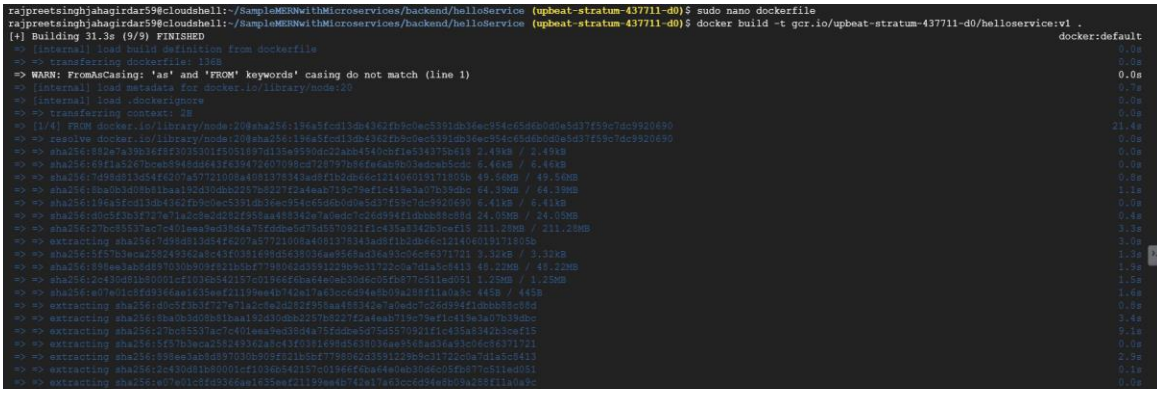
```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices (upbeat-stratum-437711-d0)$ gcloud auth configure-docker
WARNING: Your config file at [/home/rajpreetsinghjahagirdar59/.docker/config.json] contains these credential helper entries:

{
  "credHelpers": {
    "gcr.io": "gcloud",
    "us.gcr.io": "gcloud",
    "eu.gcr.io": "gcloud",
    "asia.gcr.io": "gcloud",
    "staging-k8s.gcr.io": "gcloud",
    "marketplace.gcr.io": "gcloud",
    "africa-south1-docker.pkg.dev": "gcloud",
    "asia-docker.pkg.dev": "gcloud",
    "asia-east1-docker.pkg.dev": "gcloud",
    "asia-east2-docker.pkg.dev": "gcloud",
    "asia-northeast1-docker.pkg.dev": "gcloud",
    "asia-northeast2-docker.pkg.dev": "gcloud",
    "asia-northeast3-docker.pkg.dev": "gcloud",
    "asia-south1-docker.pkg.dev": "gcloud",
    "asia-south2-docker.pkg.dev": "gcloud",
    "asia-southeast1-docker.pkg.dev": "gcloud",
    "asia-southeast2-docker.pkg.dev": "gcloud",
    "australia-southeast1-docker.pkg.dev": "gcloud",
    "australia-southeast2-docker.pkg.dev": "gcloud",
    "europe-docker.pkg.dev": "gcloud",
    "europe-central2-docker.pkg.dev": "gcloud",
    "europe-north1-docker.pkg.dev": "gcloud",
    "europe-southwest1-docker.pkg.dev": "gcloud",
    "europe-west1-docker.pkg.dev": "gcloud",
    "europe-west2-docker.pkg.dev": "gcloud",
    "europe-west3-docker.pkg.dev": "gcloud",
    "docker.europe-west3.rep.pkg.dev": "gcloud",
    "europe-west4-docker.pkg.dev": "gcloud",
    "europe-west6-docker.pkg.dev": "gcloud",
    "europe-west8-docker.pkg.dev": "gcloud",
    "docker.europe-west8.rep.pkg.dev": "gcloud",
    "europe-west9-docker.pkg.dev": "gcloud",
    "docker.europe-west9.rep.pkg.dev": "gcloud",
```

**2) Build and tag Docker images**: You need to build and tag Docker images for the backend (helloService, profileService) and frontend.

**Backend (helloService)**:-

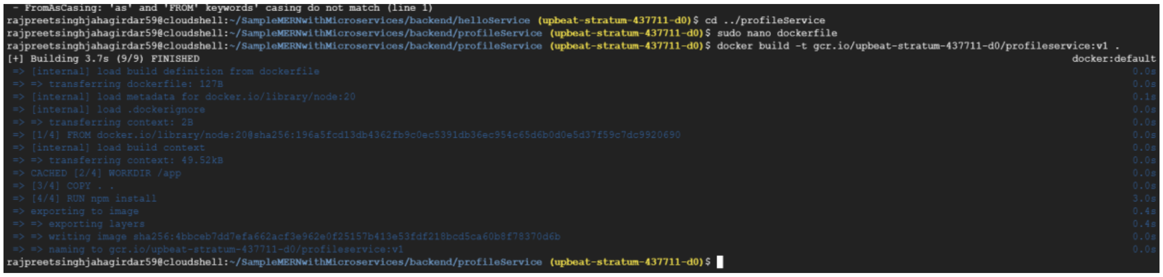
cd backend/helloService

docker build -t gcr.io/[PROJECT_ID]/helloservice:v1 .



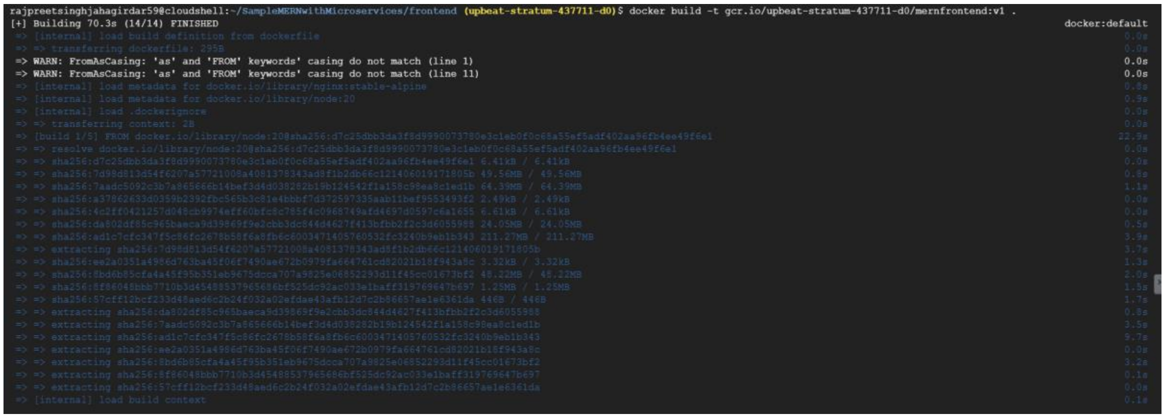
```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/helloService (upbeat-stratum-437711-d0)$ sudo nano dockerfile
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/helloService (upbeat-stratum-437711-d0)$ docker build -t gcr.io/upbeat-stratum-437711-d0/helloservice:v1 .
[+] Building 31.3s (9/9) FINISHED                                                                                          docker:default
 => [internal] load build definition from dockerfile                                                                                 0.0s
 => => transferring dockerfile: 136B                                                                                                 0.0s
 => WARN: FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 1)                                                       0.0s
 => [internal] load metadata for docker.io/library/node:20                                                                           0.7s
 => [internal] load .dockerignore                                                                                                    0.0s
 => => transferring context: 2B                                                                                                      0.0s
 => [1/4] FROM docker.io/library/node:20@sha256:196a5fcd13db4362fb9c0ec5391db36ec954c65d6b0d0e5d37f59c7dc9920690                   21.4s
 => => resolve docker.io/library/node:20@sha256:196a5fcd13db4362fb9c0ec5391db36ec954c65d6b0d0e5d37f59c7dc9920690                    0.0s
 => => sha256:882e7a39b36f8f3035301f5051897d135e9590dc22abb4540cbf1e534375b619 2.49kB / 2.49kB                                      0.0s
 => => sha256:69f1a5267bceb8948dd643f639472607098cd728797b86fe6ab9b03edceb5cdc 6.46kB / 6.46kB                                      0.0s
 => => sha256:7d98d813d54f6207a57721008a4081378343ad8f1b2db66c121406019171805b 49.56MB / 49.56MB                                    0.8s
 => => sha256:8ba0b3d08b81baa192d30dbb2257b8227f2a4eab719c79ef1c419e3a07b39dbc 64.39MB / 64.39MB                                    1.1s
 => => sha256:196a5fcd13db4362fb9c0ec5391db36ec954c65d6b0d0e5d37f59c7dc9920690 6.41kB / 6.41kB                                      0.0s
 => => sha256:d0c5f3b3f727e71a2c8e2d282f958aa488342e7a0edc7c26d994f1dbbb88c88d 24.05MB / 24.05MB                                    0.4s
 => => sha256:27bc85537ac7c401eea9ed38d4a75fddbe5d75d5570921f1c435a8342b3cef15 211.28MB / 211.28MB                                  3.3s
 => => extracting sha256:7d98d813d54f6207a57721008a4081378343ad8f1b2db66c121406019171805b                                           3.0s
 => => sha256:5f57b3eca258249362a8c43f0381698d5638036ae9568ad36a93c06c86371721 3.32kB / 3.32kB                                      1.3s
 => => sha256:898ee3ab8d897030b909f821b5bf7798062d3591229b9c31722c0a7d1a5c8413 48.22MB / 48.22MB                                    1.9s
 => => sha256:2c430d81b80001cf1036b542157c01966f6ba64e0eb30d6c05fb877c511ed051 1.25MB / 1.25MB                                      1.5s
 => => sha256:e07e01c8fd9366ae1635eef21199ee4b742e17a63cc6d94e8b09a288f11a0a9c 445B / 445B                                          1.6s
 => => extracting sha256:d0c5f3b3f727e71a2c8e2d282f958aa488342e7a0edc7c26d994f1dbbb88c88d                                           0.6s
 => => extracting sha256:8ba0b3d08b81baa192d30dbb2257b8227f2a4eab719c79ef1c419e3a07b39dbc                                           3.4s
 => => extracting sha256:27bc85537ac7c401eea9ed38d4a75fddbe5d75d5570921f1c435a8342b3cef15                                           9.1s
 => => extracting sha256:5f57b3eca258249362a8c43f0381698d5638036ae9568ad36a93c06c86371721                                           0.0s
 => => extracting sha256:898ee3ab8d897030b909f821b5bf7798062d3591229b9c31722c0a7d1a5c8413                                           2.9s
 => => extracting sha256:2c430d81b80001cf1036b542157c01966f6ba64e0eb30d6c05fb877c511ed051                                           0.1s
 => => extracting sha256:e07e01c8fd9366ae1635eef21199ee4b742e17a63cc6d94e8b09a288f11a0a9c                                           0.0s
```

**Backend (profileService)**:-



```
 - FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 1)
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/helloService (upbeat-stratum-437711-d0)$ cd ../profileService
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/profileService (upbeat-stratum-437711-d0)$ sudo nano dockerfile
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/profileService (upbeat-stratum-437711-d0)$ docker build -t gcr.io/upbeat-stratum-437711-d0/profileservice:v1 .
[+] Building 3.7s (9/9) FINISHED                                                                                           docker:default
 => [internal] load build definition from dockerfile                                                                                 0.0s
 => => transferring dockerfile: 127B                                                                                                 0.0s
 => [internal] load metadata for docker.io/library/node:20                                                                           0.1s
 => [internal] load .dockerignore                                                                                                    0.0s
 => => transferring context: 2B                                                                                                      0.0s
 => [1/4] FROM docker.io/library/node:20@sha256:196a5fcd13db4362fb9c0ec5391db36ec954c65d6b0d0e5d37f59c7dc9920690                    0.0s
 => [internal] load build context                                                                                                    0.0s
 => => transferring context: 49.52kB                                                                                                 0.0s
 => CACHED [2/4] WORKDIR /app                                                                                                        0.0s
 => [3/4] COPY . .                                                                                                                   0.0s
 => [4/4] RUN npm install                                                                                                            3.0s
 => exporting to image                                                                                                               0.4s
 => => exporting layers                                                                                                              0.4s
 => => writing image sha256:4bbceb7dd7efa662acf3e962e0f25157b413e53fdf218bcd5ca60b8f78370d6b                                        0.0s
 => => naming to gcr.io/upbeat-stratum-437711-d0/profileservice:v1                                                                   0.0s
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/profileService (upbeat-stratum-437711-d0)$
```

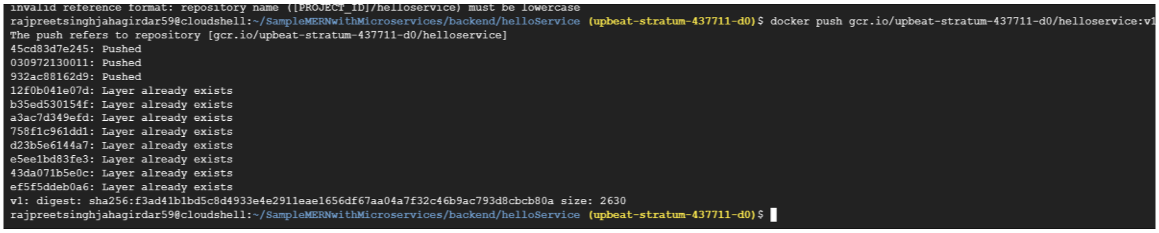
**Frontend: -**



```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/frontend (upbeat-stratum-437711-d0)$ docker build -t gcr.io/upbeat-stratum-437711-d0/mernfrontend:v1 .
[+] Building 70.3s (14/14) FINISHED                                                                                              docker:default
 => [internal] load build definition from dockerfile                                                                                      0.0s
 => => transferring dockerfile: 295B                                                                                                      0.0s
 => WARN: FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 1)                                                            0.0s
 => WARN: FromAsCasing: 'as' and 'FROM' keywords' casing do not match (line 11)                                                           0.0s
 => [internal] load metadata for docker.io/library/nginx:stable-alpine                                                                    0.8s
 => [internal] load metadata for docker.io/library/node:20                                                                                0.9s
 => [internal] load .dockerignore                                                                                                         0.0s
 => => transferring context: 2B                                                                                                           0.0s
 => [build 1/5] FROM docker.io/library/node:20@sha256:d7c25dbb3da3f8d9990073780e3c1eb0f0c68a55ef5adf402aa96fb4ee49f6e1                   22.9s
 => => resolve docker.io/library/node:20@sha256:d7c25dbb3da3f8d9990073780e3c1eb0f0c68a55ef5adf402aa96fb4ee49f6e1                          0.0s
 => => sha256:d7c25dbb3da3f8d9990073780e3c1eb0f0c68a55ef5adf402aa96fb4ee49f6e1 6.41kB / 6.41kB                                            0.0s
 => => sha256:7d98d813d54f6207a57721008a4081378343ad8f1b2db66c121406019171805b 49.56MB / 49.56MB                                          0.8s
 => => sha256:7aadc5092c3b7a865666b14bef3d4d038282b19b124542f1a158c98ea8c1ed1b 64.39MB / 64.39MB                                          1.1s
 => => sha256:a37862633d0359b2392fbc565b3c81e4bbbf7d372597335aab11bef9553493f2 2.49kB / 2.49kB                                            0.0s
 => => sha256:4c2ff0421257d048cb9974eff60bfc8c785f4c0968749afd4697d0597c6a1655 6.61kB / 6.61kB                                            0.0s
 => => sha256:da802df85c965baeca9d39869f9e2cbb3dc844d4627f413bfbb2f2c3d6055988 24.05MB / 24.05MB                                          0.5s
 => => sha256:ad1c7cfc347f5c86fc2678b58f6a8fb6c600347140576053 2fc3240b9eb1b343 211.27MB / 211.27MB                                       3.9s
 => => extracting sha256:7d98d813d54f6207a57721008a4081378343ad8f1b2db66c121406019171805b                                                 3.7s
 => => sha256:ee2a0351a4986d763ba45f06f7490ae672b0979fa664761cd82021b18f943a8c 3.32kB / 3.32kB                                            1.3s
 => => sha256:8bd6b85cfa4a45f95b351eb9675dcca707a9825e06852293d11f45cc01673bf2 48.22MB / 48.22MB                                          2.0s
 => => sha256:8f86048bbb7710b3d4548853796568 6bf525dc92ac033e1baff319769647b697 1.25MB / 1.25MB                                           1.5s
 => => sha256:57cff12bcf233d48aed6c2b24f032a02efdae43afb12d7c2b86657ae1e6361da 446B / 446B                                                1.7s
 => => extracting sha256:da802df85c965baeca9d39869f9e2cbb3dc844d4627f413bfbb2f2c3d6055988                                                 0.8s
 => => extracting sha256:7aadc5092c3b7a865666b14bef3d4d038282b19b124542f1a158c98ea8c1ed1b                                                 3.5s
 => => extracting sha256:ad1c7cfc347f5c86fc2678b58f6a8fb6c6003471405760532fc3240b9eb1b343                                                 9.7s
 => => extracting sha256:ee2a0351a4986d763ba45f06f7490ae672b0979fa664761cd82021b18f943a8c                                                 0.0s
 => => extracting sha256:8bd6b85cfa4a45f95b351eb9675dcca707a9825e06852293d11f45cc01673bf2                                                 3.2s
 => => extracting sha256:8f86048bbb7710b3d4548853796568 6bf525dc92ac033e1baff319769647b697                                                0.1s
 => => extracting sha256:57cff12bcf233d48aed6c2b24f032a02efdae43afb12d7c2b86657ae1e6361da                                                 0.0s
 => [internal] load build context                                                                                                         0.1s
```

**3. Push the Docker images to GCR**:-

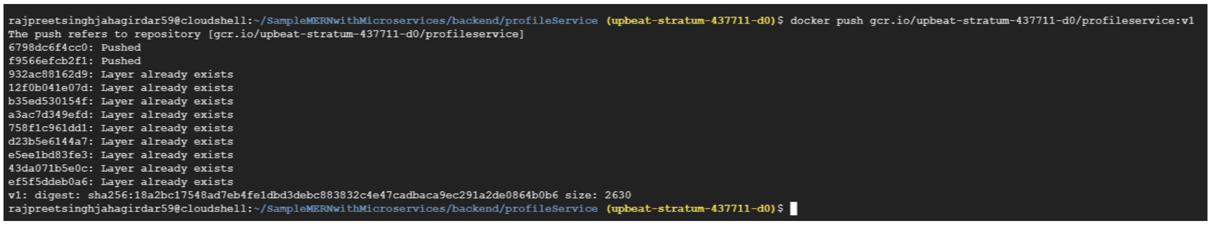
docker push gcr.io/[PROJECT_ID]/helloservice:v1



```
invalid reference format: repository name ([PROJECT_ID]/helloservice) must be lowercase
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/helloService (upbeat-stratum-437711-d0)$ docker push gcr.io/upbeat-stratum-437711-d0/helloservice:v1
The push refers to repository [gcr.io/upbeat-stratum-437711-d0/helloservice]
45cd83d7e245: Pushed
030972130011: Pushed
932ac88162d9: Pushed
12f0b041e07d: Layer already exists
b35ed530154f: Layer already exists
a3ac7d349efd: Layer already exists
758f1c961dd1: Layer already exists
d23b5e6144a7: Layer already exists
e5ee1bd83fe3: Layer already exists
43da071b5e0c: Layer already exists
ef5f5ddeb0a6: Layer already exists
v1: digest: sha256:f3ad41b1bd5c8d4933e4e2911eae1656df67aa04a7f32c46b9ac793d8cbcb80a size: 2630
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/helloService (upbeat-stratum-437711-d0)$
```

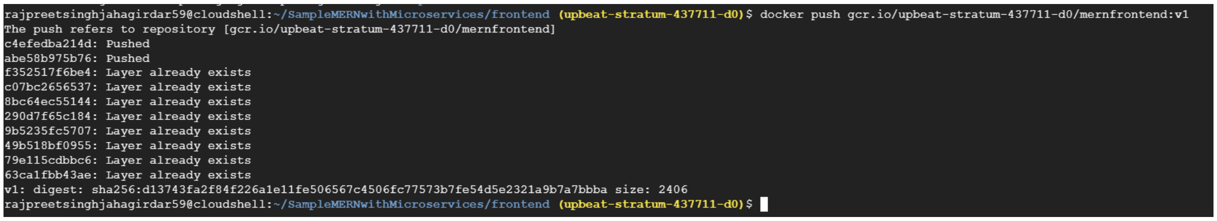
For profileService:-

docker push gcr.io/[PROJECT_ID]/profileservice:v1



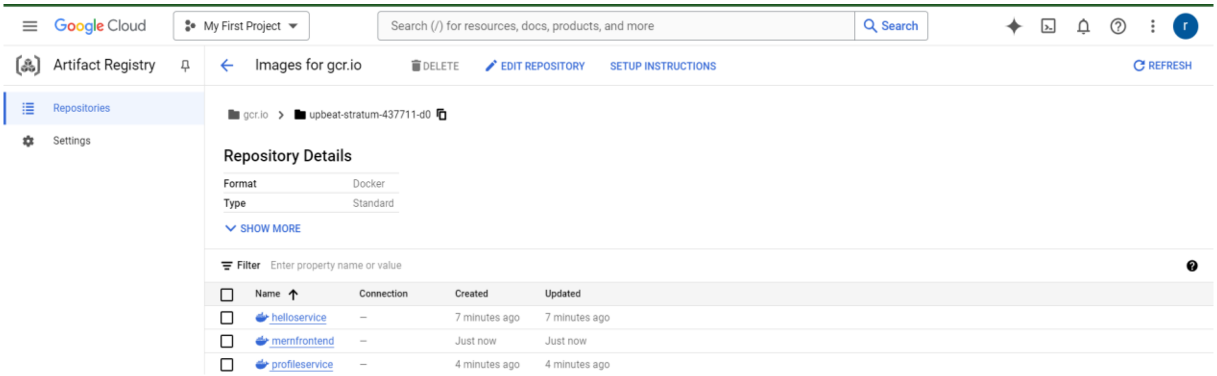
```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/profileService (upbeat-stratum-437711-d0)$ docker push gcr.io/upbeat-stratum-437711-d0/profileservice:v1
The push refers to repository [gcr.io/upbeat-stratum-437711-d0/profileservice]
6798dc6f4cc0: Pushed
f9566efcb2f1: Pushed
932ac88162d9: Layer already exists
12f0b041e07d: Layer already exists
b35ed530154f: Layer already exists
a3ac7d349efd: Layer already exists
758f1c961dd1: Layer already exists
d23b5e6144a7: Layer already exists
e5ee1bd83fe3: Layer already exists
43da071b5e0c: Layer already exists
ef5f5ddeb0a6: Layer already exists
v1: digest: sha256:18a2bc17548ad7eb4fe1dbd3debc883832c4e47cadbaca9ec291a2de0864b0b6 size: 2630
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/profileService (upbeat-stratum-437711-d0)$
```

- For mernfrontend:-



```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/frontend (upbeat-stratum-437711-d0)$ docker push gcr.io/upbeat-stratum-437711-d0/mernfrontend:v1
The push refers to repository [gcr.io/upbeat-stratum-437711-d0/mernfrontend]
c4efedba214d: Pushed
abe58b975b76: Pushed
f352517f6be4: Layer already exists
c07bc2656537: Layer already exists
8bc64ec55144: Layer already exists
290d7f65c184: Layer already exists
9b5235fc5707: Layer already exists
49b518bf0955: Layer already exists
79e115cdbbc6: Layer already exists
63ca1fbb43ae: Layer already exists
v1: digest: sha256:d13743fa2f84f226a1e11fe506567c4506fc77573b7fe54d5e2321a9b7a7bbba size: 2406
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/frontend (upbeat-stratum-437711-d0)$
```

Step 5: Create Kubernetes Deployment and Service Files:-



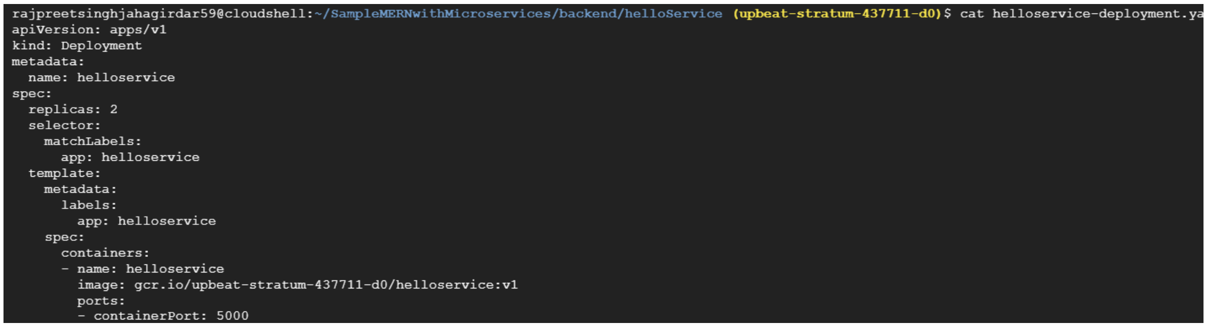
Repository Details

| | |
|---|---|
| Format | Docker |
| Type | Standard |

| □ | Name ↑ | Connection | Created | Updated |
|---|---|---|---|---|
| □ | helloservice | – | 7 minutes ago | 7 minutes ago |
| □ | mernfrontend | – | Just now | Just now |
| □ | profileservice | – | 4 minutes ago | 4 minutes ago |

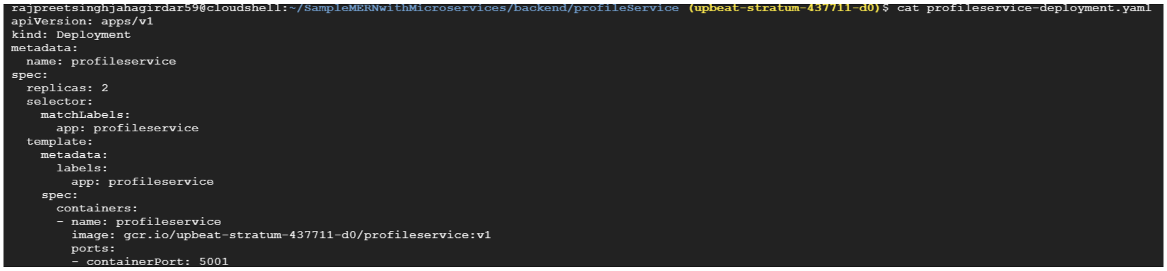
In Cloud Shell, create Kubernetes YAML files for each microservice.

1. **Create helloservice-deployment.yaml:-**

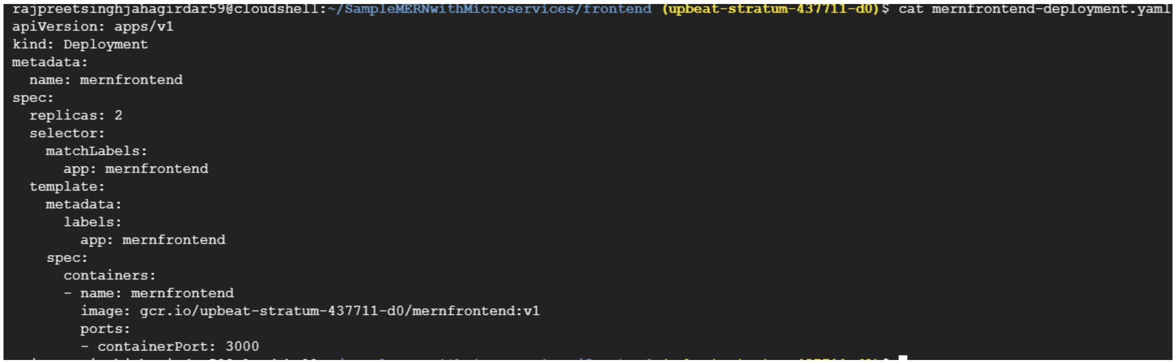


```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/helloService (upbeat-stratum-437711-d0)$ cat helloservice-deployment.ya
apiVersion: apps/v1
kind: Deployment
metadata:
  name: helloservice
spec:
  replicas: 2
  selector:
    matchLabels:
      app: helloservice
  template:
    metadata:
      labels:
        app: helloservice
    spec:
      containers:
      - name: helloservice
        image: gcr.io/upbeat-stratum-437711-d0/helloservice:v1
        ports:
        - containerPort: 5000
```

2.Create `profileservice-deployment.yaml`:-

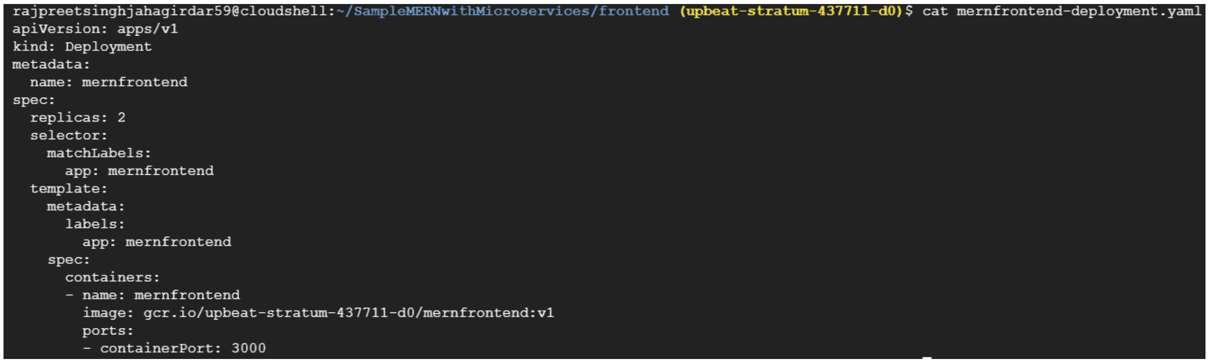


```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/profileService (upbeat-stratum-437711-d0)$ cat profileservice-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: profileservice
spec:
  replicas: 2
  selector:
    matchLabels:
      app: profileservice
  template:
    metadata:
      labels:
        app: profileservice
    spec:
      containers:
      - name: profileservice
        image: gcr.io/upbeat-stratum-437711-d0/profileservice:v1
        ports:
        - containerPort: 5001
```



```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/frontend (upbeat-stratum-437711-d0)$ cat mernfrontend-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mernfrontend
spec:
  replicas: 2
  selector:
    matchLabels:
      app: mernfrontend
  template:
    metadata:
      labels:
        app: mernfrontend
    spec:
      containers:
      - name: mernfrontend
        image: gcr.io/upbeat-stratum-437711-d0/mernfrontend:v1
        ports:
        - containerPort: 3000
```

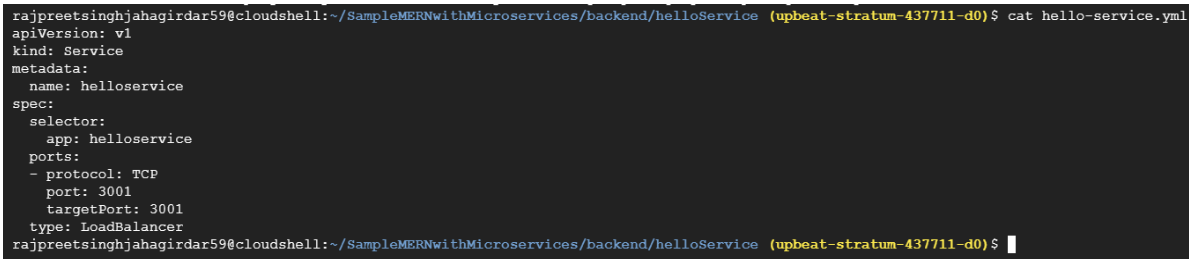
### mernfrontend-service.yaml:-



```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/frontend (upbeat-stratum-437711-d0)$ cat mernfrontend-deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mernfrontend
spec:
  replicas: 2
  selector:
    matchLabels:
      app: mernfrontend
  template:
    metadata:
      labels:
        app: mernfrontend
    spec:
      containers:
      - name: mernfrontend
        image: gcr.io/upbeat-stratum-437711-d0/mernfrontend:v1
        ports:
        - containerPort: 3000
```

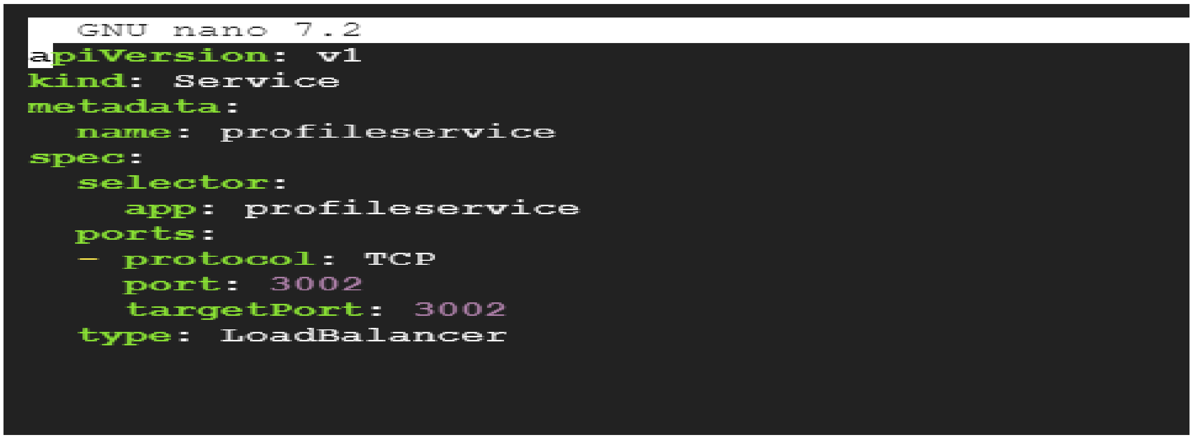
### 4. Create Service files for each microservice: -

### helloservice-service.yaml:-



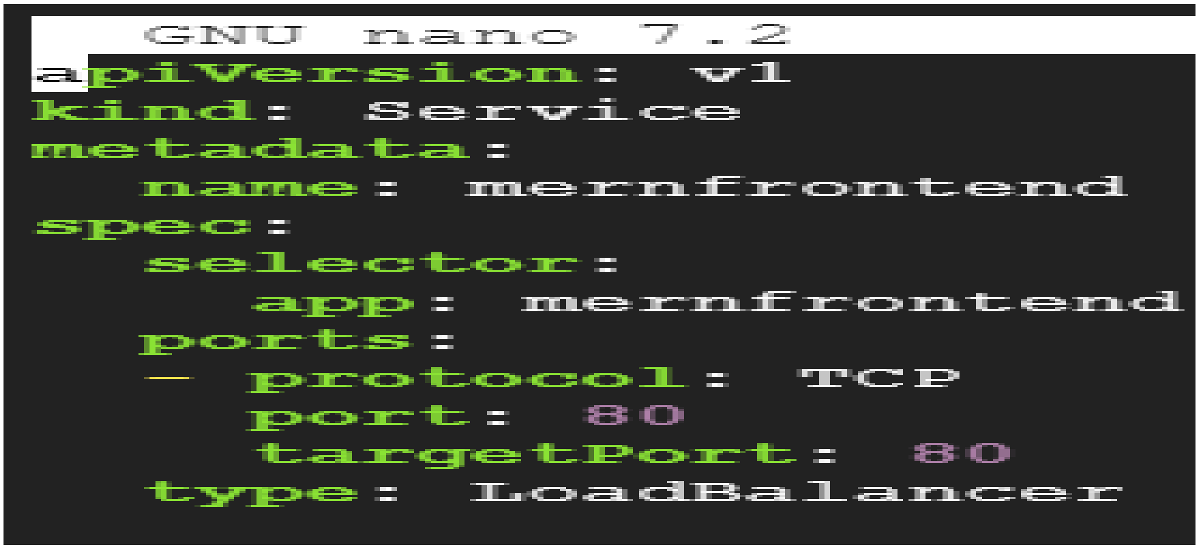
```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/helloService (upbeat-stratum-437711-d0)$ cat hello-service.yml
apiVersion: v1
kind: Service
metadata:
  name: helloservice
spec:
  selector:
    app: helloservice
  ports:
  - protocol: TCP
    port: 3001
    targetPort: 3001
  type: LoadBalancer
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/helloService (upbeat-stratum-437711-d0)$
```

### profileservice-service.yaml:-



```
  GNU nano 7.2
apiVersion: v1
kind: Service
metadata:
  name: profileservice
spec:
  selector:
    app: profileservice
  ports:
  - protocol: TCP
    port: 3002
    targetPort: 3002
  type: LoadBalancer
```

**mernfrontend-service.yaml:-**

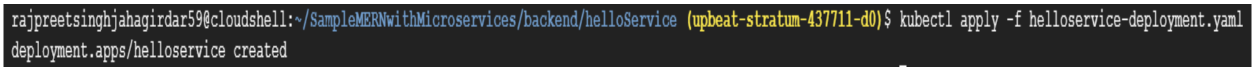


```
GNU nano 7.2
apiVersion: v1
kind: Service
metadata:
  name: mernfrontend
spec:
  selector:
    app: mernfrontend
  ports:
  - protocol: TCP
    port: 80
    targetPort: 80
  type: LoadBalancer
```

**Step 6: Deploy the Services to GKE:-**

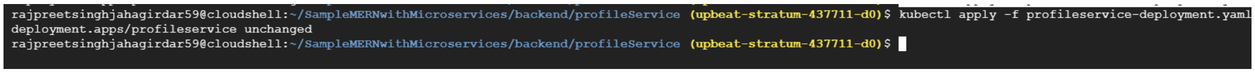
1. **Apply the deployments and services**:

kubectl apply -f helloservice-deployment.yaml



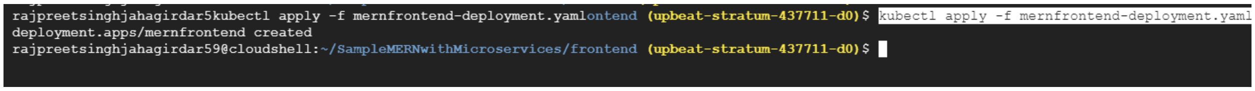
```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/helloService (upbeat-stratum-437711-d0)$ kubectl apply -f helloservice-deployment.yaml
deployment.apps/helloservice created
```

kubectl apply -f profileservice-deployment.yaml



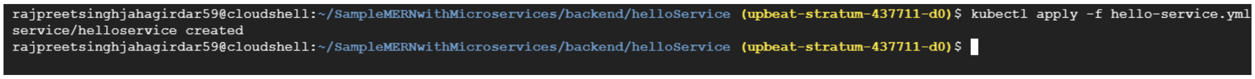
```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/profileService (upbeat-stratum-437711-d0)$ kubectl apply -f profileservice-deployment.yaml
deployment.apps/profileservice unchanged
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/profileService (upbeat-stratum-437711-d0)$
```

kubectl apply -f mernfrontend-deployment.yaml



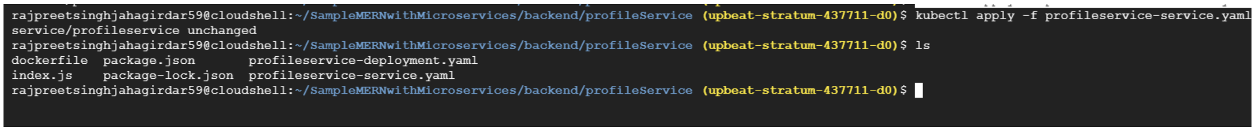
```
rajpreetsinghjahagirdar5kubectl apply -f mernfrontend-deployment.yamlontend (upbeat-stratum-437711-d0)$ kubectl apply -f mernfrontend-deployment.yaml
deployment.apps/mernfrontend created
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/frontend (upbeat-stratum-437711-d0)$
```

kubectl apply -f helloservice-service.yaml



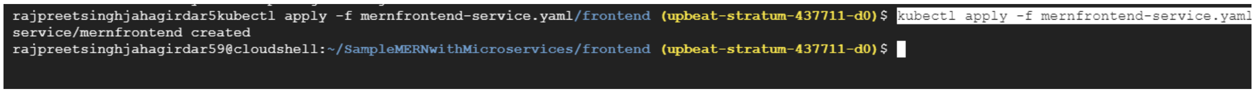
```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/helloService (upbeat-stratum-437711-d0)$ kubectl apply -f hello-service.yml
service/helloservice created
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/helloService (upbeat-stratum-437711-d0)$
```

kubectl apply -f profileservice-service.yaml



```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/profileService (upbeat-stratum-437711-d0)$ kubectl apply -f profileservice-service.yaml
service/profileservice unchanged
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/profileService (upbeat-stratum-437711-d0)$ ls
dockerfile  package.json       profileservice-deployment.yaml
index.js    package-lock.json  profileservice-service.yaml
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/backend/profileService (upbeat-stratum-437711-d0)$
```

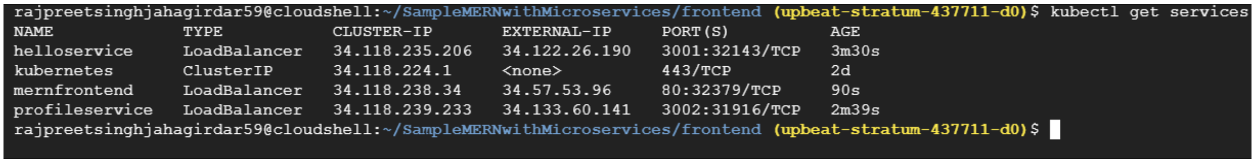
kubectl apply -f mernfrontend-service.yaml



```
rajpreetsinghjahagirdar5kubectl apply -f mernfrontend-service.yaml/frontend (upbeat-stratum-437711-d0)$ kubectl apply -f mernfrontend-service.yaml
service/mernfrontend created
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/frontend (upbeat-stratum-437711-d0)$
```

**2.Verify that the services are running**:-

kubectl get services

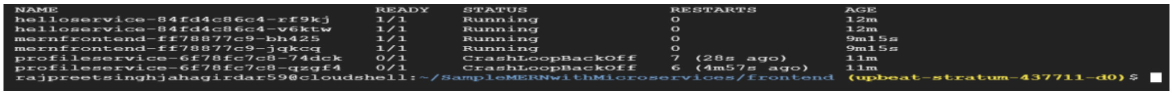


```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/frontend (upbeat-stratum-437711-d0)$ kubectl get services
NAME             TYPE           CLUSTER-IP       EXTERNAL-IP      PORT(S)          AGE
helloservice     LoadBalancer   34.118.235.206   34.122.26.190    3001:32143/TCP   3m30s
kubernetes       ClusterIP      34.118.224.1     <none>           443/TCP          2d
mernfrontend     LoadBalancer   34.118.238.34    34.57.53.96      80:32379/TCP     90s
profileservice   LoadBalancer   34.118.239.233   34.133.60.141    3002:31916/TCP   2m39s
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/frontend (upbeat-stratum-437711-d0)$
```

Take note of the external IP addresses for each service. These IPs can be used to access the application in a browser.

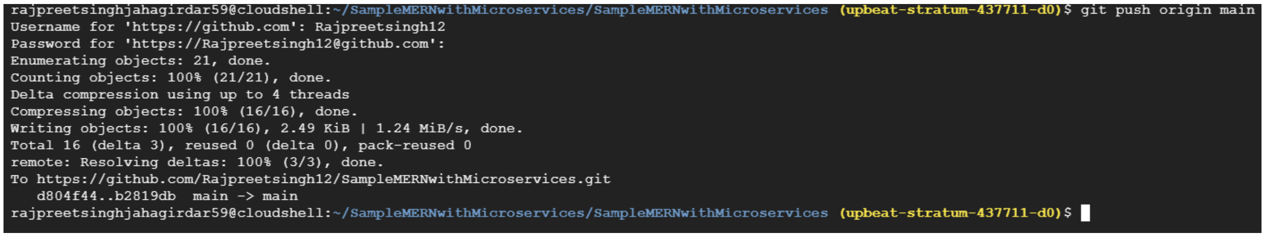
verify once if all the microservices pods are in running state:-

1. Kubectl get pods:-



```
NAME                               READY   STATUS             RESTARTS        AGE
helloservice-84fd4c86c4-rf9kj      1/1     Running            0               12m
helloservice-84fd4c86c4-v6ktw      1/1     Running            0               12m
mernfrontend-ff78877c9-bh425       1/1     Running            0               9m15s
mernfrontend-ff78877c9-jqkcq       1/1     Running            0               9m15s
profileservice-6f78fc7c8-74dck     0/1     CrashLoopBackOff   7 (28s ago)     11m
profileservice-6f78fc7c8-qzgf4     0/1     CrashLoopBackOff   6 (4m57s ago)   11m
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/frontend (upbeat-stratum-437711-d0)$
```

Step 9: Push to GitHub



```
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/SampleMERNwithMicroservices (upbeat-stratum-437711-d0)$ git push origin main
Username for 'https://github.com': Rajpreetsingh12
Password for 'https://Rajpreetsingh12@github.com':
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 4 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (16/16), 2.49 KiB | 1.24 MiB/s, done.
Total 16 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/Rajpreetsingh12/SampleMERNwithMicroservices.git
   d804f44..b2819db  main -> main
rajpreetsinghjahagirdar59@cloudshell:~/SampleMERNwithMicroservices/SampleMERNwithMicroservices (upbeat-stratum-437711-d0)$
```