- Create a new project (e.g., `gcp-demo-project`).
- Enable APIs
  Cloud Build API
  Cloud Deploy API
  Artifact Registry API

- Install Google Cloud SDK
- Authenticate and Set Project

```
Unset
gcloud auth login
gcloud config set project gcp-demo-project
```

- Create Application Directory

```
Unset
mkdir gcp-demo-app
cd gcp-demo-app
```

- Create repo on github for application (app.js)

```
JavaScript
const express = require('express');
const app = express();
const PORT = process.env.PORT || 8080;

app.get('/', (req, res) => {
    res.send('Hello, World! This is a GCP demo app.');
});

app.listen(PORT, () => {
    console.log(`Server is running on port ${PORT}`);
});
```

- Dockerfile

```
Unset
# Use the official Node.js image.
```

```
FROM node:14

# Set the working directory.
WORKDIR /app

# Copy package.json and install dependencies.
COPY package*.json ./
RUN npm install

# Copy the rest of the application code.
COPY . .

# Expose the port the app runs on.
EXPOSE 8080

# Command to run the application.
CMD ["node", "app.js"]
```

- Create `package.json`

```
Unset
{
  "name": "gcp-demo-app",
  "version": "1.0.0",
  "main": "app.js",
  "scripts": {
    "start": "node app.js"
  },
  "dependencies": {
    "express": "^4.17.1"
  }
}
```

- Create an Artifact Registry Repository manually
- Create a trigger and connect to github
- Configure Cloud Build

cloudbuild.yaml

```
Unset
steps:
- name: 'gcr.io/cloud-builders/docker'
  args: ['build', '-t',
'us-central1-docker.pkg.dev/PROJECT_ID/gcp-demo-repo/gcp-demo-app', '.']
- name: 'gcr.io/cloud-builders/docker'
  args: ['push',
'us-central1-docker.pkg.dev/PROJECT_ID/gcp-demo-repo/gcp-demo-app']
images:
- 'us-central1-docker.pkg.dev/PROJECT_ID/gcp-demo-repo/gcp-demo-app'

options:
  logging: CLOUD_LOGGING_ONLY
```

- Replace `PROJECT_ID` with your Google Cloud Project ID.
- Start the build