OpenShift CLI (oc) OpenShift cluster CLI

CONCEPTS

oc contains kubectl: get, describe, apply, logs, exec, scale and every other kubectl verb work unchanged as oc <verb>. This sheet lists only what OpenShift adds.

Git source new-app / new-build
↓ start-build
Build BuildConfig
↓ push
ImageStream tag / import-image
↓ trigger
Deployment rollout
↓ expose
Route public hostname
Source-to-route pipeline

A project is a namespace with extra metadata and access control. -n <project> works on every command.

LOGIN & SESSION

oc login *<url>*
Log in, prompting for credentials

oc login --token=*<token>* **--server=***<url>*
Log in with a token

oc login *<url>* **--web**
Log in through a browser

LOGIN & SESSION (cont.)

oc login -u *<user>* **-p** *<password>* *<url>*
Non-interactive password login

oc whoami
Show logged-in user

oc whoami --show-console
Print web console URL

oc whoami -t
Print current token

oc whoami --show-server
Print API server URL

oc logout
End the session

PROJECTS

oc projects
List projects you can access

oc project
Show current project

oc project *<name>*
Switch project

oc new-project *<name>*
Create and switch to project

oc delete project *<name>*
Delete project and all in it

oc status
Overview of the current project

oc status --suggest
Overview with issue hints

APPS & BUILDS

oc new-app *<git-url>*
Detect, build and deploy source

oc new-app --image=*<img>* **--name=***<name>*
Deploy an existing image

oc new-app *<builder>*~*<git-url>*
Build source with a builder image

oc new-app *<img>* **-e** *<KEY>*=*<value>*
Set env vars on create

oc new-app --template=*<tpl>* **-p** *<KEY>*=*<value>*
Instantiate a template

oc new-app --search *<term>*
Search images and templates

oc new-build *<git-url>*
BuildConfig only, no deploy

oc start-build *<bc>*
Start a build

oc start-build *<bc>* **--follow**
Start and stream build log

oc start-build *<bc>* **--from-dir=***<dir>* **--follow**
Build from local files

oc logs -f bc/*<bc>*
Follow latest build log

ROUTES & NETWORK

oc expose svc/*<name>*
Create a route for a service

oc expose svc/*<name>* **--hostname=***<host>*
Route with a custom host

oc get routes
List routes and hosts

oc create route edge --service=*<svc>*
TLS route, edge termination

oc port-forward pod/*<pod>* *<local>*:*<remote>*
Tunnel a local port

DEPLOY & ROLLOUT

oc rollout status deploy/*<name>*
Watch a rollout

oc rollout restart deploy/*<name>*
Restart all pods

oc rollout undo deploy/*<name>*
Roll back one revision

oc rollout history deploy/*<name>*
List revisions

oc set env deploy/*<name>* *<KEY>*=*<value>*
Set env var (triggers rollout)

oc set env deploy/*<name>* **--from=secret/***<secret>*
Env from a secret

IMAGES

oc get is
List image streams

oc import-image *<is>*:*<tag>* **--from=***<img>* **--confirm**
Import an external image

oc tag *<is>*:*<tag>* *<is>*:*<newtag>*
Point a tag at an image

oc registry login
Log in to the integrated registry

DEBUG & FILES

oc rsh *<pod>*
Shell in first container

oc rsync *<dir>*/ *<pod>*:*<path>*
Copy local dir to pod

oc debug deploy/*<name>*
Copy pod and open a shell

oc debug node/*<node>*
Root shell on a node

ADMIN (OC ADM)

oc adm policy add-role-to-user *<role>* *<user>*
Grant role in project

oc adm must-gather
Collect cluster data for support

oc get co
Cluster operator health