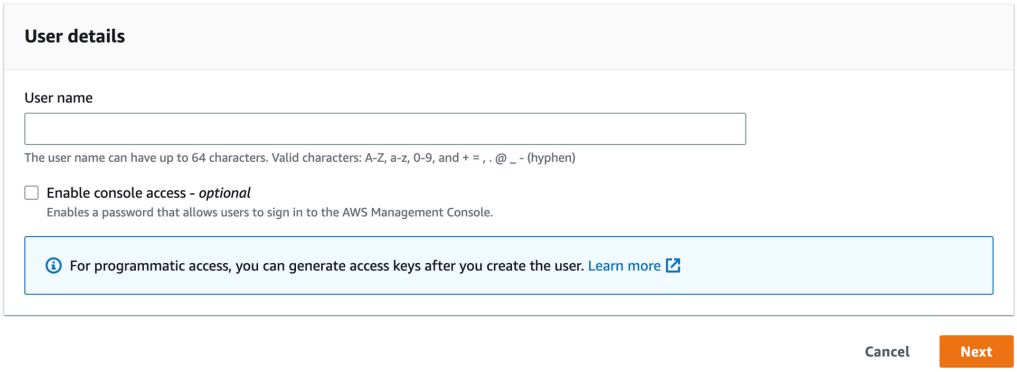
Steps to implement Hands-on Project - Mission 2

Amazon Web Services

- Access AWS console and go to IAM service
- Under Access management, Click in "Users", then "Add users". Insert the User name **luxxy-covid-testing-system-en-app1** and click in **Next** to create a programmatic user.

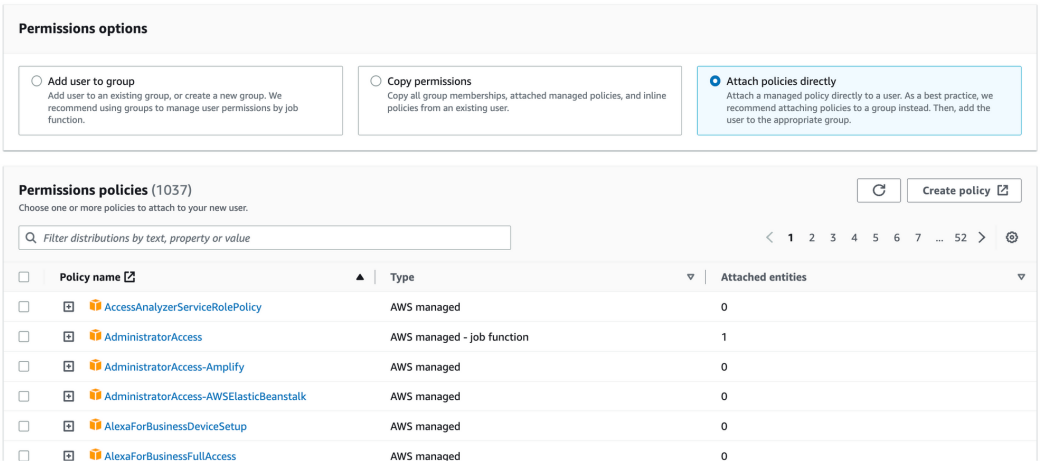
Specify user details



- On Set permissions, Permissions options, click in "Attach policies directly" button.

Set permissions

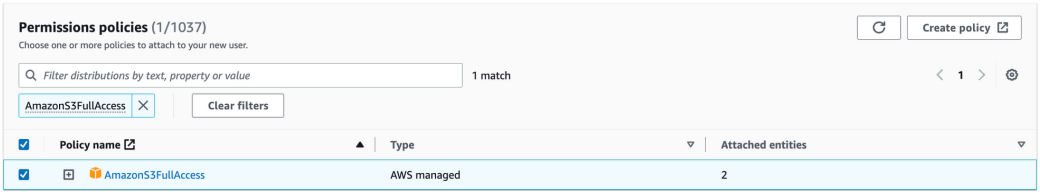
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more



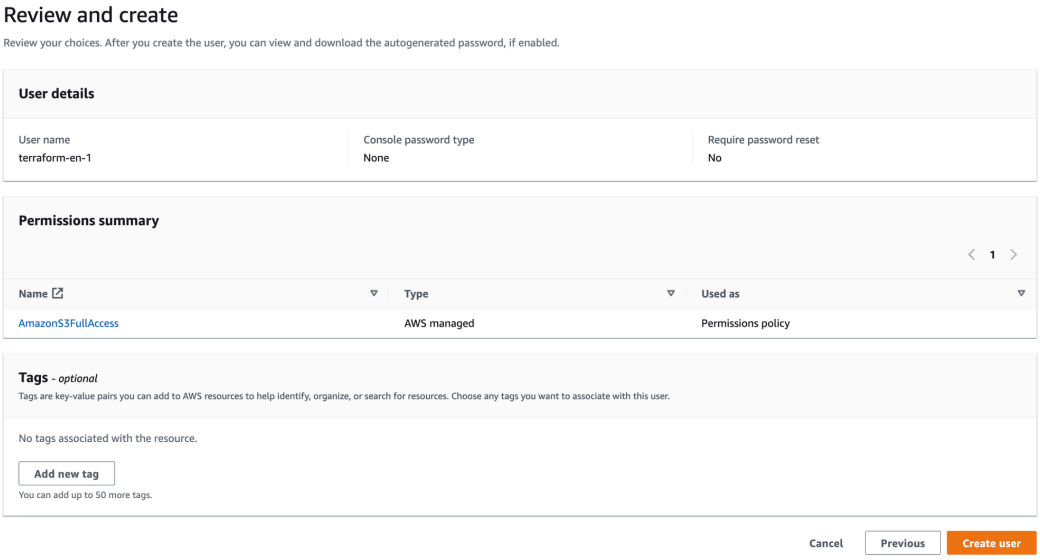
- Type **AmazonS3FullAccess** in **Filter distributions by text, property or**

value, press Enter.

- Select **AmazonS3FullAccess**

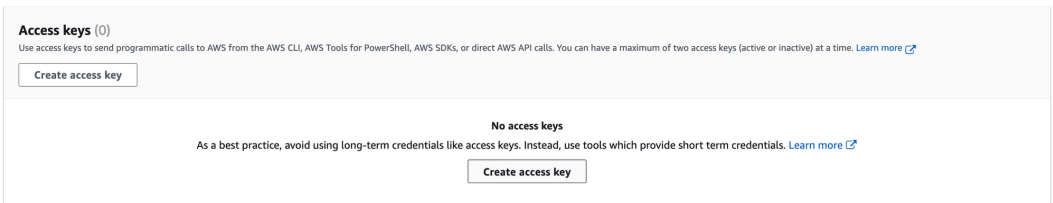


- Click in **Next**
- Review all details and click in Create user



Steps to create access key:

- Click on the user you have created.
- Go to Security credentials tab.
- Scroll down and go to Access keys section.
- Click on Create access key

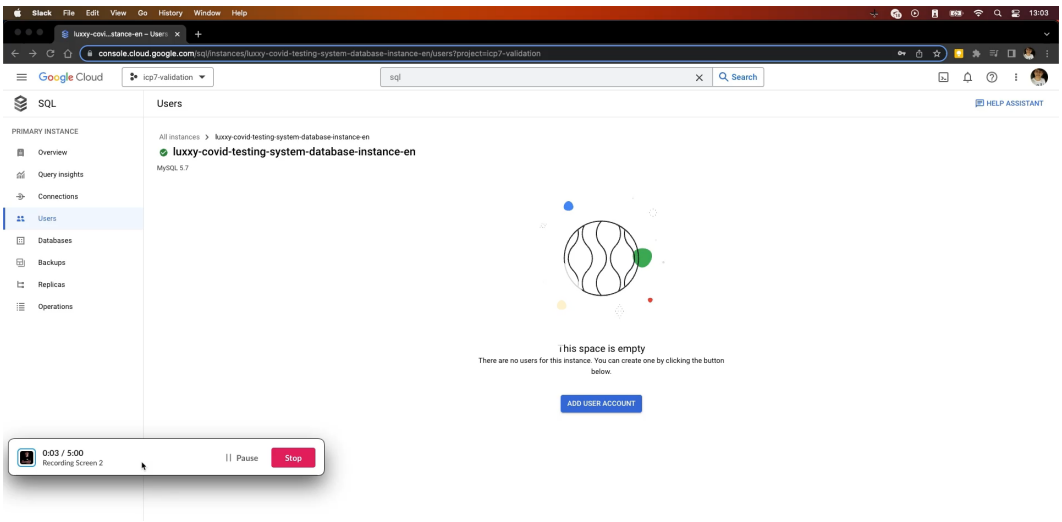


- Select **Command Line Interface (CLI)** and **I understand the above recommendation and want to proceed to create an access key** checkbox.
- Click Next

- Click Next
- Click on Create access key
- Click on Download .csv file
- After download, click Done.
- Now, rename .csv file downloaded to **luxxy-covid-testing-system-en-app1.csv**

Google Cloud Platform (GCP)

- Navigate to Cloud SQL instance and create a new user **app** with password **welcome123456** on Cloud SQL MySQL database



0:00 / 0:20

- Connect to Google Cloud Shell
- Download the mission2 files to Google Cloud Shell using the wget command as shown below

```
cd mkdir mission2_en cd mission2_en wget https://tcb-public-events.s3.amazonaws.com/icp/mission2.zip unzip mission2.zip
```

- Connect to MySQL DB running on Cloud SQL (once it prompts for the password, provide **welcome123456**)

```
mysql --host=<public_ip_cloudsql> --port=3306 -u app -p
```

- Once you're connected to the database instance, create the products table for testing purposes

```
use dbcovidtesting; source ~/mission2_en/mission2/en/db
/create_table.sql; show tables; exit;
```

- Enable Cloud Build API via Cloud Shell.

```
# Command to enable Cloud Build API gcloud services enable
cloudbuild.googleapis.com
```

Known issue during this step

If you see the error below, please follow the steps to fix it:

```
ERROR: (gcloud.builds.submit) INVALID_ARGUMENT: could not resolv
e source: googleapi: Error 403: 989404026119@cloudbuild.gservice
account.com does not have storage.objects.get access to the Goog
le Cloud Storage object., forbidden To solve it: 1. Access IAM &
Admin; 2. Click on your Cloud Build Service Account Example: 989
404026119@cloudbuild.gserviceaccount.com Cloud Build Service Acc
ount 3. On your Cloud Build Service Account, right side, click o
n Edit principal 4. Click on Add another role 5. Click on Select
Role, and filter by Storage Admin or gcs. Select Storage Admin
(Full control of GCS resources). 6. Click on Save and go to Clou
d Shell.
```

- Build the Docker image and push it to Google Container Registry. Please replace the <PROJECT_ID> with your My First Project ID.

```
cd ~/mission2_en/mission2/en/app gcloud builds submit --tag
gcr.io/<PROJECT_ID>/luxxy-covid-testing-system-app-en
```

- Open the Cloud Editor and edit the Kubernetes deployment file (luxxy-covid-testing-system.yaml) and update the variables below in

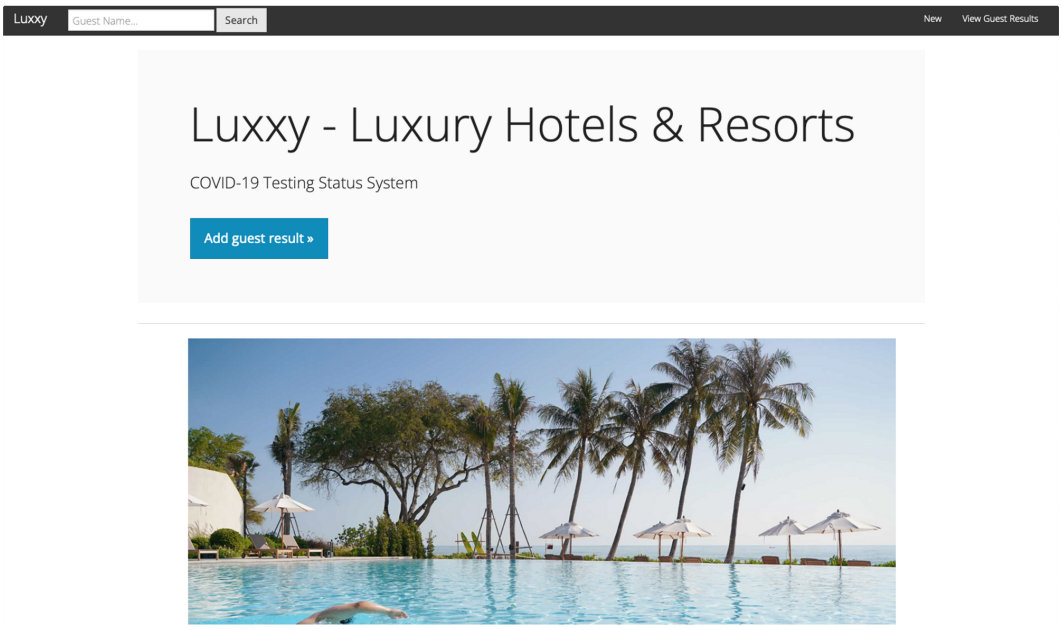
red with your <PROJECT_ID> on the Google Container Registry path, AWS Bucket name, AWS Keys (from **luxxy-covid-testing-system-en-app1.csv**) and Cloud SQL Database Private IP.

```
cd ~/mission2/en/kubernetes luxxy-covid-testing-system.yaml
image: gcr.io/<PROJECT_ID>/luxxy-covid-testing-system-app-
en:latest ... - name: AWS_BUCKET value: "luxxy-covid-testing-
system-pdf-en-xxxx" - name: S3_ACCESS_KEY value:
"xxxxxxxxxxxxxxxxxxxx" - name: S3_SECRET_ACCESS_KEY value:
"xxxxxxxxxxxxxxxxxxxx" - name: DB_HOST_NAME value: "172.21.0.3"
```

- Connect to the GKE (Google Kubernetes Engine) cluster via Console (follow the video)
- Deploy the application Luxxy in the Cluster

```
cd ~/mission2_en/mission2/en/kubernetes kubectl apply -f luxxy-
covid-testing-system.yaml
```

- Get the Public IP and test the application (CLICK HERE to download COVID-19 Testing result sample)
- You should see the app up & running! Congrats!



© The Cloud Bootcamp. All rights reserved (Created for educational purposes only - prohibited production)