

# Operation

- Run a container with kubectl
- Use Port Forwarding to Access Applications in a Cluster

# Run a container with kubectl

In Docker you can run a one-time container with the following

```
docker run --rm -it centos /bin/bash
```

To use the same in Kubernetes you need to use kubectl in any cluster

```
kubectl run tmp-shell --restart=Never --rm -i --tty --image centos -- /bin/bash
```

## Links

For mor information in general:

[kubectl for Docker Users](#)

# Use Port Forwarding to Access Applications in a Cluster

If you need to connect directly to a tcp-port in a pod you can do this with port forwarding.

So for example to access the redis pod in the namespace redis you will use this on the client:

```
kubectrl port-forward redis1-master-0 6379:6379 -n redis
```

"redis1-master" is the pod name.

You can then access redis with a client to localhost:6379

## Get pod names

To get all pod names from namespace redis use the following:

```
$ kubectrl get pods -n redis
```

| NAME              | READY | STATUS  | RESTARTS | AGE |
|-------------------|-------|---------|----------|-----|
| redis1-master-0   | 1/1   | Running | 0        | 39h |
| redis1-replicas-0 | 1/1   | Running | 0        | 39h |
| redis1-replicas-1 | 1/1   | Running | 0        | 39h |
| redis1-replicas-2 | 1/1   | Running | 0        | 39h |

## Links

<https://kubernetes.io/docs/tasks/access-application-cluster/port-forward-access-application-cluster/>