**Task 1. Create a Cloud Workstations configuration**

Cloud Workstations cluster named cepf-workstation-cluster is pre-created for you.

Create a configuration for a Cloud Workstations instance named cepf-workstation-config with Base Editor (Code OSS) as the base image.

Click **Check my progress** to verify your progress.

Create a Cloud Workstations configuration

Check my progress

**Task 2. Create a Cloud Workstations instance**

Create a Cloud Workstations instance named cepf-workstation using the configuration you created in the previous step.

Click **Check my progress** to verify your progress.

Create a Cloud Workstations instance

Check my progress

**Task 3. Use Cloud Build to build a container application**

1. Download the application code to Cloud Workstations from the following application code location: gs://cloud-training/cepf/cepf023/cepf023-app-code.zip.
2. Create a Docker format Artifact Registry called cepf-repo.
3. Build the application from the provided Dockerfile and push it to the Artifact Repository using Cloud Build.

Your Cloud Build should run in the Default region region.

Click **Check my progress** to verify your progress.

Use Cloud Build to build a container application

Check my progress

**Task 4. Use Google Cloud Deploy to deploy an application to Cloud Run**

Deploy the application you have built in previous step to Cloud Run using Google Cloud Deploy.

1. Create a Skaffold configuration.
2. Create service definition files for two Cloud Run services; cepf-dev-service and cepf-prod-service.
3. Define your Google Cloud Deploy delivery pipeline called cepf-run-app-pipeline with two deployment targets cepf-dev-service and cepf-prod-service, which point to the two services.
4. Instantiate your delivery pipeline by creating a release, which automatically deploys to the first target, cepf-dev-service.
5. Allow unauthenticated invocations on the cepf-dev-service Cloud Run service.
6. Verify the application by visiting the cepf-dev-service Cloud Run service URL.

**Note:** Make sure you create the pipeline and Cloud Run services in the Default region region.

It should show the **Oops, something went wrong...** webpage.

Click **Check my progress** to verify your progress.

Use Google Cloud Deploy to deploy an application to Cloud Run

Check my progress

**Task 5. Fix any issues and redeploy the app**

The application you deployed in the previous step is not working as expected because you deployed the wrong (index\_v1.html) HTML file.

1. To fix this issue, update the Dockerfile to use the correct (index\_v2.html) HTML file and build the application image again using Cloud Build.
2. Abandon the older release, which deployed the incorrect application.
3. Instantiate your delivery pipeline by creating a new release, which automatically deploys the new image to the first target, cepf-dev-service.
4. Verify the application by visiting the cepf-dev-service Cloud Run service URL.

It should show the **It's running!** webpage.

Click **Check my progress** to verify your progress.

Fix the issue and redeploy the app

Check my progress

**Task 6. Promote the release to the production environment once the app passes verification**

1. Now that the correct application is deployed on cepf-dev-service, promote the release to cepf-prod-service.
2. Allow unauthenticated invocations on the cepf-prod-service Cloud Run service.
3. Verify the application by visiting the cepf-prod-service Cloud Run service URL.

It should show the **It's running!** webpage.

Click **Check my progress** to verify your progress.

Promote the release to the production environment once the app passes verification

Check my progress

**Task 7. Configure the load balancer for the Cloud Run app**

1. Restrict ingress for Cloud Run to **Internal and Cloud Load Balancing**.
2. Set up a global external HTTP load balancer to route requests to Cloud Run.
3. Set the Frontend forwarding rule name to cepf-frontend.
4. Verify the application is working by accessing it via the load balancer IP address.

Click **Check my progress** to verify your progress.

Configure the load balancer for the Cloud Run app

Check my progress

**Congratulations!**

gs://cloud-training/cepf/cepf023/cepf023-app-code.zip

gsutil cp gs://cloud-training/cepf/cepf023/cepf023-app-code.zip

gcloud artifacts repositories create cepf-repo \  
    --repository-format=docker \  
    --location=us-east1 \  
    --description="Repository for containerized applications"

gcloud artifacts repositories list --location us-west1

- name: 'gcr.io/cloud-builders/docker'  
  args: ['build', '-t', 'us-west1 docker.pkg.dev qwiklabs-gcp-01-68f6ef4d95c9cepf-repo APP\_INNOVA'', '.']  
- name: 'gcr.io/cloud-builders/docker'  
  args: ['push', 'us-west1 docker.pkg.dev qwiklabs-gcp-01-68f6ef4d95c9cepf-repo/APP\_INNOVA']

images:  
- 'us-west1-docker.pkg.dev/ qwiklabs-gcp-01-68f6ef4d95c9/cepf-repo/APP\_INNOVA'

gcloud deploy releases create cepf-release-001 \  
  --delivery-pipeline=cepf-run-app-pipeline \  
  --region=us-west1

gcloud builds submit --region=us-east1 --config=cloudbuild.yaml .

gcloud deploy releases create release-1 \  
  --pipeline=cepf-run-app-pipeline \  
  --region=us-east1 \  
  --images=gcr.io/ qwiklabs-gcp-02-fb4a89231e4a /cepf-run-app=gcr.io/ qwiklabs-gcp-02-fb4a89231e4a /cepf-run-app:latest

gcloud deploy releases create release-1 \  
  --delivery-pipeline=cepf-run-app-pipeline \  
  --region=us-east1 \  
  --images=gcr.io/[PROJECT\_ID]/cepf-run-app=gcr.io/ qwiklabs-gcp-02-fb4a89231e4a /cepf-run-app:latest

gcloud deploy releases create my-release \  
    --delivery-pipeline= cepf-run-app-pipeline \  
    --region=us-east1 \  
    --annotations="description='First deployment'" \  
    --verbosity=debug

gcloud deploy releases create my-release \  
    --delivery-pipeline=cepf-run-app-pipeline \  
    --region=us-east1 \  
    --annotations="description='First deployment'" \  
    --verbosity=debug

curl -Lo skaffold <https://storage.googleapis.com/skaffold/releases/latest/skaffold-linux-amd64>  
chmod +x skaffold  
sudo mv skaffold /usr/local/bin

gcloud artifacts repositories create cepf-repo \  
    --repository-format=docker \  
    --location=us-east1 \  
    --description="Repository for containerized applications"