```
export CLUSTER_NAME="ml-gke"

gcloud container clusters create $CLUSTER_NAME \
  --enable-image-streaming \
  --addons=HttpLoadBalancing \
  --machine-type=e2-standard-2 \
  --shielded-secure-boot \
  --shielded-integrity-monitoring \
  --region=us-central1 \
  --num-nodes=1 \
  --enable-ip-alias \
  --release-channel=rapid \
  --node-locations=us-central1-a \
  --addons=RayOperator


kubectl apply -f - <<'EOF'
apiVersion: ray.io/v1
kind: RayJob
metadata:
  name: rayjob-sample
spec:
  # submissionMode specifies how RayJob submits the Ray job to the RayCluster.
  # The default value is "K8sJobMode", meaning RayJob will submit the Ray job via a
submitter Kubernetes Job.
  # The alternative value is "HTTPMode", indicating that KubeRay will submit the Ray job by
sending an HTTP request to the RayCluster.
  # submissionMode: "K8sJobMode"
  entrypoint: python /home/ray/samples/sample_code.py
  # shutdownAfterJobFinishes specifies whether the RayCluster should be deleted after the
RayJob finishes. Default is false.
  # shutdownAfterJobFinishes: false

  # ttlSecondsAfterFinished specifies the number of seconds after which the RayCluster will
be deleted after the RayJob finishes.
  # ttlSecondsAfterFinished: 10

  # activeDeadlineSeconds is the duration in seconds that the RayJob may be active before
  # KubeRay actively tries to terminate the RayJob; value must be positive integer.
  # activeDeadlineSeconds: 120

  # RuntimeEnvYAML represents the runtime environment configuration provided as a
multi-line YAML string.
  # See https://docs.ray.io/en/latest/ray-core/handling-dependencies.html for details.
  # (New in KubeRay version 1.0.)
  runtimeEnvYAML: |
    pip:
      - requests==2.26.0
      - pendulum==2.1.2
    env_vars:
      counter_name: "test_counter"

  # Suspend specifies whether the RayJob controller should create a RayCluster instance.
  # If a job is applied with the suspend field set to true, the RayCluster will not be created and
we will wait for the transition to false.
  # If the RayCluster is already created, it will be deleted. In the case of transition to false, a
new RayCluste rwill be created.
  # suspend: false

  # rayClusterSpec specifies the RayCluster instance to be created by the RayJob controller.
  rayClusterSpec:
    rayVersion: '2.9.3' # should match the Ray version in the image of the containers
    # Ray head pod template
    headGroupSpec:
      # The `rayStartParams` are used to configure the `ray start` command.
      # See
https://github.com/ray-project/kuberay/blob/master/docs/guidance/rayStartParams.md for the
default settings of `rayStartParams` in KubeRay.
      # See https://docs.ray.io/en/latest/cluster/cli.html#ray-start for all available options in
`rayStartParams`.
      rayStartParams:
        dashboard-host: '0.0.0.0'
      #pod template
      template:
        spec:
          containers:
            - name: ray-head
              image: rayproject/ray:2.9.3
              ports:
                - containerPort: 6379
                  name: gcs-server
                - containerPort: 8265 # Ray dashboard
                  name: dashboard
                - containerPort: 10001
                  name: client
              resources:
                limits:
                  cpu: "1"
                requests:
                  cpu: "200m"
              volumeMounts:
                - mountPath: /home/ray/samples
                  name: code-sample
          volumes:
            # You set volumes at the Pod level, then mount them into containers inside that Pod
            - name: code-sample
            configMap:
              # Provide the name of the ConfigMap you want to mount.
              name: ray-job-code-sample
              # An array of keys from the ConfigMap to create as files
              items:
                - key: sample_code.py
                  path: sample_code.py
    workerGroupSpecs:
      # the pod replicas in this group typed worker
      - replicas: 1
        minReplicas: 1
        maxReplicas: 5
        # logical group name, for this called small-group, also can be functional
        groupName: small-group
        # The `rayStartParams` are used to configure the `ray start` command.
        # See
https://github.com/ray-project/kuberay/blob/master/docs/guidance/rayStartParams.md for the
default settings of `rayStartParams` in KubeRay.
        # See https://docs.ray.io/en/latest/cluster/cli.html#ray-start for all available options in
`rayStartParams`.
        rayStartParams: {}
        #pod template
        template:
          spec:
            containers:
              - name: ray-worker # must consist of lower case alphanumeric characters or '-', and
must start and end with an alphanumeric character (e.g. 'my-name',  or '123-abc'
                image: rayproject/ray:2.9.3
                lifecycle:
                  preStop:
                    exec:
                      command: [ "/bin/sh","-c","ray stop" ]
                resources:
                  limits:
                    cpu: "1"
                  requests:
                    cpu: "200m"
  # SubmitterPodTemplate is the template for the pod that will run the `ray job submit`
command against the RayCluster.
  # If SubmitterPodTemplate is specified, the first container is assumed to be the submitter
container.
  # submitterPodTemplate:
  #   spec:
  #     restartPolicy: Never
  #     containers:
  #       - name: my-custom-rayjob-submitter-pod
  #         image: rayproject/ray:2.9.3
```

```
 #        # If Command is not specified, the correct command will be supplied at runtime
using the RayJob spec `entrypoint` field.
 #        # Specifying Command is not recommended.
 #        # command: ["sh", "-c", "ray job submit
--address=http://$RAY_DASHBOARD_ADDRESS
--submission-id=$RAY_JOB_SUBMISSION_ID -- echo hello world"]


######################Ray code sample##################################
# this sample is from
https://docs.ray.io/en/latest/cluster/job-submission.html#quick-start-example
# it is mounted into the container and executed to show the Ray job at work
---
apiVersion: v1
kind: ConfigMap
metadata:
  name: ray-job-code-sample
data:
  sample_code.py: |
    import ray
    import os
    import requests

    ray.init()

    @ray.remote
    class Counter:
        def __init__(self):
            # Used to verify runtimeEnv
            self.name = os.getenv("counter_name")
            assert self.name == "test_counter"
            self.counter = 0

        def inc(self):
            self.counter += 1

        def get_counter(self):
            return "{} got {}".format(self.name, self.counter)

    counter = Counter.remote()

    for _ in range(5):
        ray.get(counter.inc.remote())
        print(ray.get(counter.get_counter.remote()))

    # Verify that the correct runtime env was used for the job.
    assert requests.__version__ == "2.26.0"
EOF
```

```
Kubectl get svc

Kubectl port-forward svc/  8265:8265


Kubectl get raycluster

Kubectl edit raycluster


kubectl apply -f - <<'EOF'
apiVersion: ray.io/v1
kind: RayService
metadata:
  name: text-summarizer
spec:
  serviceUnhealthySecondThreshold: 900 # Config for the health check threshold for Ray Serve applications. Default value is 900.
  deploymentUnhealthySecondThreshold: 300 # Config for the health check threshold for Ray dashboard agent. Default value is 300.
  serveConfigV2: |
    applications:
      - name: text_summarizer
        import_path: text_summarizer.text_summarizer:deployment
        runtime_env:
          working_dir: "https://github.com/ray-project/serve_config_examples/archive/refs/heads/master.zip"
  rayClusterConfig:
    rayVersion: '2.9.3' # Should match the Ray version in the image of the containers
    ######################headGroupSpecs#################################
    # Ray head pod template.
    headGroupSpec:
      # The `rayStartParams` are used to configure the `ray start` command.
      # See https://github.com/ray-project/kuberay/blob/master/docs/guidance/rayStartParams.md for the default settings of `rayStartParams` in KubeRay.
      # See https://docs.ray.io/en/latest/cluster/cli.html#ray-start for all available options in `rayStartParams`.
      rayStartParams:
        dashboard-host: '0.0.0.0'
      # Pod template
      template:
        spec:
          containers:
          - name: ray-head
            image: rayproject/ray-ml:2.9.3
            ports:
```

```
            - containerPort: 6379
              name: gcs
            - containerPort: 8265
              name: dashboard
            - containerPort: 10001
              name: client
            - containerPort: 8000
              name: serve
            volumeMounts:
              - mountPath: /tmp/ray
                name: ray-logs
            resources:
              limits:
                cpu: "2"
                memory: "8G"
              requests:
                cpu: "2"
                memory: "8G"
          volumes:
            - name: ray-logs
              emptyDir: {}
    workerGroupSpecs:
    # The pod replicas in this group typed worker
    - replicas: 1
      minReplicas: 1
      maxReplicas: 10
      groupName: gpu-group
      rayStartParams: {}
      # Pod template
      template:
        spec:
          containers:
          - name: ray-worker
            image: rayproject/ray-ml:2.9.3-gpu
            resources:
              limits:
                cpu: 4
                memory: "16G"
                nvidia.com/gpu: 1
              requests:
                cpu: 3
                memory: "12G"
                nvidia.com/gpu: 1
          tolerations:
            - key: "nvidia.com/gpu"
              operator: "Exists"
              effect: "NoSchedule"
EOF
```