Working with Minikube

Install minikube, docker, gcloud and kubectl on the Macbook, with Auto complete command options.

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ brew list
==> Formulae
docker          hyperkit        libev           openssl@1.1     readline  terraform     xz
gdbm            kubernetes-cli  minikube        python          sqlite          terraform-docs
```

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ echo $PATH
/Users/pavan.kumar.bijjala/google-cloud-sdk/bin:/Library/Frameworks/Python.framework/Versions/3.10/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/Library/Apple/usr/bin
```

Start Docker & start minikube.

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ minikube start
😄  minikube v1.27.1 on Darwin 12.6.2
✨  Automatically selected the docker driver. Other choices: hyperkit, ssh
📌  Using Docker Desktop driver with root privileges
👍  Starting control plane node minikube in cluster minikube
🚜  Pulling base image ...
    > gcr.io/k8s-minikube/kicbase:  387.11 MiB / 387.11 MiB  100.00% 17.64 MiB
    > gcr.io/k8s-minikube/kicbase:  0 B [_________________________] ?% ? p/s 12s
🔥  Creating docker container (CPUs=2, Memory=4000MB) ...
🐳  Preparing Kubernetes v1.25.2 on Docker 20.10.18 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

~/.kube/config

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl config view
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: DATA+OMITTED
    server: https://kubernetes.docker.internal:6443
  name: docker-desktop
- cluster:
    certificate-authority: /Users/pavan.kumar.bijjala/.minikube/ca.crt
    extensions:
    - extension:
        last-update: Wed, 11 Jan 2023 16:30:53 PST
        provider: minikube.sigs.k8s.io
        version: v1.27.1
      name: cluster_info
    server: https://127.0.0.1:56498
  name: minikube
contexts:
- context:
    cluster: docker-desktop
    user: docker-desktop
  name: docker-desktop
- context:
    cluster: minikube
    extensions:
    - extension:
        last-update: Wed, 11 Jan 2023 16:30:53 PST
        provider: minikube.sigs.k8s.io
        version: v1.27.1
      name: context_info
    namespace: default
    user: minikube
  name: minikube
current-context: minikube
kind: Config
preferences: {}
users:
- name: docker-desktop
  user:
    client-certificate-data: REDACTED
    client-key-data: REDACTED
- name: minikube
  user:
    client-certificate: /Users/pavan.kumar.bijjala/.minikube/profiles/minikube/client.crt
    client-key: /Users/pavan.kumar.bijjala/.minikube/profiles/minikube/client.key
```

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl config get-contexts
CURRENT   NAME             CLUSTER          AUTHINFO         NAMESPACE
          docker-desktop   docker-desktop   docker-desktop
*         minikube         minikube         minikube         default
```

Incase required,

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ minikube delete -all
```

(minikube start --driver=hyperkit **--alsologtostderr -v=5**)

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl cluster-info
Kubernetes control plane is running at https://127.0.0.1:56498
CoreDNS is running at https://127.0.0.1:56498/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
```

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl get namespaces
NAME              STATUS   AGE
default           Active   2m27s
kube-node-lease   Active   2m28s
kube-public       Active   2m28s
kube-system       Active   2m28s
```

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl get po -A
NAMESPACE     NAME                               READY   STATUS    RESTARTS      AGE
```

```
kube-system   coredns-565d847f94-wkljc         1/1     Running   0             3m38s
kube-system   etcd-minikube                    1/1     Running   0             3m49s
kube-system   kube-apiserver-minikube          1/1     Running   0             3m49s
kube-system   kube-controller-manager-minikube   1/1     Running   0             3m51s
kube-system   kube-proxy-wbn6g                 1/1     Running   0             3m38s
kube-system   kube-scheduler-minikube          1/1     Running   0             3m50s
kube-system   storage-provisioner              1/1     Running   1 (3m37s ago)   3m47s
```

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl get deployments --namespace=default
No resources found in default namespace.
```

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl get deployments --namespace=kube-system
NAME      READY   UP-TO-DATE   AVAILABLE   AGE
coredns   1/1     1            1           151m
```

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl create deployment hello-minikube
--image=kicbase/echo-server:1.0
deployment.apps/hello-minikube created
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl expose deployment hello-minikube
--type=NodePort --port=8080
service/hello-minikube exposed
```

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl get deployments --namespace=default
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
hello-minikube   1/1     1            1           23s
```

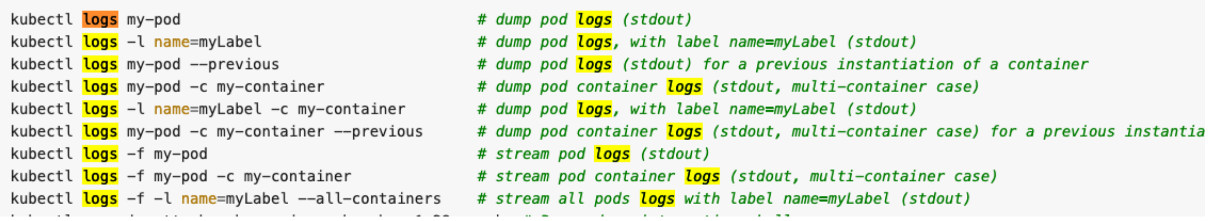
```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl get services --namespace=default
NAME             TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
hello-minikube   NodePort    10.101.193.158   <none>        8080:32405/TCP   143m
kubernetes       ClusterIP   10.96.0.1        <none>        443/TCP          150m
```

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl describe
pod/hello-minikube-7ddcbc9b8b-7cdmj
Name:             hello-minikube-7ddcbc9b8b-7cdmj
Namespace:        default
Priority:         0
Service Account:  default
Node:             minikube/192.168.49.2
Start Time:       Wed, 11 Jan 2023 16:37:10 -0800
Labels:           app=hello-minikube
                  pod-template-hash=7ddcbc9b8b
Annotations:      <none>
Status:           Running
IP:               172.17.0.3
IPs:
  IP:           172.17.0.3
Controlled By:  ReplicaSet/hello-minikube-7ddcbc9b8b
Containers:
  echo-server:
    Container ID:   docker://2256015e34934340c454865a95c78acfe4b4dec8ecd3287d8034a4436403ee4b
    Image:          kicbase/echo-server:1.0
    Image ID:
docker-pullable://kicbase/echo-server@sha256:127ac38a2bb9537b7f252addff209ea6801edcac8a92c8b1104dac
d66a583ed6
    Port:          <none>
    Host Port:     <none>
    State:         Running
      Started:     Wed, 11 Jan 2023 16:37:14 -0800
    Ready:         True
    Restart Count:  0
    Environment:   <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-jg57l (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-jg57l:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   BestEffort
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason     Age   From               Message
  ----    ------     ----  ----               -------
  Normal  Scheduled  157m  default-scheduler  Successfully assigned default/hello-minikube-7ddcbc9b8b-7cdmj
to minikube
  Normal  Pulling    157m  kubelet            Pulling image "kicbase/echo-server:1.0"
  Normal  Pulled     157m  kubelet            Successfully pulled image "kicbase/echo-server:1.0" in 3.156610344s
  Normal  Created    157m  kubelet            Created container echo-server
  Normal  Started    157m  kubelet            Started container echo-server
```

kubectl Cheat Sheet | Kubernetes

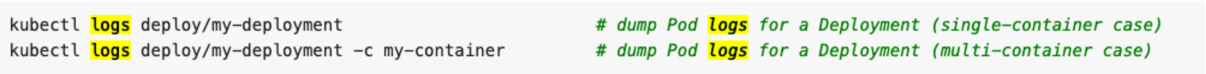
Interacting with system

logs



```
kubectl logs my-pod                                 # dump pod logs (stdout)
kubectl logs -l name=myLabel                        # dump pod logs, with label name=myLabel (stdout)
kubectl logs my-pod --previous                      # dump pod logs (stdout) for a previous instantiation of a container
kubectl logs my-pod -c my-container                 # dump pod container logs (stdout, multi-container case)
kubectl logs -l name=myLabel -c my-container        # dump pod logs, with label name=myLabel (stdout)
kubectl logs my-pod -c my-container --previous      # dump pod container logs (stdout, multi-container case) for a previous instantia
kubectl logs -f my-pod                              # stream pod logs (stdout)
kubectl logs -f my-pod -c my-container              # stream pod container logs (stdout, multi-container case)
kubectl logs -f -l name=myLabel --all-containers    # stream all pods logs with label name=myLabel (stdout)
```

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl logs hello-minikube-7ddcbc9b8b-7cdmj
Echo server listening on port 8080.
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl logs hello-minikube-7ddcbc9b8b-7cdmj -c echo-server
Echo server listening on port 8080.
```



```
kubectl logs deploy/my-deployment                         # dump Pod logs for a Deployment (single-container case)
kubectl logs deploy/my-deployment -c my-container         # dump Pod logs for a Deployment (multi-container case)
```

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl port-forward service/hello-minikube 7080:8080
Forwarding from 127.0.0.1:7080 -> 8080
Forwarding from [::1]:7080 -> 8080
..
```

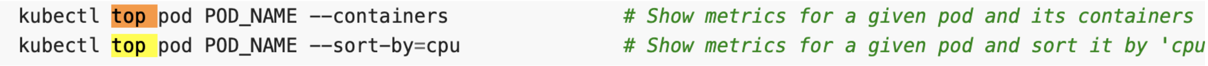
http://http://localhost:7080/

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ netstat | grep 7080
tcp6       0      0  localhost.7080         localhost.58185        ESTABLISHED
tcp6       0      0  localhost.7080         localhost.58184        ESTABLISHED
tcp6       0      0  localhost.58185         localhost.7080         ESTABLISHED
tcp6       0      0  localhost.58184         localhost.7080         ESTABLISHED
```

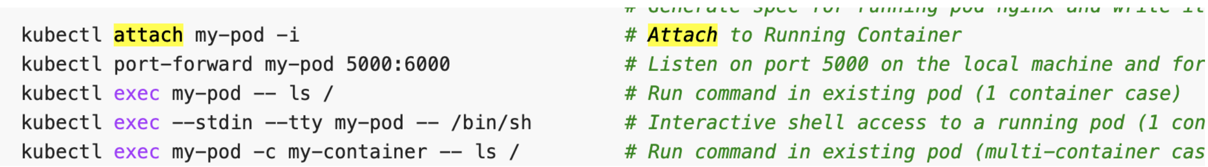
Enable metrics-server

```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl top pod hello-minicube-7ddcbc9b8b-7cdmj
error: Metrics API not available
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ minikube addons enable metrics-server
💡  metrics-server is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.
You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS
    ▪ Using image k8s.gcr.io/metrics-server/metrics-server:v0.6.1
🌟  The 'metrics-server' addon is enabled
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl rollout status deployment metrics-server -n kube-system
deployment "metrics-server" successfully rolled out

AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl top pod hello-minikube-7ddcbc9b8b-7cdmj
NAME                              CPU(cores)   MEMORY(bytes)
hello-minikube-7ddcbc9b8b-7cdmj   1m           5Mi
```



```
kubectl top pod POD_NAME --containers               # Show metrics for a given pod and its containers
kubectl top pod POD_NAME --sort-by=cpu              # Show metrics for a given pod and sort it by 'cpu
```

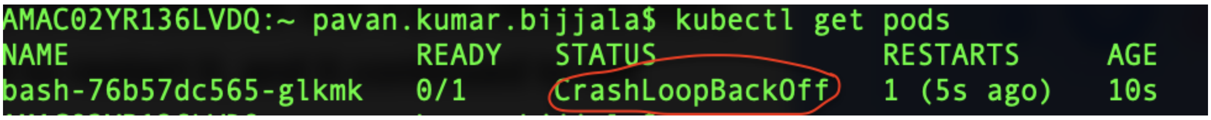


```
                                                     # Generate spec for running pod nginx and write it
kubectl attach my-pod -i                             # Attach to Running Container
kubectl port-forward my-pod 5000:6000                # Listen on port 5000 on the local machine and for
kubectl exec my-pod -- ls /                          # Run command in existing pod (1 container case)
kubectl exec --stdin --tty my-pod -- /bin/sh         # Interactive shell access to a running pod (1 con
kubectl exec my-pod -c my-container -- ls /          # Run command in existing pod (multi-container cas
```

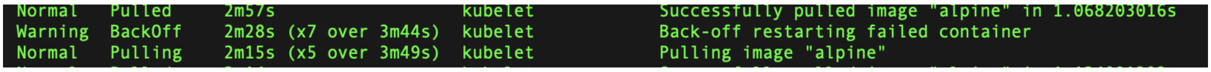
Again, the main difference is in the process you interact with in the container:

- **exec**: any one you want to create
- **attach**: the one currently running (no choice)

Unable to start alpine image on minikube, failing with below message. Couldn’t see *logs* and *describe* only gives



```
AMAC02YR136LVDQ:~ pavan.kumar.bijjala$ kubectl get pods
NAME                    READY   STATUS             RESTARTS     AGE
bash-76b57dc565-glkmk   0/1     CrashLoopBackOff   1 (5s ago)   10s
```



```
Normal   Pulled     2m57s                  kubelet            Successfully pulled image "alpine" in 1.068203016s
Warning  BackOff    2m28s (x7 over 3m44s)  kubelet            Back-off restarting failed container
Normal   Pulling    2m15s (x5 over 3m49s)  kubelet            Pulling image "alpine"
```

```
kubectl create -f ./pod.json
```

display detailed information of all pods,

kubectl describe pods

Resources

Editing a resource

```
kubectl edit svc/docker-registry                      # Edit the service named docker-registry
KUBE_EDITOR="nano" kubectl edit svc/docker-registry   # Use an alternative editor
```

Updating resources

Via Patch or Set Or Rollout Undo

```
kubectl rollout history deployment/frontend                      # Check the history of deployments in
kubectl rollout undo deployment/frontend                         # Rollback to the previous deployment
kubectl rollout undo deployment/frontend --to-revision=2         # Rollback to a specific revision
kubectl rollout status -w deployment/frontend                    # Watch rolling update status of "fro
kubectl rollout restart deployment/frontend                      # Rolling restart of the "frontend" d
```

Deleting a resource

```
kubectl delete -f ./pod.json                                      # Delete a pod using the type and na
kubectl delete pod unwanted --now                                 # Delete a pod with no grace period
```

Nodes operations

```
kubectl cordon my-node                                                # Mark my-node as unschedulable
kubectl drain my-node                                                 # Drain my-node in preparation f
kubectl uncordon my-node                                              # Mark my-node as schedulable
```

Exposing a new resource from existing,

```
# Create a service for a replicated nginx, which serves on port 80 and connects to the containers on
kubectl expose rc nginx --port=80 --target-port=8000
```