

# GCP Settings

1. Create a new Google Cloud Project (or select an existing project): [Create a Cloud Project](#)
2. Create the App Engine: [App Engine](#)
3. Create a service account: [Service Account](#)
4. Add the the following roles to your new servive account (For more reading: [IAM Roles](#)) to your newly made service account:
  - App Engine Admin - allows for the creation of new App Engine apps
  - App Engine Service Admin - allows to view and change traffic splits, scaling settings, and delete old versions
  - Storage Admin - allows upload of source code
  - Storage Object Admin - allows to control GCS objects
  - Cloud Build Service Agent - allows to manage resources
  - Viewer - allows to access all resources
5. Enable app engine API: [App Engine API](#)
6. Enable cloud build API: [Cloud Build API](#)
7. Enable your "App Engine Admin" role and "Service Account user" role: [Cloud Build Setting](#)
8. Create a service account key (JSON) for the account "@appspot.gserviceaccount.com": [Service Account Key](#)

# GitHub Settings

9. Add the following to the "Secret" section in your Github repository "Settings" (For more read: [Repository Secrets](#)):
  - Name: GAE\_SA\_KEY (**FOR ADAM'S PROJECT**)
  - Value: Copy paste the code from the downloaded JSON file
10. Go to your Github repository and copy your code with "HTTP"
11. Clone your repository on google cloud shell:
  - git clone + COPY PASTE THE HTTP CODE => (<https://github.com/YOUR-GITHUB-USERNAME/YOUR-REPOSITORY-NAME>.git)
12. Go to your cloned repository (DIRECTORY):
  - cd DIR\_NAME

# Deployment

13. Deploy your app.yaml from the google cloud shell:
  - gcloud app deploy --quiet app.yaml --project **YOUR-PROJECT-ID** --promote
14. A file called .gcloudignore is added to your project directory, now we PUSH:
  - git push
15. Go back to your Github and go to "Actions" to see if you pass all the workflows

Console gcloud API Python

To create a new project:

1. Go to the [Manage resources](#) page in the Cloud Console.
2. On the **Select organization** drop-down list at the top of the page, select the organization in which you want to create a project. If you are a free trial user, skip this step, as this list does not appear.
3. Click **Create Project**.
4. In the **New Project** window that appears, enter a project name and select a billing account as applicable. A project name can contain only letters, numbers, single quotes, hyphens, spaces, or exclamation points, and must be between 4 and 30 characters.
5. Enter the parent organization or folder in the **Location** box. That resource will be the hierarchical parent of the new project.
6. When you're finished entering new project details, click **Create**.

#### Project Name:

- Human-Readable
- Editable at any time
- Not used by Google API

#### Project ID:

- Generated automatically IF not defined by user
- Unique
- Used by Google API

#### Project Number:

- Automatically generated

## In our case, let's name our project "foundations-cicd"

New Project

**1.**

You have 22 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name \* foundations-cicd

Project ID: foundations-cicd-305109. It cannot be changed later. [EDIT](#)

Organization \* code.berlin

Select an organization to attach it to a project. This selection can't be changed later.

Location \* code.berlin [BROWSE](#)

Parent organization or folder

**CREATE** **CANCEL**

**2.**

Google Cloud Platform

Manage resources [+ CREATE PROJECT](#) [CREATE FOLDER](#) MOVE DELETE

Filter Filter

Name	ID	Last accessed	Charges	Labels
code.berlin	744572392533	February 16, 2021		

RESOURCES PENDING DELETION

Notifications

- Create Project: foundations-cicd Just now [SELECT PROJECT](#)
- Create Project: My First Project 3 days ago [SELECT PROJECT](#)
- Create App Engine App se-foundations21-1 in europe-west3 se-foundations21-1 3 days ago
- Create Project: se-foundations21-1 3 days ago [SELECT PROJECT](#)

**3.**

Google Cloud Platform foundations-cicd

DASHBOARD ACTIVITY RECOMMENDATIONS [CUSTOMIZE](#)

Project info

- Project name foundations-cicd
- Project ID foundations-cicd-305109
- Project number 195861820902

[ADD PEOPLE TO THIS PROJECT](#)

[Go to project settings](#)

Resources

This project has no resources

Trace

No trace data from the past 7 days

[Get started with Trace](#)

Getting Started

RPI APIs

Requests (requests/sec)

[Go to APIs overview](#)

Google Cloud Platform status

All services normal

[Go to Cloud status dashboard](#)

Billing

Estimated charges USD \$0.00 For the billing period Feb 1 – 17, 2021

[Take a tour of billing](#)

[View detailed charges](#)

Monitoring

Set up alerting policies

Create uptime checks

[View all dashboards](#)

[Go to Monitoring](#)

gcloud API Console

To create a Cloud project and App Engine application with billing enabled:

1. Go to the App Engine page: [Go to the App Engine page](#)
2. Select or create a Cloud project.
3. In the **Your first app with Node.js** page, select a region and enable billing:
  - a. Select a region to specify where you want to your App Engine application located. [More information about App Engine locations](#)
  - b. Select or create a billing account to enable billing in your project.

After the App Engine application is created in your project, the **Dashboard** page opens.

1. Create App Engine
2. Choose your project
3. Choose europe-west-3 (Frankfurt)
4. If the result mentioned an error, just click the link and it will redirect us to a page that say "App Engine has been created"
4. 1. (OPTIONAL) While still on the page click "DOWNLOAD SDK" and follow **OPTIONAL STEP**
5. Otherwise click "Get Started" on our App Engine page (NO NEED)

## OPTIONAL STEP: Download SDK

- cloud SDK supports Python version 3, so check your python version from your terminal:

```
● ● ● ninovationlab@MacBook-Pro-3:~ Last login: Wed Feb 17 10:40:32 on ttys000
(base) ninovationlab:~/ $ python -V
Python 3.8.5
(base) ninovationlab:~/ $
```

- If your python is version 2, download it here <https://www.python.org/downloads/>

- Check your OS version from your terminal:

```
(base) ninovationlab:~/ $ getconf LONG_BIT
64
```

- Based on your OS choose between these 2 options:

Platform	Package	Size	SHA256 Checksum
macOS 64-bit (x86_64)	google-cloud-sdk-328.0.0-darwin-x86_64.tar.gz	110.1 MB	6871707329ede314344f527e3934b182db54f 56dd514902a0b09049d560f059
macOS 32-bit (x86)	google-cloud-sdk-328.0.0-darwin-x86.tar.gz	87.6 MB	c43843ed63255685f3c1098d1c2ba3eb77f51e 28b1ea83b5d3200f8c65e0f215

- (OPTIONAL) Add Cloud SDK to PATH from your terminal:

```
• cd google-cloud-sdk
• install.sh
```

- Initialize the SDK:

```
• cd google-cloud-sdk
• gcloud init
```

Console gcloud REST C# Go Java Python

1. In the Cloud Console, go to the Service accounts page  
[Go to the Service accounts page](#)

2. Select a project.

3. Click Create service account

4. Enter a service account name to display in the Cloud Console.  
The Cloud Console generates a service account ID based on this name. Edit the ID if necessary. You cannot change the ID later.

5. Optional: Enter a description of the service account.

6. If you do not want to set access controls now, click Done to finish creating the service account.  
To set access controls now, click Create and continue to the next step.

7. Optional: Choose one or more IAM roles to grant to the service account on the project.

8. When you're done adding roles, click Continue.

9. Optional: In the Service account users role field, add members that can impersonate the service account.

10. Optional: In the Service account admins role field, add members that can manage the service account.

11. Click Done to finish creating the service account.

1. Create "Service Account"
2. Fill in the service account name => "Create"
3. Click "Select Role" and type in the 6 roles => "Done"

**1.** Service accounts + CREATE SERVICE ACCOUNT DELETE SHOW INFO PANEL

Service accounts for project "foundations-cicd"  
A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts](#).  
Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies](#).

Email	Status	Name	Description	Key ID	Key creation date	Actions
foundations-cicd-305109@appspot.gserviceaccount.com	✓	App Engine default service account		No keys		⋮

Create service account

① Service account details

Service account name: foundations-cicd-305109.iam.gserviceaccount.com

Service account description: Describe what this service account will do

CREATE

② Grant this service account access to project (optional)

③ Grant users access to this service account (optional)

DONE CANCEL

**2.** Create service account

① Service account details

Grant this service account access to project (optional)  
Grant this service account access to foundations-cicd so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Role: App Engine Admin Condition: Add condition

Role: Select a role Condition: Add condition

+ ADD ANOTHER ROLE

CONTINUE

② Grant users access to this service account (optional)

Storage Object Admin

Storage Object Admin Full control of GCS objects.

Environment and Storage Object Administrator Full control of Cloud Composer environments and Cloud Storage objects.

MANAGE ROLES

③ Grant users access to this service account (optional)

Grant this service account access to project (optional)  
Grant this service account access to foundations-cicd so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Role: App Engine Admin Condition: Add condition

Role: Storage Object Admin Condition: Add condition

Role: Storage Admin Condition: Add condition

Role: App Engine Service Admin Condition: Add condition

Role: Viewer Condition: Add condition

Role: Cloud Build Service Agent Condition: Add condition

MANAGE ROLES

**3.** Service accounts + CREATE SERVICE ACCOUNT DELETE SHOW INFO PANEL

Service accounts for project "foundations-cicd"  
A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts](#).  
Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies](#).

Email	Status	Name	Description	Key ID	Key creation date	Actions
foundations-cicd-305109@appspot.gserviceaccount.com	✓	App Engine default service account		No keys		⋮
se-01-192@foundations-cicd-305109.iam.gserviceaccount.com	✓	se-01		No keys		⋮

**Service Account (IAM)**

- Enable APP Engine API

**App Engine Admin API**  
Google

Provisions and manages developers' App Engine applications.

**ENABLE** TRY THIS API ↗

**OVERVIEW** DOCUMENTATION

**APIs & Services** App Engine Admin API Overview DISABLE API

**Details**

- Name: App Engine Admin API
- By: Google
- Service name: appengine.googleapis.com
- Overview: Provisions and manages developers' App Engine applications.
- Activation status: Enabled

**Tutorials and documentation**

- Learn more
- Try in API Explorer

**Traffic by response code**

Request/sec (2 hr average)

Response Code	Value
1.0%	1.0%
0.6%	0.6%
0.6%	0.6%
0.4%	0.4%
0.2%	0.2%
0	0

Jan 24 Jan 31 Feb 07 Feb 14

→ View metrics

- Enable Cloud Build API

**Cloud Build API**  
Google

Continuously build, test, and deploy.

**ENABLE** TRY THIS API ↗

Click to enable this API

**OVERVIEW** PRICING DOCUMENTATION

**APIs & Services** Cloud Build API Overview DISABLE API

**Details**

- Name: Cloud Build API
- By: Google
- Service name: cloudbuild.googleapis.com
- Overview: Creates and manages builds on Google Cloud Platform.
- Activation status: Enabled

**Tutorials and documentation**

- Quickstart
- Documentation
- Try in API Explorer

**Traffic by response code**

Request/sec (2 hr average)

Response Code	Value
1.0/s	1.0/s
0.8/s	0.8/s
0.6/s	0.6/s
0.4/s	0.4/s
0.2/s	0.2/s
0	0

Jan 24 Jan 31 Feb 07 Feb 14

→ View metrics

- Enable GCP Service: "App Engine" and "Service Account User"

**Cloud Build** Settings

**SERVICE ACCOUNT** DATA SHARING

**Service account permissions**

Cloud Build executes builds with the permissions granted to the [Cloud Build service account](#) tied to the project. You can grant additional roles to the service account to allow Cloud Build to interact with other GCP services.

Service account email: 195861820902@cloudbuild.gserviceaccount.com

GCP Service	Role	Status
Cloud Functions	Cloud Functions Developer	DISABLED
Cloud Run	Cloud Run Admin	DISABLED
App Engine	App Engine Admin	DISABLED
Kubernetes Engine	Kubernetes Engine Developer	DISABLED
Compute Engine	Compute Instance Admin (v1)	DISABLED
Firebase	Firebase Admin	DISABLED
Cloud KMS	Cloud KMS CryptoKey Decrypter	DISABLED
Secret Manager	Secret Manager Secret Accessor	DISABLED
Service Accounts	Service Account User	DISABLED

Roles not listed here can be managed in the [IAM section](#)

**Settings**

**SERVICE ACCOUNT** DATA SHARING

**Service account permissions**

Cloud Build executes builds with the permissions granted to the [Cloud Build service account](#) tied to the project. You can grant additional roles to the service account to allow Cloud Build to interact with other GCP services.

Service account email: 195861820902@cloudbuild.gserviceaccount.com

GCP Service	Role	Status
Cloud Functions	Cloud Functions Developer	DISABLED
Cloud Run	Cloud Run Admin	DISABLED
App Engine	App Engine Admin	DISABLED
Kubernetes Engine	Kubernetes Engine Developer	Enable
Compute Engine	Compute Instance Admin (v1)	Disable
Firebase	Firebase Admin	DISABLED
Cloud KMS	Cloud KMS CryptoKey Decrypter	DISABLED
Secret Manager	Secret Manager Secret Accessor	DISABLED
Service Accounts	Service Account User	DISABLED

Roles not listed here can be managed in the [IAM section](#)

**GCP Service** **Role** ? **Status**

GCP Service	Role	Status
Cloud Functions	Cloud Functions Developer	DISABLED
Cloud Run	Cloud Run Admin	DISABLED
App Engine	App Engine Admin	ENABLED
Kubernetes Engine	Kubernetes Engine Developer	DISABLED
Compute Engine	Compute Instance Admin (v1)	DISABLED
Firebase	Firebase Admin	DISABLED
Cloud KMS	Cloud KMS CryptoKey Decrypter	DISABLED
Secret Manager	Secret Manager Secret Accessor	DISABLED
Service Accounts	Service Account User	ENABLED

Roles not listed here can be managed in the [IAM section](#)

1. In the Cloud Console, go to the **Service Accounts** page.

[Go to the Service Accounts page](#)

2. Click **Select a project**, choose a project, and click **Open**.

3. Find the row of the service account that you want to create a key for. In that row, click the **More :** button, and then click **Create key**.

4. Select a **Key type** and click **Create**.

Clicking **Create** downloads a service account key file. After you download the key file, you cannot download it again.

Make sure to store the key file securely, because it can be used to authenticate as your service account. You can move and rename this file however you would like.

1. Go to Service Account page

2. Click on the 3 dots on the right side of the account with @appspot.gserviceaccount.com

3. Click Create Key

4. Choose JSON (It will be downloaded automatically to your local machine)

5. Save the file somewhere, so you remember where it is

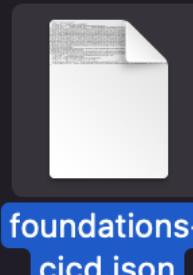
## Service accounts for project "foundations-cicd"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts](#).

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies](#).

Filter Filter table		Status	Name ↑	Description	Key ID	Key creation date	Actions
<input type="checkbox"/>	Email						
<input type="checkbox"/>	foundations-cicd-305109@appspot.gserviceaccount.com	✓	App Engine default service account	No keys	4e354693172eb807cdb58067988f90ffd97a171c	Feb 17, 2021	<input type="button" value="⋮"/>

DOWNLOADED AUTOMATICALLY



# 1. Copy the code in the downloaded JSON file

A screenshot of a code editor showing a JSON file named "foundations-cicd.json". The file contains a service account configuration. Several fields are highlighted in red, indicating they are sensitive or private. These include "client\_email", "client\_id", "auth\_uri", "token\_uri", "auth\_provider\_x509\_cert\_url", and "client\_x509\_cert\_url". A note at the bottom left says "ALSO PRIVATE =)". The entire file is numbered from 1 to 12.

```
1 {
2   "type": "service_account",
3   "project_id": "foundations-cicd",
4   "private_key_id": "PRIVATE ID", // REDACTED
5   "private_key": "PRIVATE KEY", // REDACTED
6   "client_email": "foundations-cicd-305109@appspot.gserviceaccount.com",
7   "client_id": "112215970595250767840",
8   "auth_uri": "https://accounts.google.com/o/oauth2/auth",
9   "token_uri": "https://oauth2.googleapis.com/token",
10  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
11  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/foundations-cicd-305109.appspot.gserviceaccount.com"
12 }
```

## 2. Go to your GitHub Repository => Settings => Secrets => New Repository Secret

A screenshot of the GitHub repository settings page for "Artificial-Ninelligence / foundations-sample-website". The "Settings" tab is selected. On the left, a sidebar lists various repository settings like Options, Manage access, Security & analysis, Branches, Webhooks, Notifications, Integrations, Deploy keys, Autolink references, Actions, Environments, Secrets, and Moderation settings. The "Secrets" section is currently active. The main area shows basic repository details: Repository name "foundations-sample-website", a "Template repository" checkbox, and a "Social preview" section with a placeholder image area and download template link.

A screenshot of the GitHub repository settings page for "Artificial-Ninelligence / foundations-sample-website", focusing on the "Actions secrets" section. The "Actions" tab is selected in the sidebar. The main area displays a message stating "There are no secrets for this repository's environments." Below it, there is a "Manage your environments and add environment secrets" button. Another section below is titled "Repository secrets" with the same message "There are no secrets for this repository." and a "Manage your environments and add environment secrets" button.

## 3. New Secret name => GAE\_SA\_KEY, value => Paste the code from JSON file

A screenshot of the GitHub Actions secrets creation dialog titled "Actions secrets / New secret". It has two sections: "Name" and "Value". The "Name" field is filled with "GAE\_SA\_KEY". The "Value" field contains the JSON code from the previous screenshot, which includes the redacted private key. The JSON code is as follows:

```
"client_email": "foundations-cicd-305109@appspot.gserviceaccount.com",
"client_id": "112215970595250767840",
"auth_uri": "https://accounts.google.com/o/oauth2/auth",
"token_uri": "https://oauth2.googleapis.com/token",
"auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
"client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/foundations-cicd-305109.appspot.gserviceaccount.com"
```

## 1. Clone the repository via HTTPS

This branch is 1 commit ahead, 4 commits behind DrAdamRoe:main.

Clone  
HTTPS SSH GitHub CLI  
https://github.com/Artificial-Ninol...  
Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop  
Download ZIP

2 months ago

## 2. Go to GCP Dashboard with your newly created project and clone the repo via cloud shell

Cloud Shell

Clone your repo: git clone + paste

Welcome to Cloud Shell! Type "help" to get started.  
Your Cloud Platform project in this session is set to foundations-cicd-305109.  
Use "gcloud config set project [PROJECT\_ID]" to change to a different project.  
nino\_lindenberg@cloudshell:~/(foundations-cicd-305109)\$ git clone https://github.com/Artificial-Ninoligence/foundations-sample-website-.git

## 3. Go to the newly cloned repository (Directory)

```
nino_lindenberg@cloudshell:~/(foundations-cicd-305109)$ ls
foundations-sample-website- README-cloudshell.txt
nino_lindenberg@cloudshell:~/(foundations-cicd-305109)$ cd foundations-sample-website-/
nino_lindenberg@cloudshell:~/foundations-sample-website- (foundations-cicd-305109)$
```

## 4. Deploy your app

- "gcloud app deploy --quiet app.yaml --project YOURPROJECTID --promote

```
nino_lindenberg@cloudshell:~/foundations-sample-website- (foundations-cicd-305109)$ gcloud app deploy --quiet app.yaml --project foundations-cicd-305109 --promote
Services to deploy:
descriptor: [/home/nino_lindenberg/foundations-sample-website-/app.yaml]
source: [/home/nino_lindenberg/foundations-sample-website-]
target project: [foundations-cicd-305109]
target service: [default]
target version: [20210217t192332]
target url: [https://foundations-cicd-305109.ey.r.appspot.com]

Beginning deployment of service [default]...
Uploading 1 file to Google Cloud Storage
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://foundations-cicd-305109.ey.r.appspot.com]

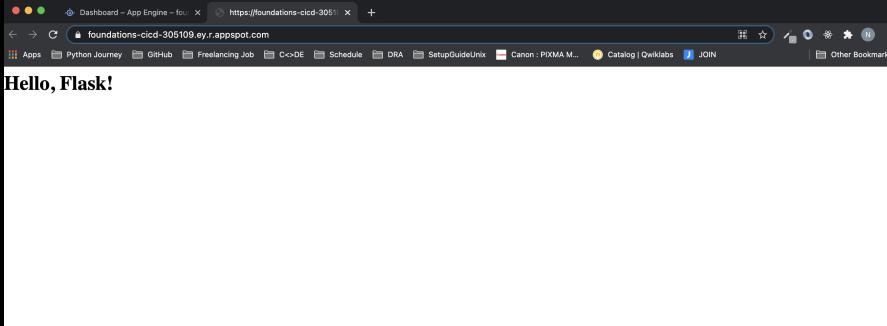
You can stream logs from the command line by running:
$ gcloud app logs tail -s default

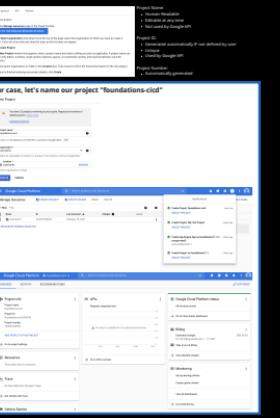
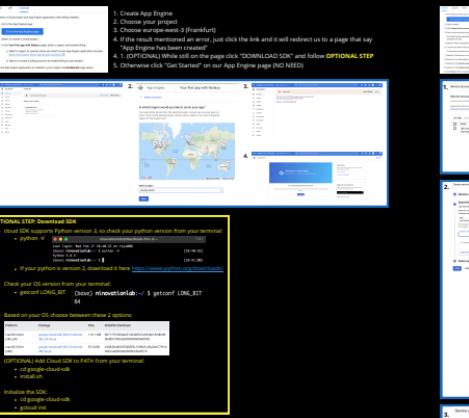
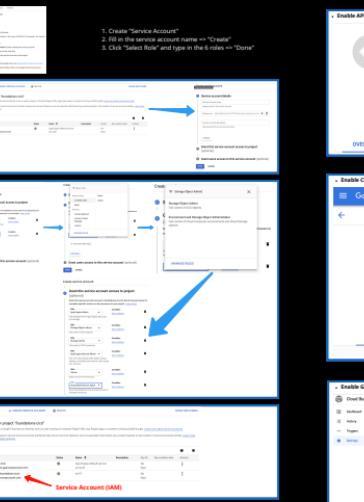
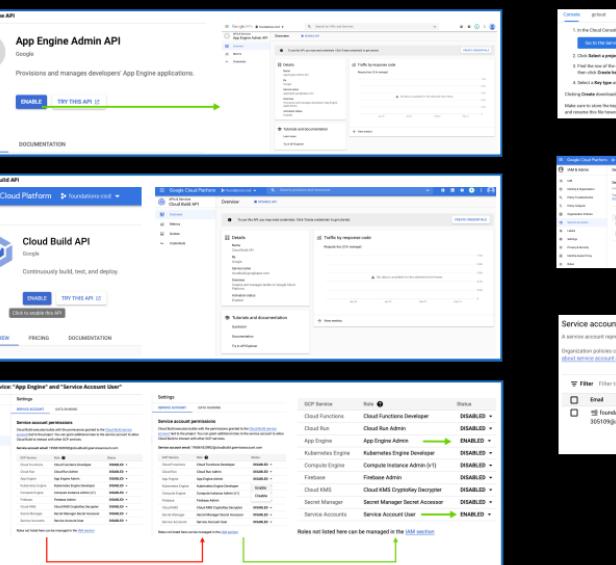
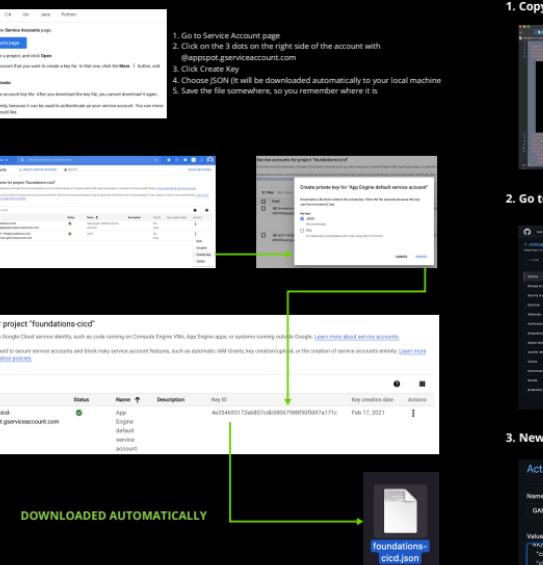
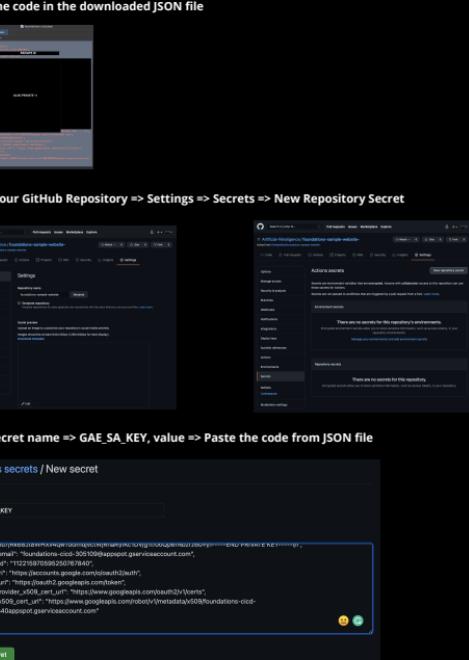
To view your application in the web browser run:
$ gcloud app browse
nino_lindenberg@cloudshell:~/foundations-sample-website- (foundations-cicd-305109)$
```

- Check your Web App: "gcloud app browse"

```
nino_lindenberg@cloudshell:~/foundations-sample-website- (foundations-cicd-305109)$ gcloud app browse
Did not detect your browser. Go to this link to view your app:
https://foundations-cicd-305109.ey.r.appspot.com
```

- Copy paste the bottom https or cmd + click



**STEP 1: Creating a new Project in GCP****STEP 2: Creating the App Engine****STEP 3: Creating Service Account and Managing its Role****STEP 4: APP Engine API, Cloud Build API, GCP Services****STEP 5: Service Account Key****STEP 6: Add Secret to GitHub Repository Settings****STEP 7: CLONE REPO via HTTPS & DEPLOY APP**