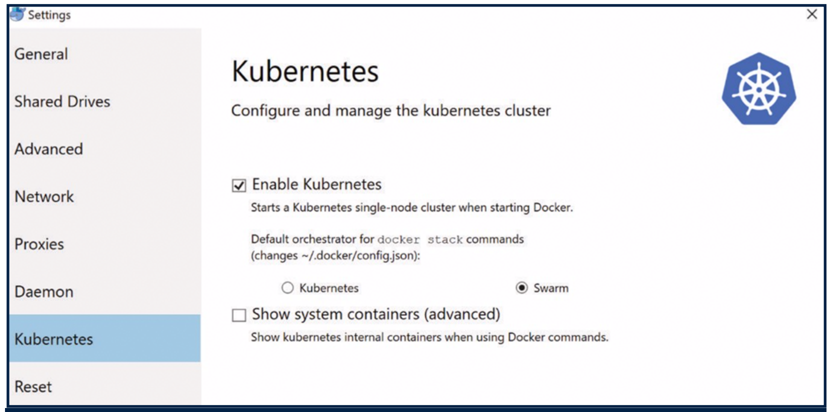
Docker Kubernetes Service Cheat Sheet

The Docker platform includes a secure and fully-conformant Kubernetes environment for developers and operators of all skill levels, providing out-of-the-box integrations for common enterprise requirements while still enabling complete flexibility for expert users.

Docker Desktop

Windows



Kubernetes

Configure and manage the kubernetes cluster

☑ Enable Kubernetes

Starts a Kubernetes single-node cluster when starting Docker.

Default orchestrator for `docker stack` commands (changes ~/.docker/config.json):

○ Kubernetes ● Swarm

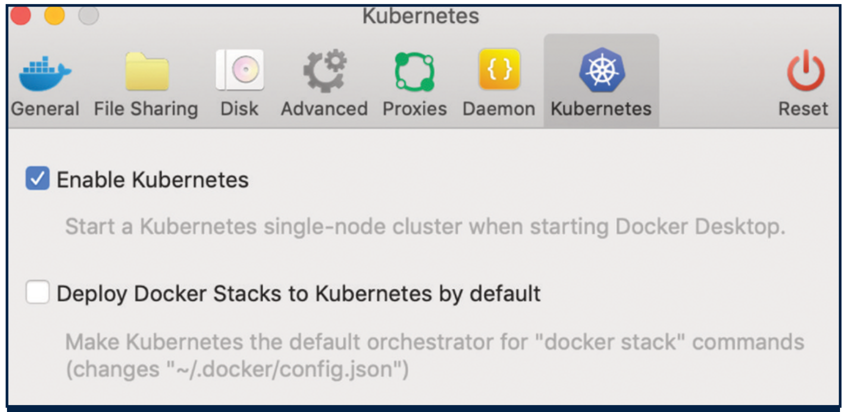
□ Show system containers (advanced)

Show kubernetes internal containers when using Docker commands.

Change default orchestrator for stack deploy commands:

```
DOCKER_STACK_ORCHESTRATOR=[kubernetes | swarm]
```

Mac



☑ Enable Kubernetes

Start a Kubernetes single-node cluster when starting Docker Desktop.

□ Deploy Docker Stacks to Kubernetes by default

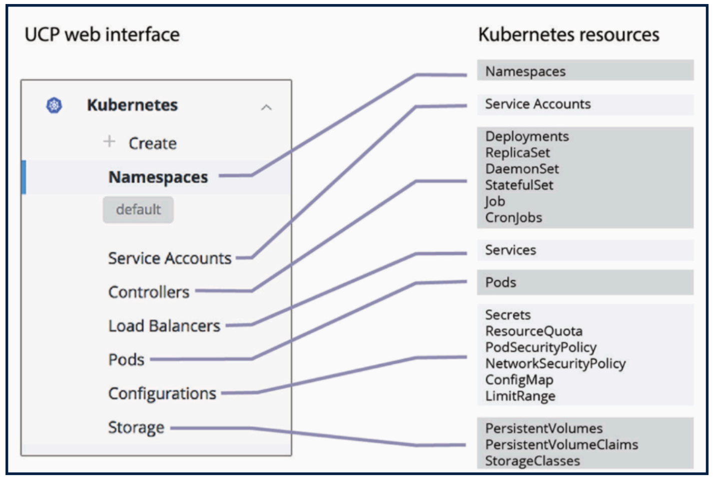
Make Kubernetes the default orchestrator for "docker stack" commands (changes "~/.docker/config.json")

Change default orchestrator for stack deploy commands:

```
DOCKER_STACK_ORCHESTRATOR=[kubernetes | swarm]
```

Docker Enterprise Universal Control Plane

Both Swarm and Kubernetes are installed and available by default in Docker Enterprise 2.0 and later.
Docker Desktop includes the Docker and Kubernetes CLIs which can be used for remote cluster access.



TIP: Docker Desktop Enterprise Version Packs ensure desktop APIs are the same version as the UCP cluster.

Authenticate to a UCP cluster from the CLI using Client Bundles:
https://docs.docker.com/ee/ucp/user-access/cli/

Deploy a workload to the Docker Enterprise Kubernetes service:

```
kubectl apply -f deployment.yaml
kubectl get deployments
```

Docker Kubernetes Service includes ***Compose on Kubernetes*** by default for both UCP clusters and Desktop. You can add ***Compose on Kubernetes*** for other Kubernetes distributions: https://github.com/docker/compose-on-kubernetes

Deploy a Compose-based application to default orchestrator in the current context:

```
docker stack deploy --compose-file /path/to/ docker-compose.yml mystack
```

Deploy a Compose-based app with Kubernetes:

```
docker stack deploy
--orchestrator kubernetes
--namespace my-app
--compose-file /path/to/ docker-compose.yml mystack
```

View deployed services:
kubectl get services
View deployed services in namespace “my-app”:

```
kubectl get services -n my-app
```

www.docker.com/kubernetes