OpenShift Cheat Sheet

A comprehensive cheat sheet covering essential OpenShift commands, concepts, and configurations for developers and operators.

Core Concepts & Commands

Basic Concepts

Project (Namespace): A logical grouping of resources, like a Kubernetes namespace.
Pod: The smallest deployable unit, containing one or more containers.
Service: Exposes an application running on a set of Pods as a network service.
Route: Exposes a service to the outside world.
DeploymentConfig: Defines the desired state of your application deployments.
Image Stream: Manages container images and their tags.

Common `oc` Commands

`oc login <openshift_url>`	Log in to the OpenShift cluster.
`oc new-project <project_name>`	Create a new project (namespace).
`oc project <project_name>`	Switch to a specific project.
`oc get pods`	List all pods in the current project.
`oc create -f <file.yaml>`	Create resources from a YAML file.
`oc apply -f <file.yaml>`	Apply changes to resources defined in a YAML file.

Resource Management

`oc describe <resource_type> <resource_name>`	Get detailed information about a resource.
`oc delete <resource_type> <resource_name>`	Delete a resource.
`oc edit <resource_type> <resource_name>`	Edit a resource directly.
`oc logs <pod_name>`	View the logs of a pod.
`oc exec -it <pod_name> -- <command>`	Execute a command inside a pod. Example: `oc exec -it my-pod -- bash`

Deployments and Services

DeploymentConfig Basics

DeploymentConfigs manage application deployments. They define the desired state (number of replicas, container image, etc.) and automatically roll out updates.

Use `oc new-app` to quickly create a DeploymentConfig from a container image or Git repository.

Example creating DeploymentConfig from image:

```
oc new-app openshift/hello-openshift --name=my-app
```

Service Management

`oc expose dc/<deployment_config_name>`	Create a service to expose a DeploymentConfig.
`oc get svc`	List services in the current project.
`oc describe svc/<service_name>`	Get details about a service.

Scaling Applications

`oc scale dc/<deployment_config_name> --replicas= <number>`	Scale the number of replicas for a DeploymentConfig.
`oc autoscale dc/<deployment_config_name> --min= <min_replicas> --max= <max_replicas>`	Configure autoscaling for a DeploymentConfig.

Rolling Updates

OpenShift supports rolling updates to minimize downtime during deployments. When you update a DeploymentConfig, OpenShift automatically updates the application instances without interrupting service.

To trigger a new deployment after changing the DeploymentConfig, use:

```
oc rollout latest dc/<deployment_config_name>
```

Routes and Networking

Route Configuration

Routes expose services to external traffic. They define the hostnames and paths that external clients use to access your applications.

Use `oc expose` to quickly create a route for a service:

```
oc expose svc/<service_name> --hostname= <desired_hostname>
```

Route Management Commands

`oc get routes`	List routes in the current project.
`oc describe route/<route_name>`	Get details about a specific route.
`oc delete route/<route_name>`	Delete a route.

Securing Routes with TLS

You can secure routes using TLS certificates. OpenShift supports edge, passthrough, and re-encrypt TLS termination.

To configure TLS, you'll need to create a secret containing your TLS certificate and key, and then reference that secret in your route definition.

Example of creating secret:

```
oc create secret tls my-tls-secret --cert=path/to/cert.pem --key=path/to/key.pem
```

Builds and Image Streams

Build Concepts

Builds transform source code into runnable container images. OpenShift supports different build strategies, including Docker, Source-to-Image (S2I), and custom builds.

BuildConfigs define how builds are executed.

Image Streams

Image Streams manage container image tags and provide a level of indirection between deployments and the underlying images. This allows you to update images without modifying your deployment configurations.

When a new image is pushed to the registry, OpenShift can automatically trigger new deployments based on the updated Image Stream tags.

Common Build Commands

`oc new-build <git_repo_url> --name=<build_name>`	Create a new build configuration from a Git repository (S2I).
`oc start-build <build_name>`	Start a build.
`oc get builds`	List builds in the current project.
`oc logs build/<build_name>`	View the logs of a build.

Working with Image Streams

`oc import-image <image_name> --from=<registry_url>/<image_name> --confirm`	Import an image from a registry into an Image Stream.
`oc get imagestreams`	List Image Streams in the current project.
`oc describe imagestream/<imagestream_name>`	Get details about an Image Stream.