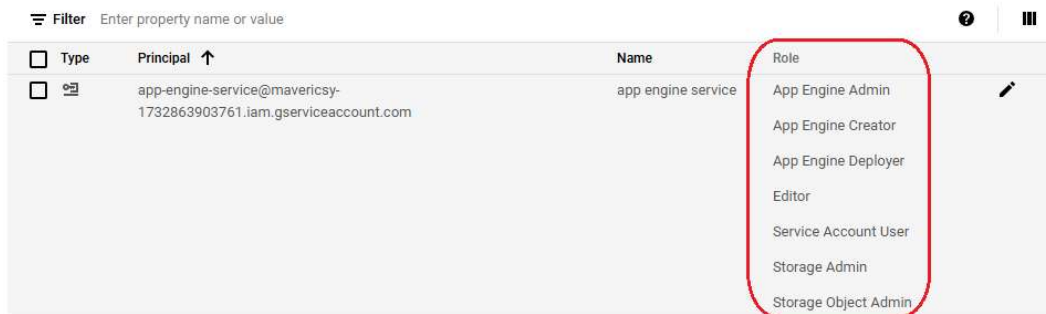


Instructions for Creating an App Engine Service through Cloushell:

1. Create a new service in App Engine don't user any existing App Engine Service Account.
2. Please use the "us-central1 (Iowa)" for App Engine creation.
3. Assign the following permissions to the service account:

- App Engine Admin
- App Engine Creator
- App Engine Deployer
- Service Account User
- Storage Object Admin
- Storage Admin
- Editor

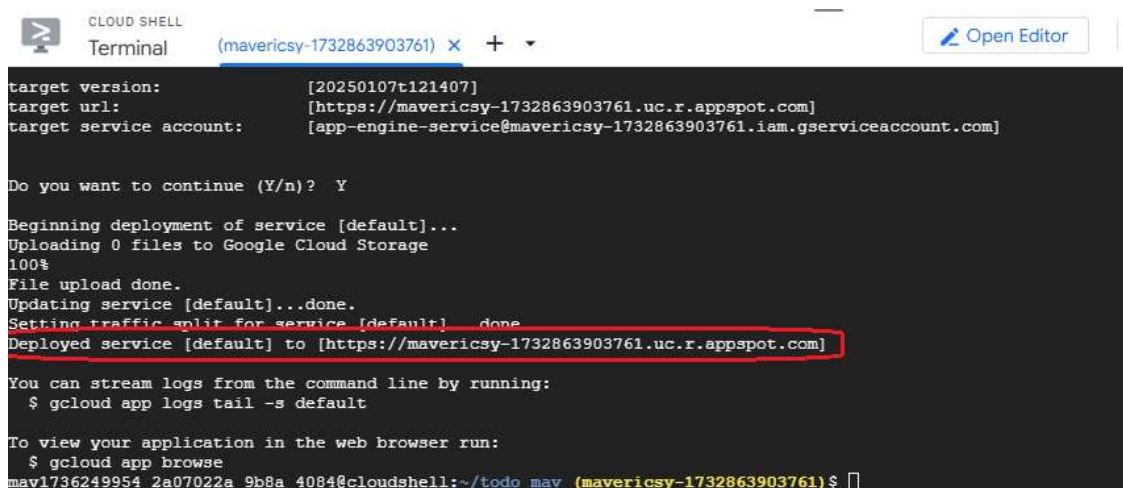


Type	Principal	Name	Role
<input type="checkbox"/>	app-engine-service@mavericsy-1732863903761.iam.gserviceaccount.com	app engine service	App Engine Admin
			App Engine Creator
			App Engine Deployer
			Editor
			Service Account User
			Storage Admin
			Storage Object Admin

- 4.
5. If the error pertains to the Build with Staging bucket, please ignore it, wait for 2 minutes, and then run `gcloud app deploy` again. The App Engine should deploy successfully.

```
Beginning deployment of service [default]...
Created .gcloudignore file. See 'gcloud topic gcloudignore' for details.
- Uploading 3 files to Google Cloud Storage
File upload done.
Updating service [default]...failed.
ERROR: (gcloud.app.deploy) Error Response: (13) Failed to create cloud build: com.google.net.rpc3.client.RpcClientException: <eye3 title="/ArgoAdminNoCloudAudit.CreateBuild, FAILED_PRECONDITION"/> APPLICATION ERROR:google.devtools.cloudbuild.v1/ArgoAdminNoCloudAudit.CreateBuild;invalid bucket "staging.mavericsy-1732863913160.appspot.com". Service account mavericsy-deploy@mavericsy-1732863913160.iam.gserviceaccount.com does not have access to the bucket;AppErrorCode=9;StartTimeMs=-1732863913160;unknown;ReqFormat=uncompressed;ServerTimeSec=0.642211739;LogBytes=256;Non-FailFast;EndUserCredsRequested;EffSecLevel=privacy_and_integrity;ReqFormat=uncompressed;ReqID=35ff41a029a3a48af;GlobalID=0;Server=12002-a0536600x5031-b0-331-6d85-7b71-4001
```

6. App Engine Created



```
CLOUD SHELL
Terminal (mavericsy-1732863903761) x + - Open Editor

target version: [20250107t121407]
target url: [https://mavericsy-1732863903761.uc.r.appspot.com]
target service account: [app-engine-service@mavericsy-1732863903761.iam.gserviceaccount.com]

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

Beginning deployment of service [default]...
Uploading 0 files to Google Cloud Storage
100%
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done
Deployed service [default] to [https://mavericsy-1732863903761.uc.r.appspot.com]

You can stream logs from the command line by running:
$ gcloud app logs tail -s default

To view your application in the web browser run:
$ gcloud app browse
mav1736249954_2a07022a_9b8a_4084@cloudshell:~/todo_mav (mavericsy-1732863903761) $
```

Instructions for Creating an App Engine Service through Compute Engine:


1. Create a new service in App Engine don't user any existing App Engine Service Account.
2. Please use the "us-central1 (Iowa)" for App Engine creation.
3. Assign the following permissions to the service account:
 - App Engine Admin
 - App Engine Creator
 - App Engine Deployer
 - Service Account User
 - Storage Object Admin
 - Storage Admin
 - Editor
 - Service Usage Admin
4. Enable the below API's (Navigate to APIs& Service -> Library)
 - Compute Engine API
 - App Engine: ([API Link](#))
 - Service Usage API ([API Link](#))
5. Create an App Engine in the "us-central1 (Iowa)" region. Use the service account you created.

Create app

1 Configure application — 2 Get started

Region

Select a region for your App Engine application. Please remember, once selected the region is permanently tied to the project.



Select a region *
us-central

Identity and API access

Select a service account *
mavericsy-deploy

If no service account is selected the default App Engine service account will be used.

Next

6. Create a virtual machine with Ubuntu 20 and select the service account which created manually.

← Create an instance [CREATE VM FROM...](#) [EQUIVALENT CODE](#)

- Machine configuration
e2-medium, us-central1
- OS and storage
Ubuntu 20.04 LTS
- Networking
1 network interface
- Observability
- Security**
- Advanced

Security

Identity and API access

Service accounts

Service account

Filter Filter service accounts

- No service account
- Compute Engine default service account
555195364853-compute@developer.gserviceaccount.com
- App Engine default service account
mavericsy-1732863913160@appspot.gserviceaccount.com
- mavericsy-deploy**
mavericsy-deploy@mavericsy-1732863913160.iam.gserviceaccount.com

[ENABLE](#)

Shielded VM

Turn on all settings for the most secure configuration.

- ☐ Turn on Secure Boot
- ☒ Turn on vTPM
- ☒ Turn on Integrity Monitoring

VM access

Manage how users connect to the VM

By default, when you connect to a VM using this console or gcloud, your SSH keys are generated automatically. [Learn more](#)

Monthly estimate

\$25.46
That's about \$0.03 hourly

Pay for what you use: no upfront costs and per second billing

Item	Monthly estimate
2 vCPU + 4 GB memory	\$24.46
10 GB balanced persistent disk	\$1.00
Total	\$25.46

[Compute Engine pricing](#)

[LESS](#)

7. And Select Allow full access to all Cloud APIs in Access Scope.

← Create an instance [CREATE VM FROM...](#)

- Machine configuration
e2-medium, us-central1
- OS and storage
Ubuntu 20.04 LTS
- Networking
1 network interface
- Observability
- Security**
- Advanced

Security

Identity and API access

Service accounts

Service account
mavericsy-deploy

Requires the Service Account User role (roles/iam.serviceAccountUser) to be set for users who want to access VMs with this service account. [Learn more](#)

Access scopes

- ☐ Allow default access
- ☒ **Allow full access to all Cloud APIs**
- ☐ Set access for each API

Confidential VM service

Confidential Computing is disabled on this VM instance

[ENABLE](#)

Shielded VM

Turn on all settings for the most secure configuration.

8. Once the Instance is created.

9. Install GCloud SDK:

- Download and install from GCloud SDK using below commands
 - **Update the package list:**
`sudo apt update`
 - **Install necessary packages:**
`sudo apt install apt-transport-https ca-certificates gnupg curl`
 - **Add the Cloud SDK distribution URI as a package source:**
`echo "deb [signed-by=/usr/share/keyrings/cloud.google.gpg] http://packages.cloud.google.com/apt cloud-sdk main" | sudo tee -a /etc/apt/sources.list.d/google-cloud-sdk.list`
 - **Import the Google Cloud public key:**
`curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo gpg --dearmor -o /usr/share/keyrings/cloud.google.gpg`
 - **Update the package list again:**
`sudo apt update`
 - **Install the Cloud SDK (Wait for 5 to 10 mins to install)**
`sudo apt install google-cloud-sdk`
 - **Initialize the Cloud SDK:**
`gcloud init`

10. Deploy the required App Engine Deployment

11. If the error pertains to the Build with bucket, please ignore it, wait for 2 minutes, and then run `gcloud app deploy` again. The App Engine should deploy successfully.

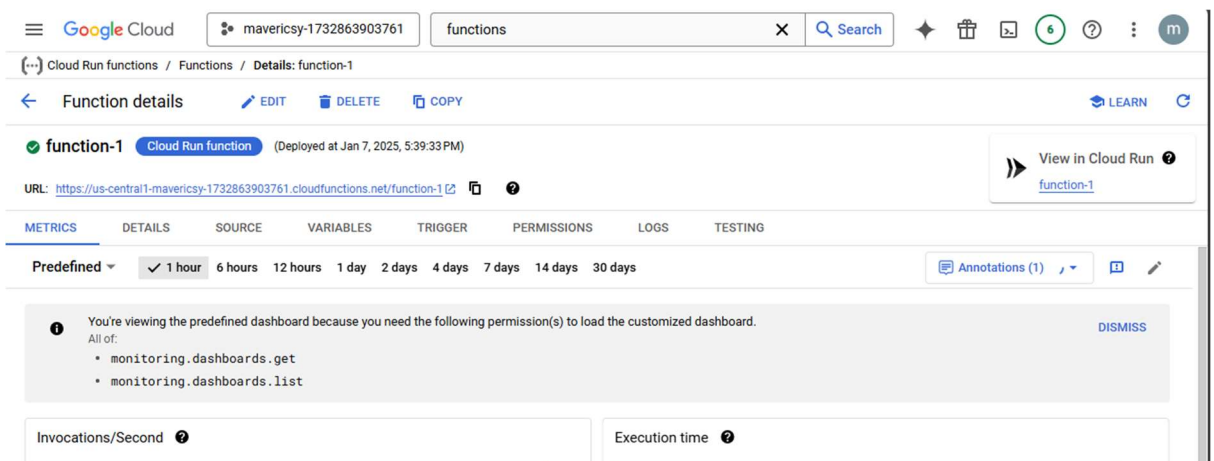
```
Beginning deployment of service [default]...
Created .gcloudignore file. See 'gcloud topic gcloudignore' for details.
-- Uploading 3 files to Google Cloud Storage --
File upload done.
Updating service [default]...failed.
ERROR: (gcloud.app.deploy) Error Response: [13] Failed to create cloud build: com.google.net.rpc3.client.RpcClientException: <eye3 title='/ArgoAdminNoCloudAudit.CreateBuild, FAILED PRECONDITION'/> APPLICATION ERROR:google.devtools.cloudbuild.v1/ArgoAdminNoCloudAudit.CreateBuild:invalid bucket "staging.mavericsy-1732863913160.appspot.com"; Service account mavericsy-deploy@mavericsy-1732863913160.iam.gserviceaccount.com does not have access to the bucket;AppErrorCode=9;StartTimeMs=-1736362361848;unknown;ResFormat=uncompressed;ServerTimeSec=0.642211739;LogBytes=256;Non-FailFast;EndUserCredsRequested;EffSecLevel=privacy_and_integrity;ReqFormat=uncompressed;ReqID=3ff6d1a028e30490f;GlobalID=0;Server=12002:a05:6600:5091:b0:391:6d85:7b71:4001
```

Instructions for Creating a Cloud Functions:

1. Create a new service for “Cloud function” don’t user any existing Service Account.
2. Please use the “us-central1 (Iowa)” for “Cloud function”.
3. Assign the following permissions to the service account:
 - Cloud Build Editor
 - Cloud Build Service Agent
 - Cloud Functions Admin
 - Cloud Functions Service Agent
 - Cloud Run Admin
 - Service Account Token Creator
 - Service Account User
 - Service Usage Consumer



- 4.
5. Cloud Function Created.



- 6.