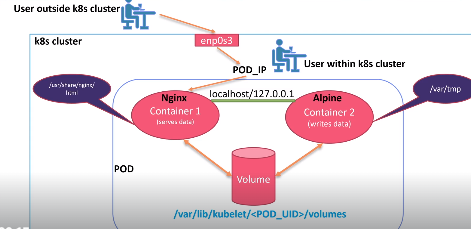
Access Pod application outside cluster

Scenario:

How to access the application outside of the cluster for the give architecture.



Pre-requisites:

1. Setup kubernetes
2. Create a Directory, create a Pod directory inside of it. Create a file with name multi-container.yaml
3. Create a yaml file which contains shared volume. One container is write-data container, another one is serves data container.
4. In the host Port entries in the code of serve-container section. By writing these entries when we browse the IP of the host machine and with the port no which we assigned to host machine we will get the application which is inside the POD.

2. Create a Directory, create a Pod directory inside of it.

Create a file with name multi-container.yaml. place the below code

apiVersion: v1

kind: Pod

metadata:

  name: nginx-demo

  labels:

    app: nginx

    type: webserver

spec:

  containers:

    - name: container-write

      image: alpine

      command: ["/bin/sh"]

      args: ["-c", "while true;do date >> /var/tmp/index.html; sleep 10;done"]

      volumeMounts:

        - name: shared-volume

          mountPath: /var/tmp

    - name: container-serve

      image: nginx

      ports:

        - containerPort: 80

          name: http

          hostPort: 8090

          protocol: TCP

      volumeMounts:

        - name: shared-volume

          mountPath: /usr/share/nginx/html

  volumes:

    - name: shared-volume

      emptyDir: {}

Create the Pod with below command.

# kubectl create -f POD/multi-container.yaml

Now check the status of the pod which is created now.

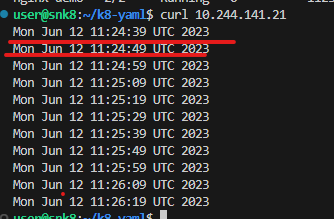
# kubectl get po –o wide



To know the response from the container which is inside the Pod use the below command with the pod IP because container IP is not directly accessible from outside of the pod. Containers within the Pod share same network namespace and IP addresses. They communicate with each other via localhost. Actually container IP’s are for internal communication within the POD.

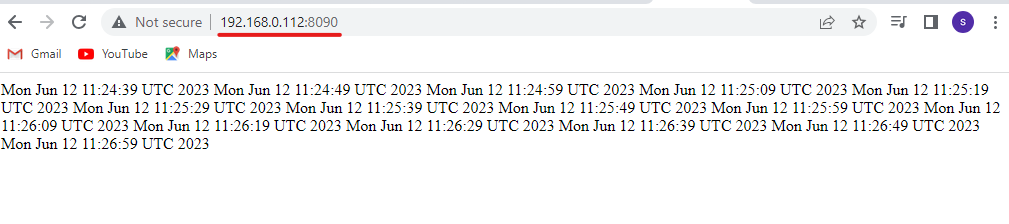
# curl 10.244.141.21

We get this output because in the code we write script to run date command with infinite loop that appends the current date to /var/tmp/index.html, and then sleep 10 seconds before execution of the command loop.



Now copy the IP of the host machine and the port which we mentioned (8090) in browser.

< <http://192.168.0.112:8090/> >



Now we are able to access the application from outside.

**The challenges we face with this host port:**

1. If we want to run any other application with the same functionality of host port we can’t use it. If I want to redeploy another application or if we want to spin up a new application with the same port no on the same node it is not possible.
2. We cannot guarantee that this application will be running on this node only. Today it is running on this node tomorrow it will run on another node. In that case how can we change the IP address in the frontend or to your clients who want to access this particular Application.

That’s why it is not feasible solution to access or to expose the application from outside of the cluster. We won’t use this process in realtime.

**References:**

1. <https://alesnosek.com/blog/2017/02/14/accessing-kubernetes-pods-from-outside-of-the-cluster/>
2. <https://kubernetes.io/docs/tasks/access-application-cluster/access-cluster-services/>
3. <https://kubernetes.io/docs/tutorials/stateless-application/expose-external-ip-address/>