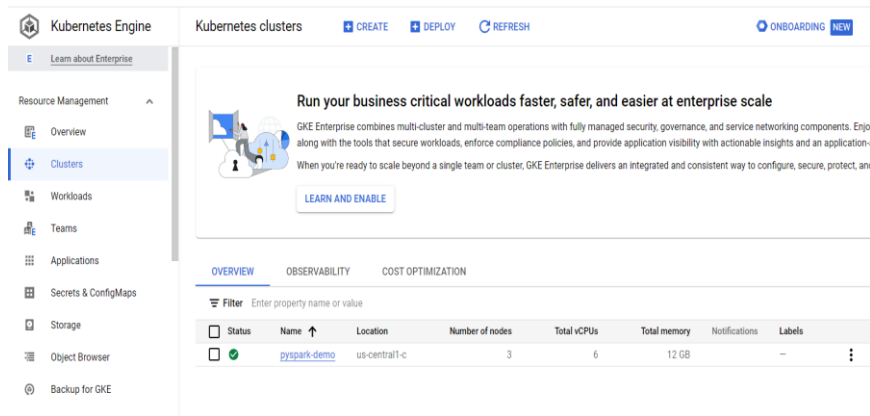
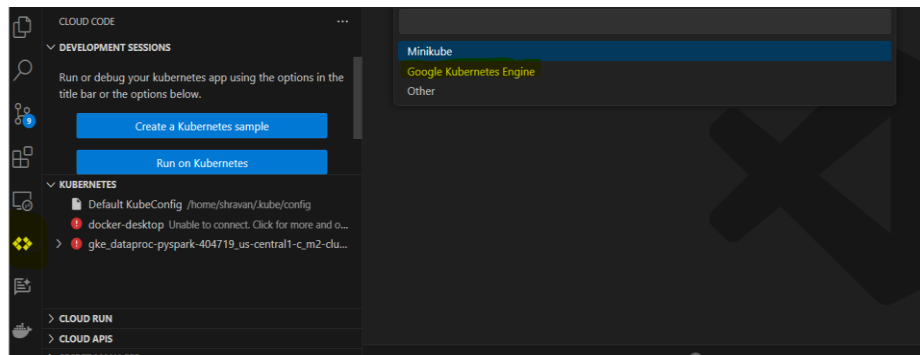


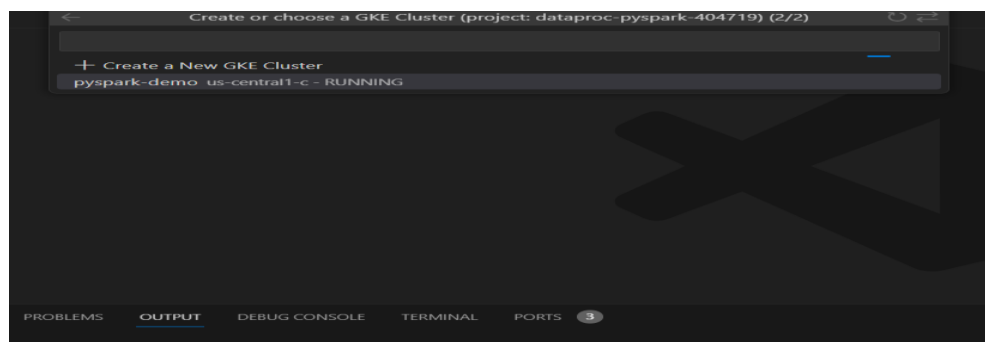
i) Create Kubernetes Cluster with Default Config



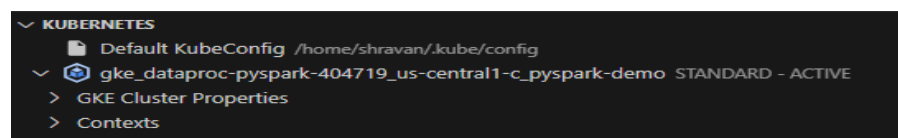
ii) Use Cloud Code plugin for VS Code and Sign in to Cloud Code with your GCP cloud account. Add your Kubernetes as shown



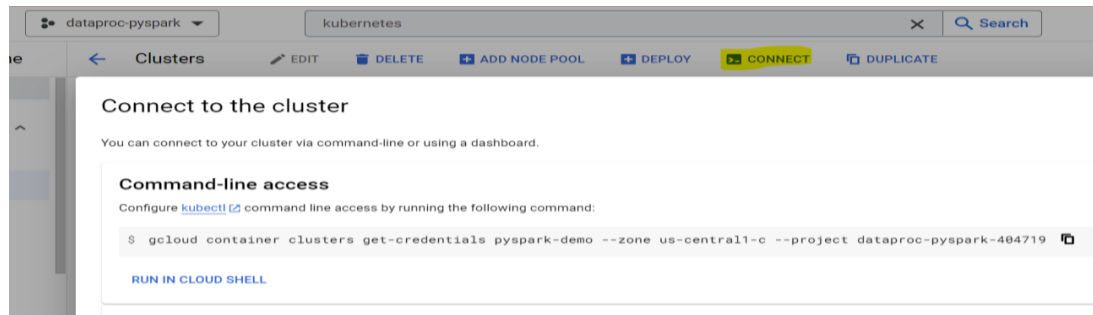
iii) You should see the cluster you created as shown below in the prompt



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.



```
honadeshhravangcp@cloudshell:~ (dataproc-pyspark-404719)$ gcloud container clusters get-credentials pyspark-demo --zone us-central1-c --project dataproc-pyspark-404719
Fetching cluster endpoint and auth data.
kubeconfig entry generated for pyspark-demo.
honadeshhravangcp@cloudshell:~ (dataproc-pyspark-404719)$ ls
k8spark  README-cloudshell.txt  setup.sh  spark-operator-1.1.26.fixed.tar
```

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

```
honadeshhravangcp@cloudshell:~ (dataproc-pyspark-404719)$ ./setup.sh
Updating helm
Error: no repositories found. You must add one before updating
Changing context to docker-desktop
error: no context exists with the name: "docker-desktop"
Installing spark operator (ignore name in use error)
NAME: my-spark-operator
LAST DEPLOYED: Sat Jan 6 21:43:42 2024
NAMESPACE: spark-operator
STATUS: deployed
REVISION: 1
TEST SUITE: None
Creating spark service account (ignore already exists error)
serviceaccount/spark created
clusterrolebinding.rbac.authorization.k8s.io/spark-role created
```

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

```
./docker_build.sh
docker tag <image-name> <repo-name>/<project-name>/<folder-name>/<remote-image-name>
Usage - docker tag myk8spark us-east4-docker.pkg.dev/dataproc-pyspark-404719/spark-docker/myk8spark-artifact
gcloud auth configure-docker us-east4-docker.pkg.dev(repo-name)
docker push <image-name> <repo-name>/<project-name>/<folder-name>/<remote-image-name>
Usage- docker push us-east4-docker.pkg.dev/dataproc-pyspark-404719/spark-docker/myk8spark-artifact
```



Artifact Registry



Repositories



Settings



Digests for myk8spark-artifact

 DELETE

[SETUP INSTRUCTIONS](#)

 us-east4-docker.pkg.dev

 dataproc-pyspark-404719

 >

 spark-docker

 >

 myk8spark-artifact



 Filter

Enter property name or value

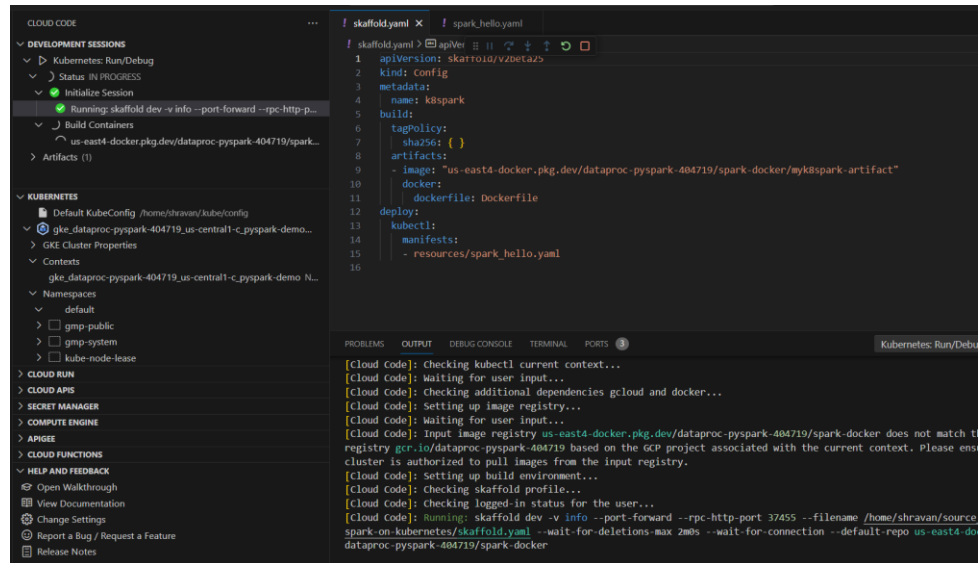
<input type="checkbox"/>	Name	Description	Tags	Created	Updated	
<input type="checkbox"/>	 b802d2fdee70		<div>latest</div>	Just now	Just now	

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

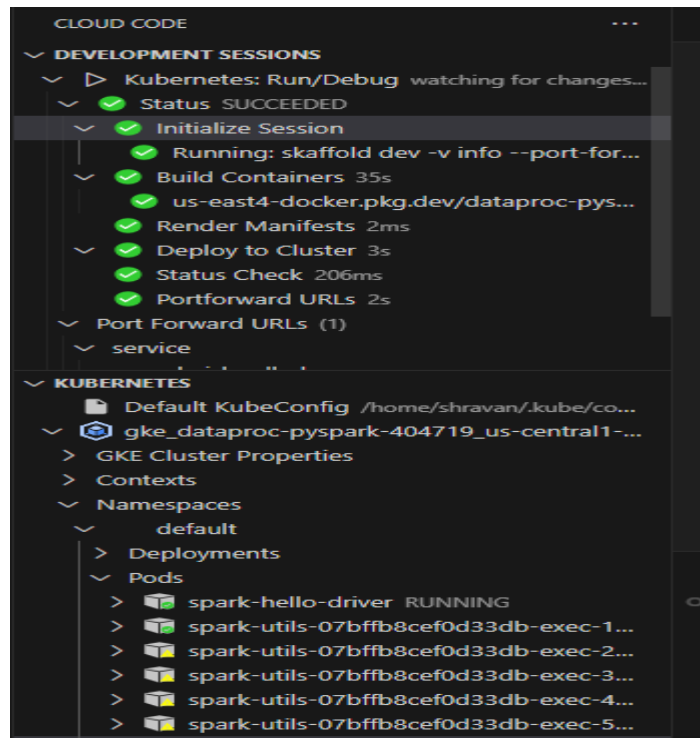
```
! scaffold.yaml X ! spark_hello.yaml Dockerfile
Ubuntu > home > shravan > source_code > spark-on-kubernetes > ! scaffold.yaml
1  apiVersion: scaffold/v2beta25
2  kind: Config
3  metadata:
4    name: k8spark
5  build:
6    tagPolicy:
7      sha256: { }
8    artifacts:
9      - image: "us-east4-docker.pkg.dev/dataproc-pyspark-404719/spark-docker/myk8spark-artifact"
10      docker:
11        dockerfile: Dockerfile
12  deploy:
13    kubectl:
14      manifests:
15        - resources/spark_hello.yaml
16
```

```
! scaffold.yaml ! spark_hello.yaml X Dockerfile
Ubuntu > home > shravan > source_code > spark-on-kubernetes > resources > ! spark_hello.yaml
1  apiVersion: "sparkoperator.k8s.io/v1beta2"
2  kind: SparkApplication
3  metadata:
4    name: spark-hello
5    namespace: default
6    labels:
7      app: spark-hello
8  spec:
9    type: Python
10   pythonVersion: "3"
11   mode: cluster
12   image: "us-east4-docker.pkg.dev/dataproc-pyspark-404719/spark-docker/myk8spark-artifact"
13   # https://scaffold.dev/docs/environment/local-cluster/
14   # Scaffold's direct loading of images into a local cluster does mean that resources specifying an
15   # imagePullPolicy: Always may fail as the images are not be pushed to the remote registry.
16   # On Docker for Desktop, don't specify imagePullPolicy
17   imagePullPolicy: Always
18   mainApplicationFile: local:///opt/spark/work-dir/runpi.py
19   sparkConf:
20     "spark.ui.port": "4040"
21   sparkVersion: "3.2"
22   restartPolicy:
23     type: Never
24   driver:
25     coreLimit: "1"
26     memoryLimit: "1G"
```

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>

```
24/01/09 16:38:51 INFO Utils: Successfully started service 'SparkUI' on port 4040.
24/01/09 16:38:51 INFO SparkUI: Bound SparkUI to 0.0.0.0, and started at http://spark-hello-ceb2868c
24/01/09 16:38:52 INFO SparkKubernetesClientFactory: Auto-configuring K8S client using current conte
24/01/09 16:38:55 INFO Utils: Using initial executors = 1, max of spark.dynamicAllocation.initialExe
24/01/09 16:38:55 INFO ExecutorPodsAllocator: Going to request 1 executors from Kubernetes for Resou
24/01/09 16:38:55 INFO BasicExecutorFeatureStep: Decommissioning not enabled, skipping shutdown scri
24/01/09 16:38:56 INFO Utils: Successfully started service 'org.apache.spark.network.netty.NettyBloc
24/01/09 16:38:56 INFO NettyBlockTransferService: Server created on spark-hello-ceb2868cef1716ea-dri
24/01/09 16:38:56 INFO BlockManager: Using org.apache.spark.storage.RandomBlockReplicationPolicy for
24/01/09 16:38:56 INFO BlockManagerMaster: Registering BlockManager BlockManagerId(driver, spark-hel
24/01/09 16:38:56 INFO BlockManagerMasterEndpoint: Registering block manager spark-hello-ceb2868cef1
c.default.svc, 7079, None)
24/01/09 16:38:56 INFO BlockManagerMaster: Registered BlockManager BlockManagerId(driver, spark-hell
24/01/09 16:38:56 INFO BlockManager: Initialized BlockManager: BlockManagerId(driver, spark-hello-ce
24/01/09 16:38:56 INFO Utils: Using initial executors = 1, max of spark.dynamicAllocation.initialExe
24/01/09 16:38:56 WARN ExecutorAllocationManager: Dynamic allocation without a shuffle service is ar
24/01/09 16:39:08 INFO KubernetesClusterSchedulerBackend$KubernetesDriverEndpoint: Registered execut
24/01/09 16:39:08 INFO ExecutorMonitor: New executor 1 has registered (new total is 1)
24/01/09 16:39:08 INFO KubernetesClusterSchedulerBackend: SchedulerBackend is ready for scheduling b
24/01/09 16:39:08 INFO BlockManagerMasterEndpoint: Registering block manager 10.36.1.9:43119 with 11
24/01/09 16:39:09 INFO SharedState: Setting hive.metastore.warehouse.dir ('null') to the value of sp
24/01/09 16:39:09 INFO SharedState: Warehouse path is 'file:/opt/spark/work-dir/spark-warehouse'.
INFO:root:56.189, seconds elapsed for spark approach and n=100000000
INFO:root:Pi is roughly 3.14142768
24/01/09 16:39:45 WARN ExecutorPodWatchSnapshotSource: Kubernetes client has been closed.
chaovar@LAPTOP-3B2B2TTO: /source_code/spark-on-kubernetes$
```

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