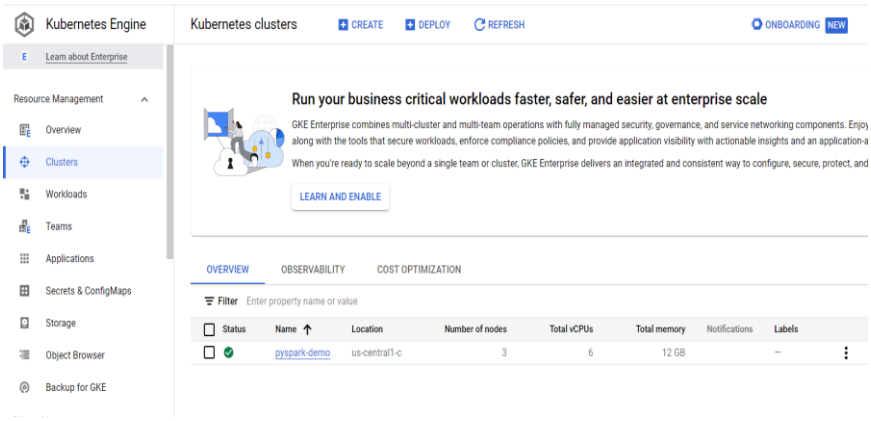
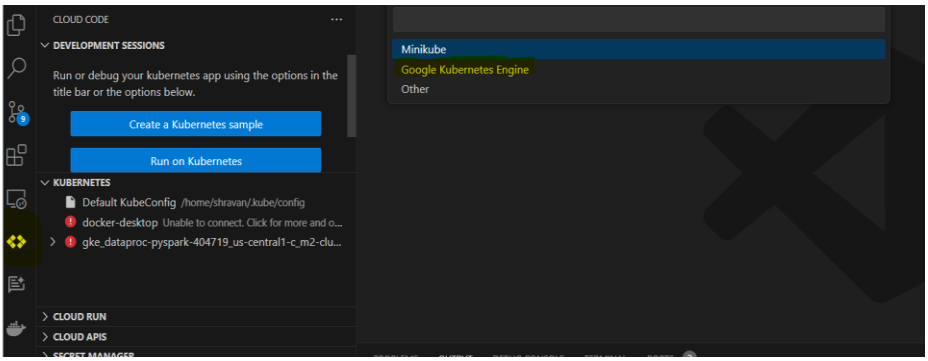
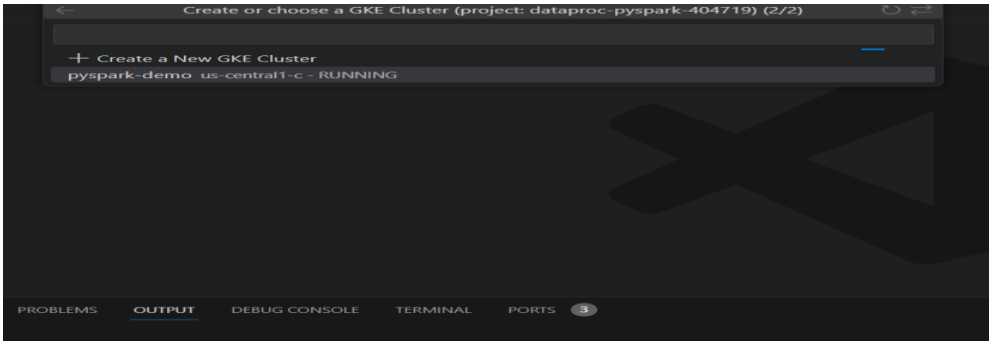
i) Create Kubernetes Cluster with Default Config



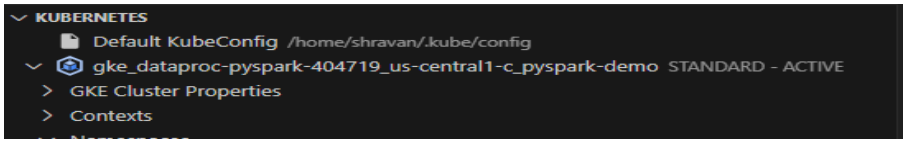

ii) Use Cloud Code plugin for VS Code and Sign in to Cloud Code with your GCP cloud account. Add your Kubernetes as shown



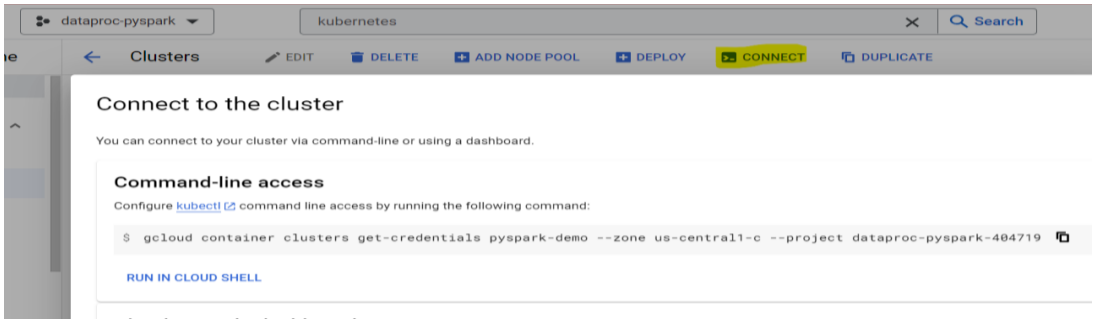

iii) You should see the cluster you created as shown below in the prompt

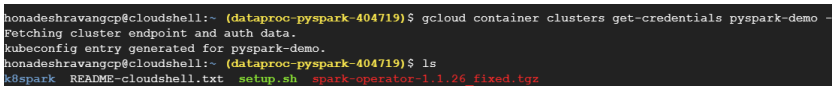



iv) Notice your gke cluster has state as active

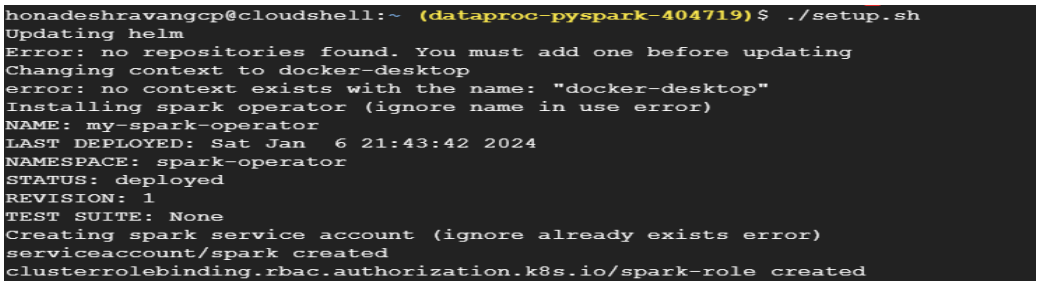


v) Connect to your cluster and upload setup and spark-connector zip using google cloud shell.




vi) Run setup.sh after upload setup and spark-connector zip – It creates some roles and installs spark-connector for Kubernetes




vii) Build Image and Push to Artifact repository. (You should have docker installed.)


viii) Edit Image Config in .yaml files to change to url of our artifact repository

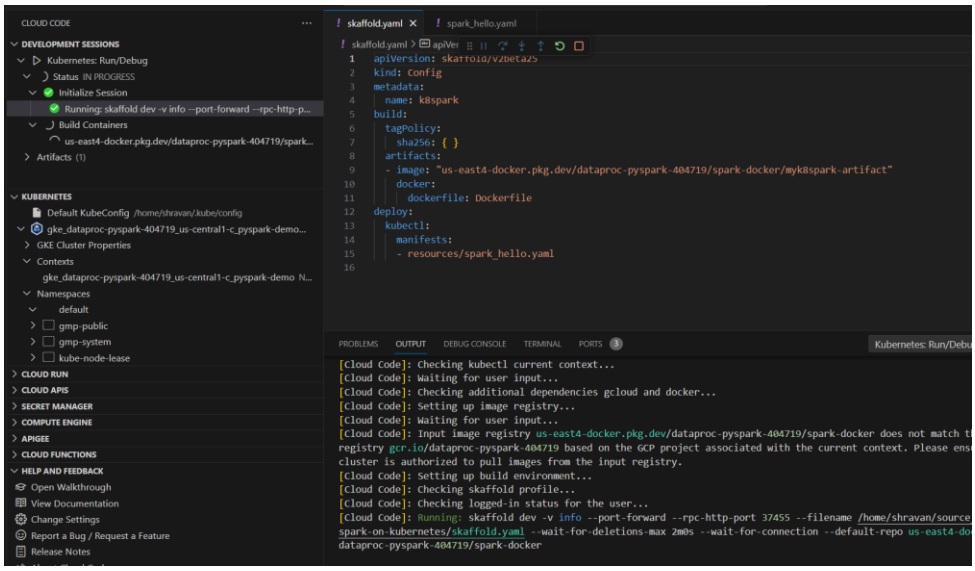
skaffold.yaml

```yaml
apiVersion: skaffold/v2beta25
kind: Config
metadata:
  name: k8spark
build:
  tagPolicy:
    sha256: { }
  artifacts:
  - image: "us-east4-docker.pkg.dev/dataproc-pyspark-404719/spark-docker/myk8spark-artifact"
    docker:
      dockerfile: Dockerfile
deploy:
  kubectl:
    manifests:
    - resources/spark_hello.yaml
```

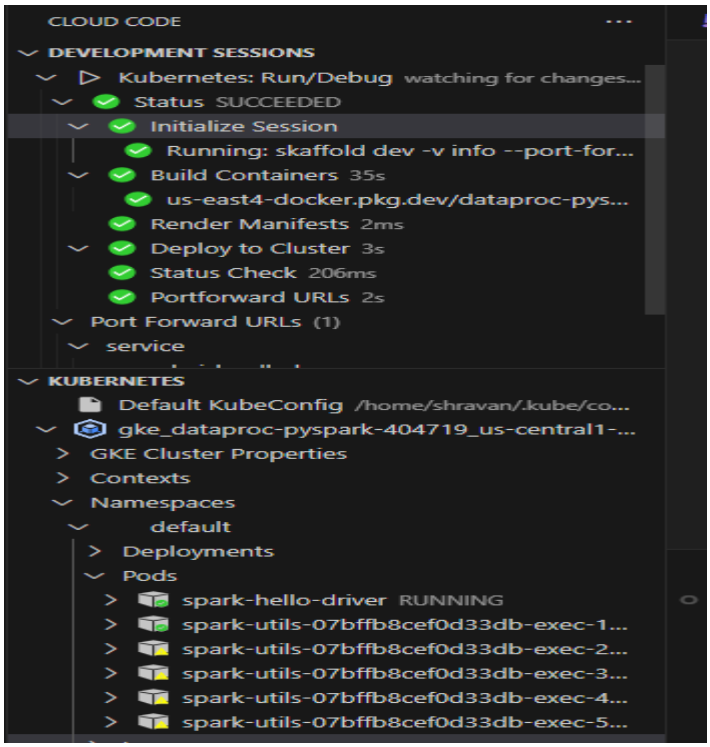
spark_hello.yaml

```yaml
apiVersion: "sparkoperator.k8s.io/v1beta2"
kind: SparkApplication
metadata:
  name: spark-hello
  namespace: default
  labels:
    app: spark-hello
spec:
  type: Python
  pythonVersion: "3"
  mode: cluster
  image: "us-east4-docker.pkg.dev/dataproc-pyspark-404719/spark-docker/myk8spark-artifact"
  # https://skaffold.dev/docs/environment/local-cluster/
  #  Skaffold's direct loading of images into a local cluster does mean that resources specifying an
  # imagePullPolicy: Always may fail as the images are not be pushed to the remote registry.
  # On Docker for Desktop, don't specify imagePullPolicy
  imagePullPolicy: Always
  mainApplicationFile: local:///opt/spark/work-dir/runpi.py
  sparkConf:
    "spark.ui.port": "4040"
  sparkVersion: "3.2"
  restartPolicy:
    type: Never
  driver:
    coreLimit: "1"
```

ix) Click on Run on Kubernetes and process should begin

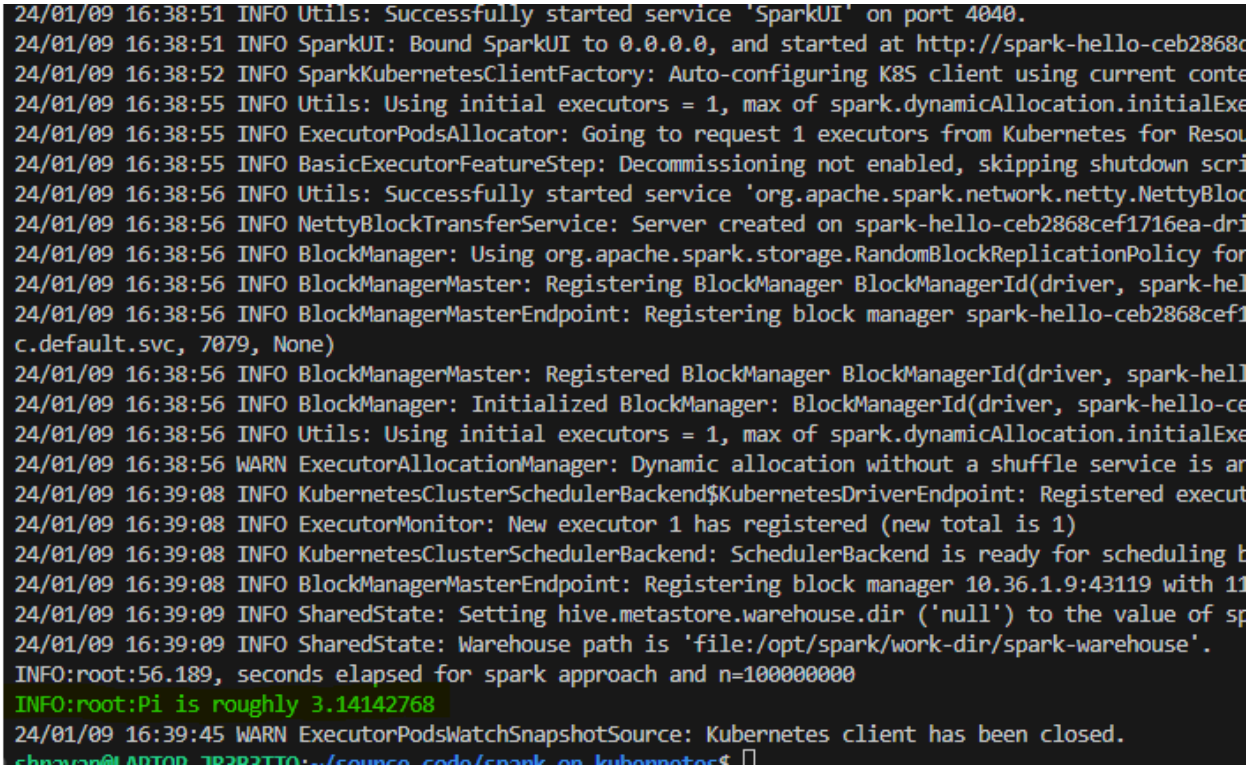



x) Check whether your builds are success as show in figure and whether driver and executor pods gets spun




xi) Check logs of your driver to check for success of your program

kubectl logs <pod-name>




Value of pi is calculated indicating success of our pyspark on a Kubernetes cluster.