## Creating a Kubernetes Engine cluster

A GKE cluster is a managed set of Compute Engine virtual machines that operate as a single GKE cluster. This tutorial uses a single node.

1. Create the cluster. Replace **YOUR\_GCP\_ZONE** with the GCP zone where you want to host your cluster. For a complete list, see [Geography and regions](https://cloud.google.com/docs/geography-and-regions).

gcloud container clusters create helloworld-gke \  
   --num-nodes 1 \  
   --enable-basic-auth \  
   --issue-client-certificate \  
   --zone ***YOUR\_GCP\_ZONE***

**Note:** A region contains one or more zones. Using a region instead of a zone for the **--zone** option (for example, **us-central1** instead of **us-central1-b**) creates a node in each zone within the region. **Note:** The default scope is set to **--scopes=gke-default**. For a full list of available scopes, see [OAuth 2.0 scopes for Google APIs](https://developers.google.com/identity/protocols/googlescopes).

1. Verify that you have access to the cluster. The following command lists the nodes in your container cluster which are up and running and indicates that you have access to it.

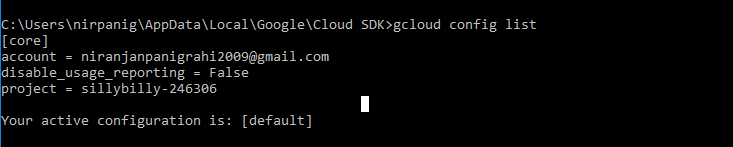
kubectl get nodes

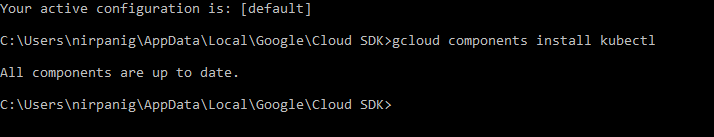
**Selecting Current Project in Gcloud SDK**

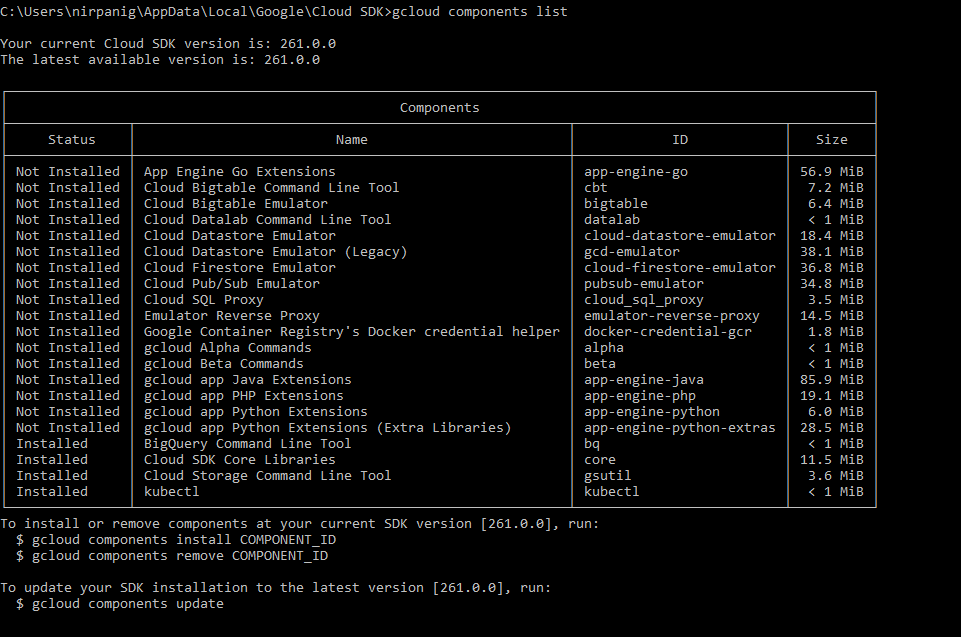
gcloud config set project my-project

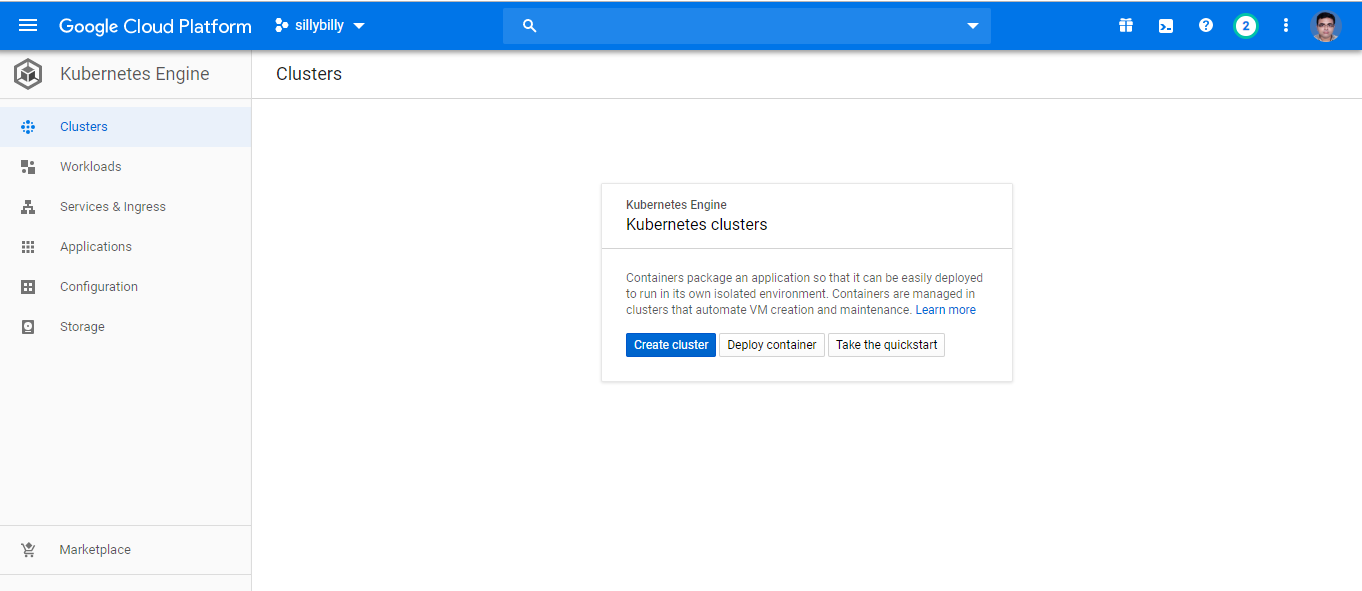
gcloud projects list

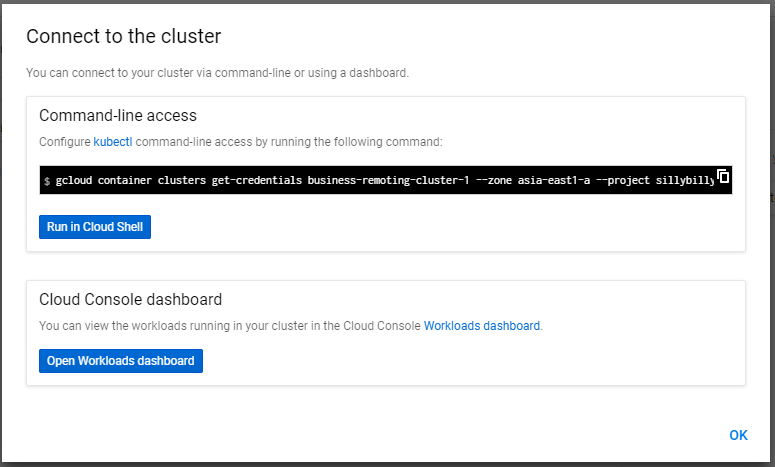
minikube









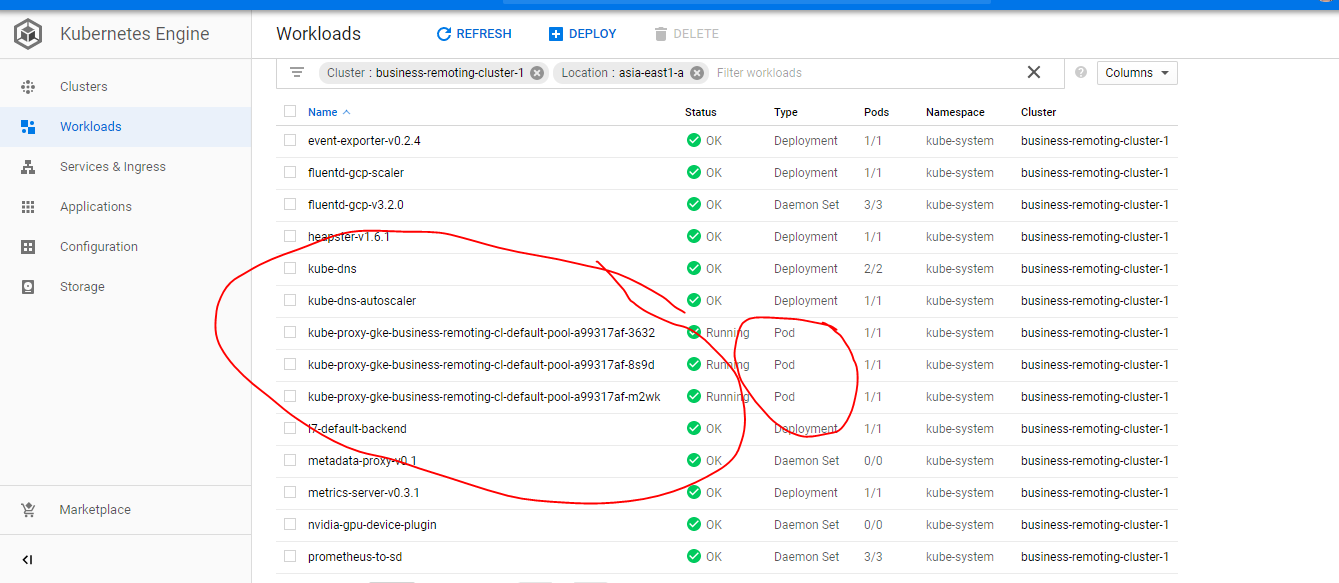


gcloud container clusters get-credentials business-remoting-cluster-1 --zone asia-east1-a --project sillybilly-246306

C:\Users\nirpanig\AppData\Local\Google\Cloud SDK>gcloud container clusters get-credentials business-remoting-cluster-1 --zone asia-east1-a --project sillybilly-246306

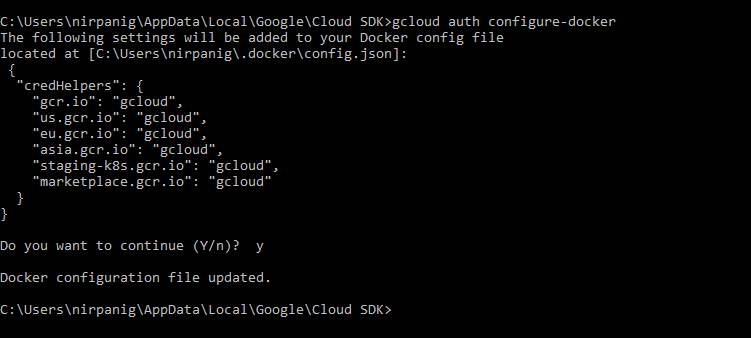
Fetching cluster endpoint and auth data.

kubeconfig entry generated for business-remoting-cluster-1.



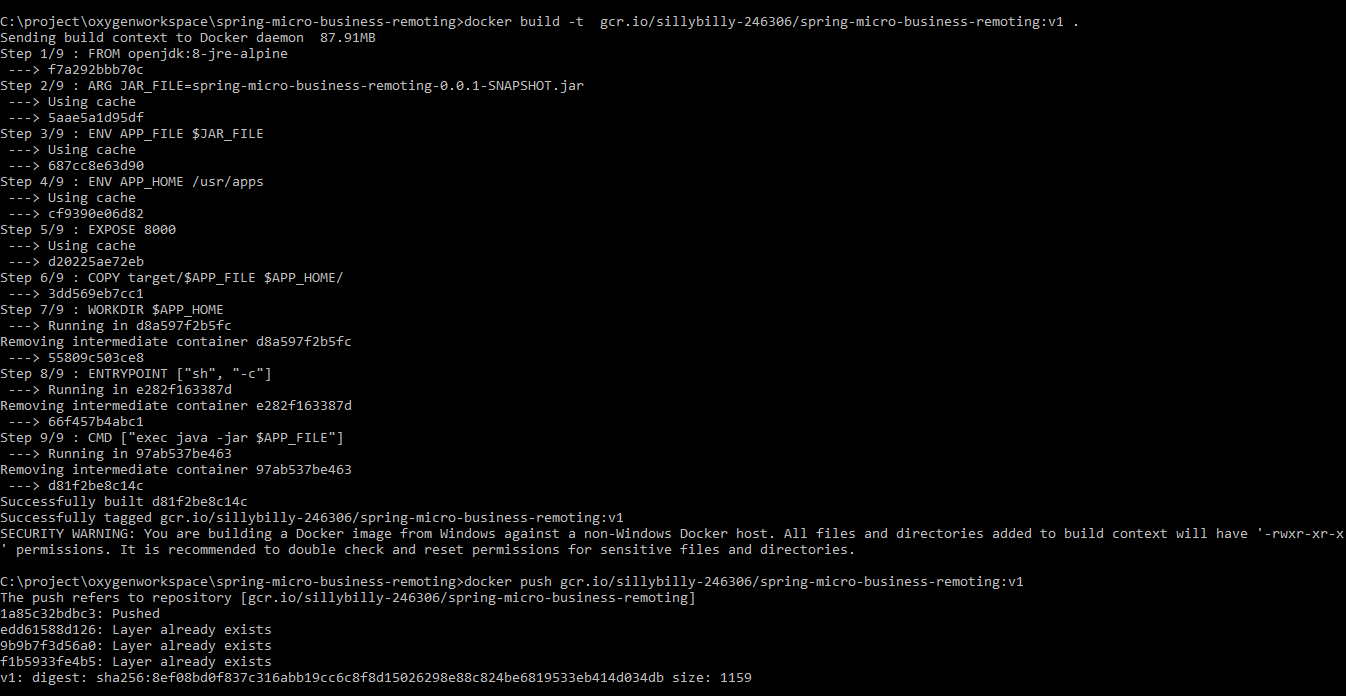
Can see all the running pods in workload tab

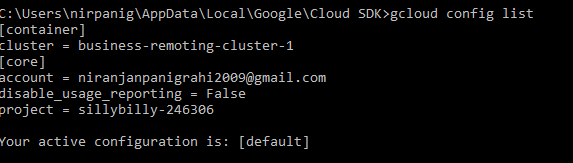
<https://cloud.google.com/container-registry/docs/pushing-and-pulling>



gcloud components install docker-credential-gcr

C:\project\oxygenworkspace\spring-micro-business-remoting>gcloud config set container/cluster business-remoting-cluster-1





**Steps to deploy images to kubernates cluster**

C:\project\oxygenworkspace\spring-micro-business-remoting>kubectl run businessremotingdeployment --image=gcr.io/sillybilly-246306/spring-micro-business-remoting:v1 --port=8000

deployment.apps "businessremotingdeployment" created

C:\project\oxygenworkspace\spring-micro-business-remoting>kubectl expose deployment businessremotingdeployment --type="LoadBalancer"

service "businessremotingdeployment" exposed

C:\project\oxygenworkspace\spring-micro-business-remoting>kubectl get services

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

businessremotingdeployment LoadBalancer 10.107.15.119 <pending> 8000:30278/TCP 54s

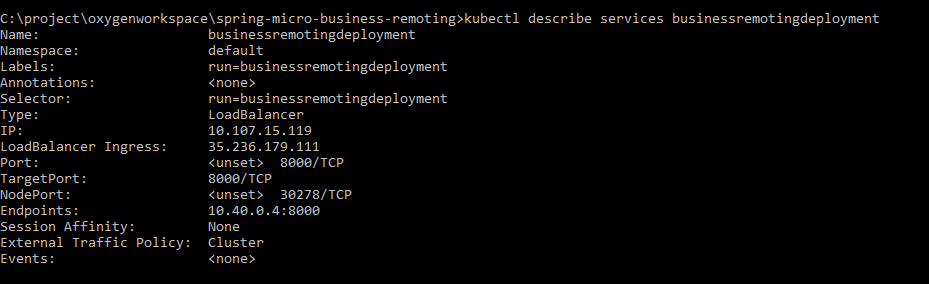
kubernetes ClusterIP 10.107.0.1 <none> 443/TCP 1d

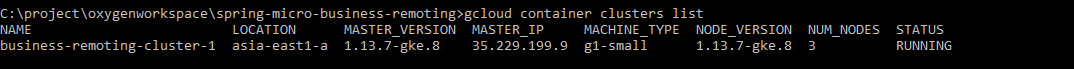
kubectl run log-exercise --image nginx

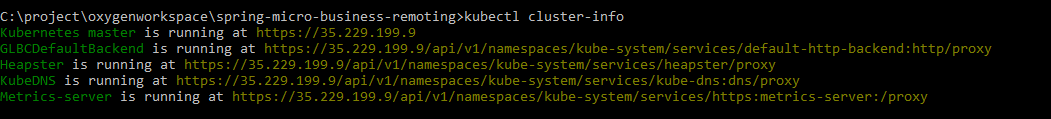
gcloud container clusters get-credentials business-remoting-cluster-1 --zone asia-east1-a --project sillybilly-246306 && kubectl get service businessremotingdeployment -o yaml

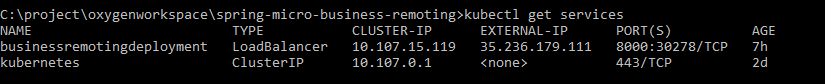
gcloud container clusters list

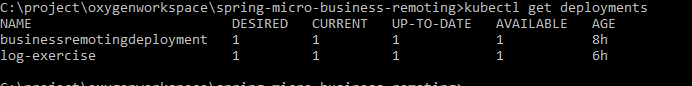
kubectl describe services businessremotingdeployment

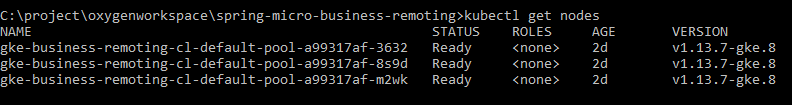


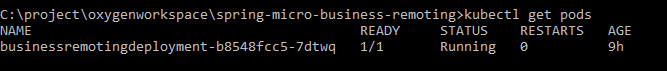












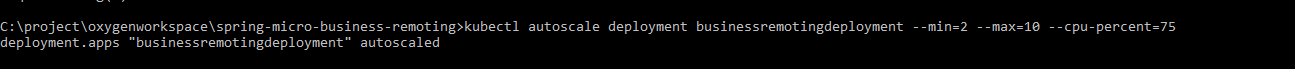
For example, the following command creates an autoscaling multi-zone cluster with six nodes across three zones, with a minimum of one node per zone and a maximum of four nodes per zone:

gcloud container clusters create example-cluster \

--zone us-central1-a \

--node-locations us-central1-a,us-central1-b,us-central1-f \

--num-nodes 2 --enable-autoscaling --min-nodes 1 --max-nodes 4

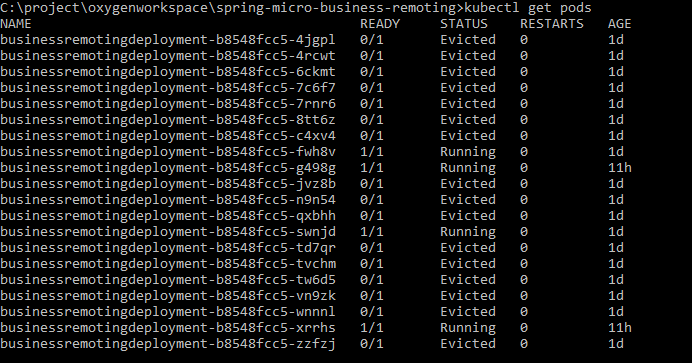


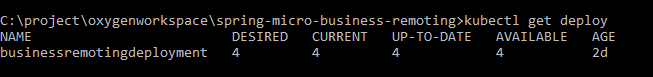
C:\project\oxygenworkspace\spring-micro-business-remoting>kubectl autoscale deployment businessremotingdeployment --min=2 --max=10 --cpu-percent=75

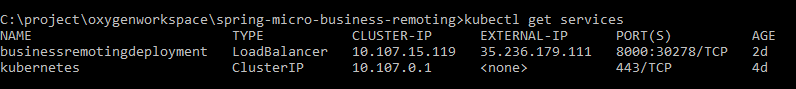
deployment.apps "businessremotingdeployment" autoscaled

C:\project\oxygenworkspace\spring-micro-business-remoting>kubectl edit hpa businessremotingdeployment

horizontalpodautoscaler.autoscaling "businessremotingdeployment" edited



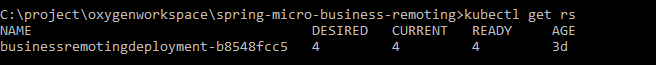




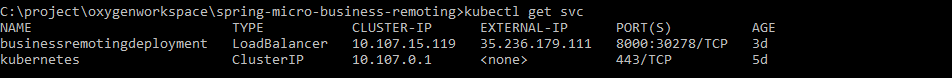
Typically we will one per pod. If we want to scale up kubernates will all more pods with same container. To scale down pods are deleted.

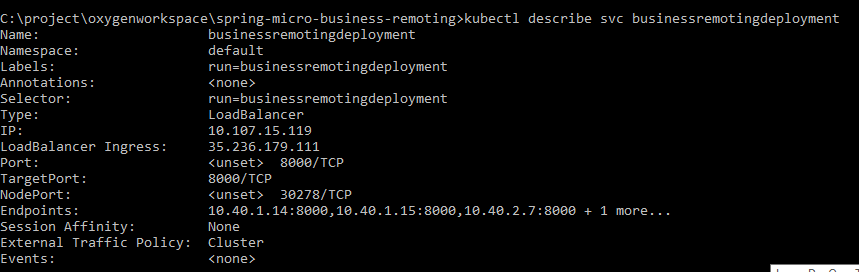
It is possible to have multiple containers per pod.

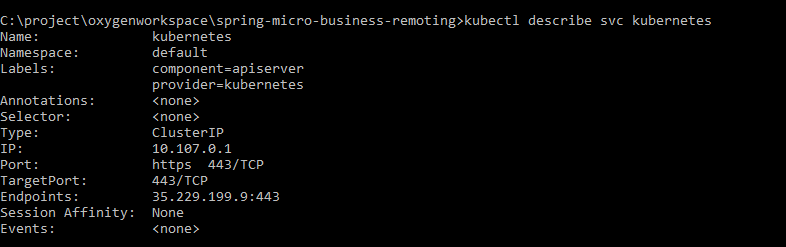
Mostly the extra containers are helper containers with are part of the pod which is used by the main container. They can refer each other as local host as they share the same host.

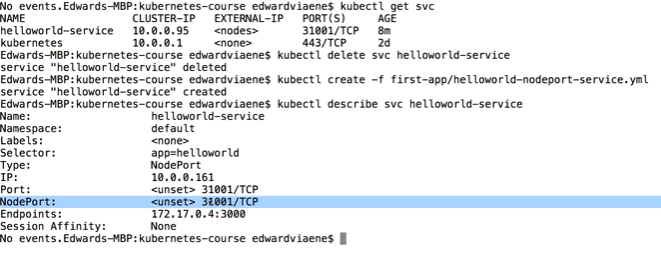


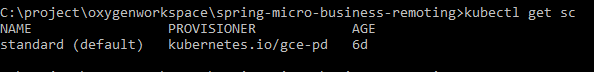
Svc shortcut for services

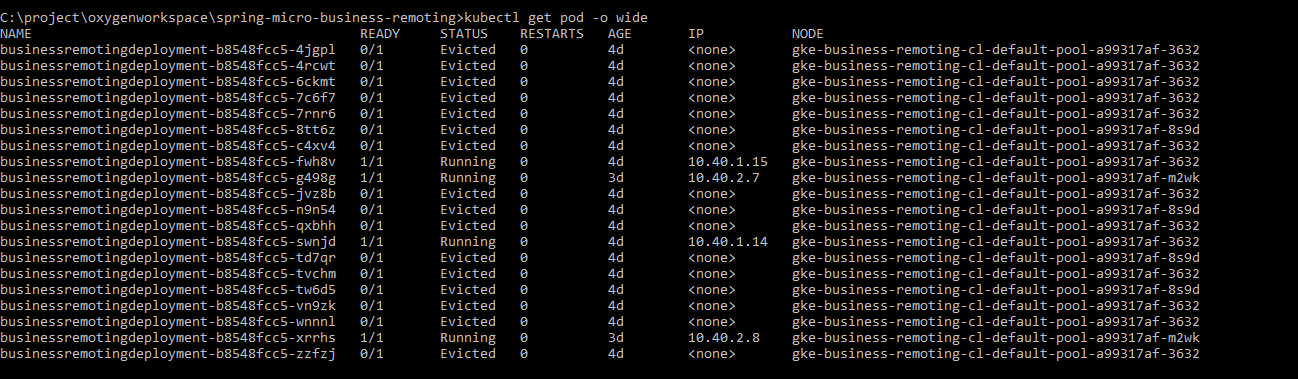


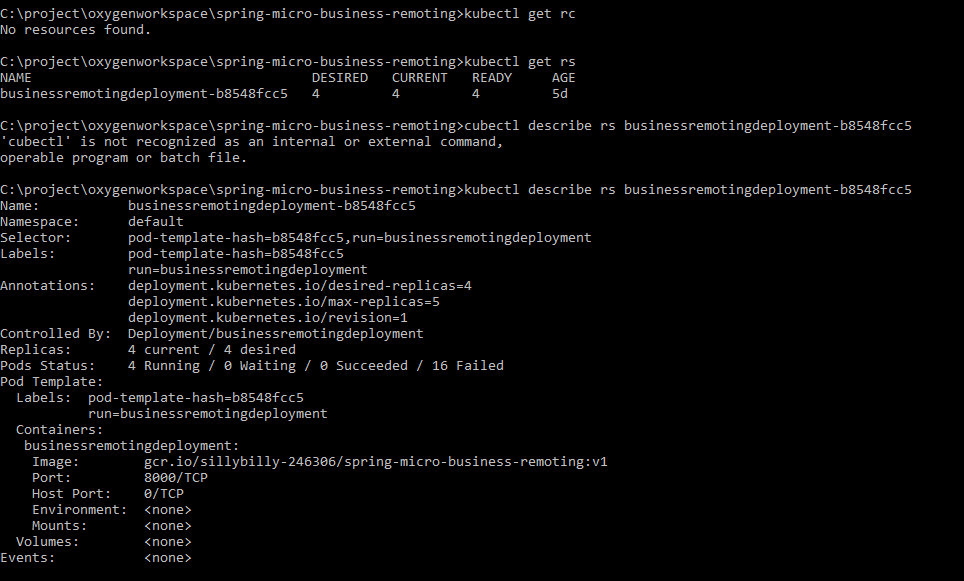












Rc 🡪 replication controller

Rs 🡪 replica set