Session 3 Lab: Deploy to Cloud Run with GitHub Actions

Prerequisites

1. To complete this lab, you must have done the prior two labs.

   If you have not already done so, complete it using the following link: [Session 1 Lab: Creating DevOps Pipelines with GitHub Actions]()

   Also, if you have not already done so, complete it using the following link: [Session 2 Lab: Building Docker Images with GitHub Actions]()

Adding Google Cloud Credentials

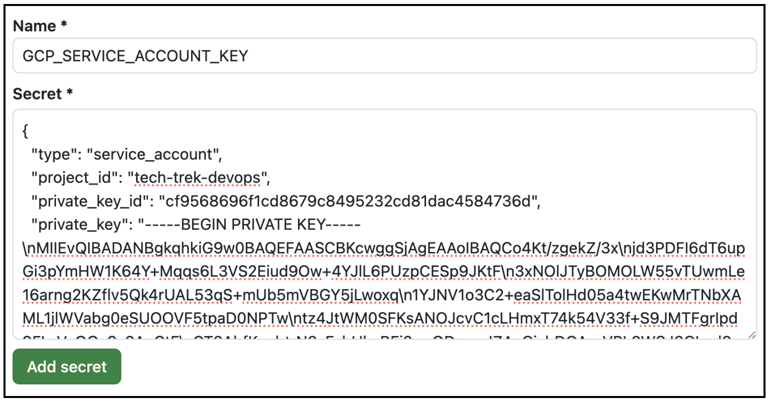
1. Go to https://github.com, sign in, and open the repository you created in the previous labs.

2. Create a new branch called `session-3`. Make sure to spell the branch exactly because the name will be used later in your code.

3. On the GitHub website, click the **Settings** link.

4. From the Security section on the left, select **Secrets and variables**, and then **Actions**.

5. Click the **New repository secret** button. Create a secret called `GCP_SERVICE_ACCOUNT_KEY`. For the value of the secret, you will need to paste in the service account key provided by your instructor. The screen will look similar to the screenshot below.



Name *

GCP_SERVICE_ACCOUNT_KEY

Secret *

```
{
  "type": "service_account",
  "project_id": "tech-trek-devops",
  "private_key_id": "cf9568696f1cd8679c8495232cd81dac4584736d",
  "private_key": "-----BEGIN PRIVATE KEY-----
\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCo4Kt/zgekZ/3x\njd3PDFl6dT6up
Gi3pYmHW1K64Y+Mqqs6L3VS2Eiud9Ow+4YJlL6PUzpCESp9JKtF\n3xNOlJTyBOMOLW55vTUwmLe
16arng2KZflv5Qk4rUAL53qS+mUb5mVBGY5jLwoxq\n1YJNV1o3C2+eaSlTolHd05a4twEKwMrTNbXA
ML1jlWVabg0eSUOOVF5tpaD0NPTw\ntz4JtWM0SFKsANOJcvC1cLHmxT74k54V33f+S9JMTFgrlpd
```

Note: If you want to use your own Google Cloud project, you can create a Google Cloud

Service account with the permissions to deploy to Cloud Run and generate a JSON key for that account. However, this is beyond the scope of this training.

Extending the CI/CD Pipeline to Deploy to Cloud Run

1. Return to your GitHub repository **Code** view. Make sure you are using the `session-3` branch and navigate to the file `.github/workflows/run-tests.yml`. Open the file in **Edit** mode by clicking on the pencil icon. Replace the current contents of the file with the following code.

Python

```
name: Test-Build-Deploy Pipeline

on:
  push:
    branches:
      - session-3
      - main
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Set up Python
        uses: actions/setup-python@v5
        with:
          python-version: '3.11'

      - name: Install dependencies
        run: |
          python -m pip install --upgrade pip
          pip install -r requirements.txt

      - name: Run tests
        run: |
          pytest --maxfail=1 --disable-warnings

  build_and_push:
    runs-on: ubuntu-latest
    needs: test
    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v2

      - name: Log in to Docker Hub
        uses: docker/login-action@v2
        with:
          username: ${{ secrets.DOCKER_HUB_USERNAME }}
          password: ${{ secrets.DOCKER_HUB_ACCESS_TOKEN }}

      - name: Build and push Docker image
        uses: docker/build-push-action@v5
        with:
          context: .
          push: true
          tags: ${{ secrets.DOCKER_HUB_USERNAME }}/tech-trek:${{ github.sha }}
  deploy:
    runs-on: ubuntu-latest
    needs: build_and_push
    steps:
      - name: Authenticate to Google Cloud
        uses: google-github-actions/auth@v1
        with:
          credentials_json: ${{ secrets.GCP_SERVICE_ACCOUNT_KEY }}

      - name: Deploy to Cloud Run
        uses: google-github-actions/deploy-cloudrun@v1
        with:
          service: ${{ secrets.DOCKER_HUB_USERNAME }}-tech-trek
          image: docker.io/${{ secrets.DOCKER_HUB_USERNAME }}/tech-trek:${{
github.sha }}
          region: us-central1
          flags: "--allow-unauthenticated"

      - name: Get Cloud Run Service URL
        run: |
          URL=$(gcloud run services describe "${{ secrets.DOCKER_HUB_USERNAME
}}-tech-trek" --region us-central1 --format 'value(status.url)')
          echo "Cloud Run service URL: $URL"
          echo "SERVICE_URL=$URL" >> $GITHUB_ENV
```

Note: This workflow now has three jobs: `test`, `build_and_push`, and `deploy`. We are interested in the third job for this session. The first two jobs were covered in previous sessions.

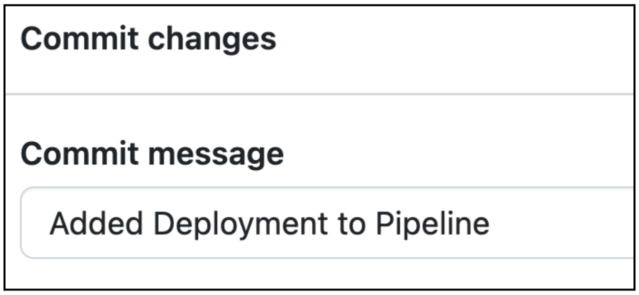
2. The first step of the deploy job uses the GitHub Google Cloud authorization action. Note that it uses the secret you just created, which includes the service account key.

3. The second step of the deploy job uses the Deploy to Cloud Run GitHub action. The name of the service will be your docker username plus the string "-tech-trek." Also, note that you are deploying the Docker image created in the previous job.

4. The third step is just to output the URL of the Cloud Run service that is being deployed. You will see the results after the pipeline executes.

5. Click the **Commit changes** button near the top of the page to save the changes to the pipeline code.

   Set the commit message to `Added Deployment to Pipeline` as shown in the screenshot below.

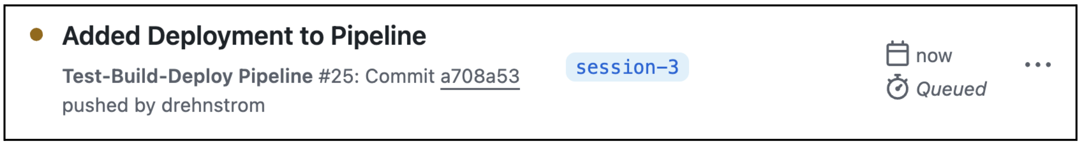
   Commit changes

   Commit message

   Added Deployment to Pipeline

Monitoring the Pipeline

1. Click the **Actions** menu at the top. Your pipeline will appear as shown below.

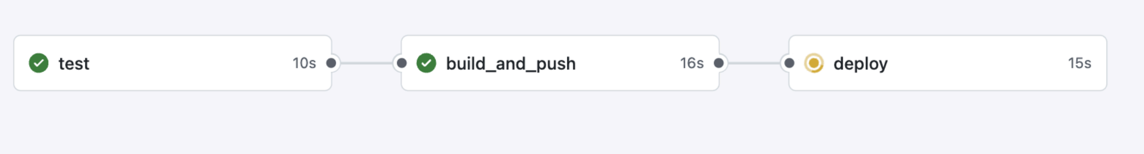


Added Deployment to Pipeline

Test-Build-Deploy Pipeline #25: Commit a708a53 session-3 now

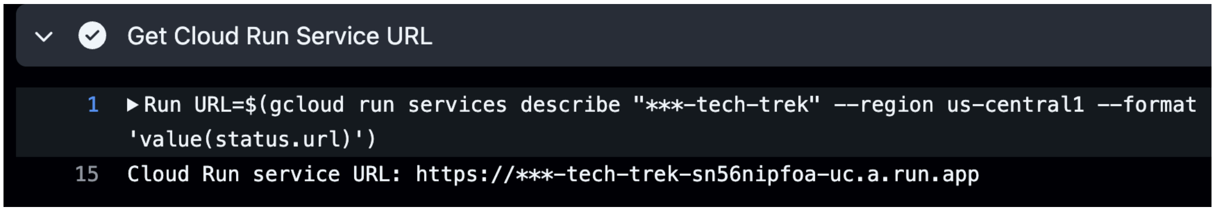
pushed by drehnstrom Queued

2. Click on the pipeline and you will be able to monitor the job progress.



3. Click the third job, deploy. Expand the steps and read the output. When the job is completed, the step **Get Cloud Run Service URL** will contain the link to your service. Note that your Docker username is masked with asterisks, as shown below. This is a security feature to prevent sensitive data from being added to any logs.

   Copy and paste the URL into a text file, replacing the asterisks with your Docker hub username. Then, test the URL in your browser. It should return a version number.

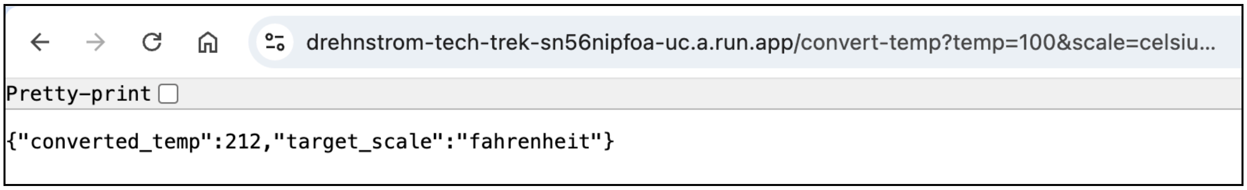


Get Cloud Run Service URL

```
▶Run URL=$(gcloud run services describe "***-tech-trek" --region us-central1 --format
'value(status.url)')
Cloud Run service URL: https://***-tech-trek-sn56nipfoa-uc.a.run.app
```

4. Add the following string to the end of the URL and it should return a temperature conversion, as shown in the screenshot.

```
Unset
/convert-temp?temp=100&scale=celsius&target_scale=fahrenheit
```



drehnstrom-tech-trek-sn56nipfoa-uc.a.run.app/convert-temp?temp=100&scale=celsiu...

Pretty-print □

```
{"converted_temp":212,"target_scale":"fahrenheit"}
```

Merge your Changes

1. Create a **Pull request** and merge the changes you made on the `session-3` branch with the `main` branch.

2. After you have merged the changes, go to the **Actions** menu. You should see that your pipeline runs again.