OpenShift

Thinknyx®

OpenShift Cheat Sheet

OpenShift

OpenShift Cheatsheet

Thinknyx®

Console Login and Authentication

```
oc login https://<API url/IP address>:6443 -u <user id> -p <password>    # User login using cli
oc login -u system:admin                                                  #  Admin login using cli
oc whoami                                                                 # Check login Information of user
oc login https://api.example.com:6443 --token=<TOKEN>                     # To login in cli get the token from web console
oc whoami -t                                                              # To login in cli get the token
oc login --kubeconfig=<PATH_TO_KUBECONFIG>                                # To Logging in cli with a kubeconfig file
oc login https://<api url>:6443 -u <UN> -p <PASS>  --namespace=<PROJECT_NAME>    # To Logging in cli for specific project
oc login https://<api url>:6443 --certificate-authority=<PATH_TO_CA_CERT>        # To Logging in cli with CA certificate
```

Cluster Information and Management

```
oc get clusterversion                                  # Version Information of Kubernetes API server and the OpenShift server
oc cluster-info                                        # Check URL of the OpenShift console & URL of the Kubernetes API server
oc cluster-info --loglevel=5                           # Display the cluster with a higher log level
kubectl cluster-info dump                              # Dump current cluster state to stdout
kubectl cluster-info dump --all-namespaces             # Dump all namespaces to stdout
kubectl cluster-info dump --output-dir=/path/to/cluster-state    # Dump current cluster state to /path/to/cluster-state
kubectl cluster-info dump --namespaces default,kube-system --output-dir=/path/to/cluster-state
#  Dump a set of namespaces to /path/to/cluster-state
```

OpenShift Cheatsheet

Application Management

oc new-app --list	# List all local templates and image streams that can be used to create an app
oc new-app . --image=registry/repo/langimage	# Create an application based on the source code in the current git repository
oc new-app --strategy=docker --binary --name <app name>	# Create an application with Docker based build strategy expecting binary input
oc new-app --search thinknyx	# Search all templates, image streams, and images that match "thinknyx"
oc new-app --search --template=thinknyx	# Search for "thinknyx", but only in stored templates
oc new-app --search --template=thinknyx --output=yaml	# Search for "thinknyx" in stored templates and print the output as YAML

Project Management

oc new-project <project name>	# Creates a new project
oc new-project <project name> --display-name=<displayname> --description=<description details>	# Creates a new project with project description and display name
oc project	# Shows current project
oc get project	# List out all the projects
oc get project <project name> -o yaml	# Retrieves information about project on YAML format
oc describe project <project name>	# Display detailed information about a project
oc delete project <project name>	# Delete a project

OpenShift

OpenShift Cheatsheet

Thinknyx®

Build and Deployment Configuration

oc get bc <name>	# Check build config
oc new-build . --image=<repo>/<image>	# Creates a new build configuration with container image
oc new-build https://github.com/openshift/ruby-hello-world	# Creates a new build configuration with remote repository

oc new-build https://github.com/openshift/ruby-hello-world -e RACK_ENV=dev

# Creates a new build configuration with remote repository with environment

oc new-build https://github.com/<youruser>/<yourgitrepo> --source-secret=<yoursecret>

# Creates a new build configuration with remote repository and secret

oc start-build <hello-world>	# Starts a new build
oc start-build --from-build=hello-world-1	# Starts build from a previous build "hello-world-1"
oc start-build hello-world --from-dir=src/	# Starts build from a directory as build input
oc start-build hello-world --follow	# Starts build & watch the logs until the build complete
oc cancel-build <hello-world>	# Cancels a build in progress
oc cancel-build <hello-world1> <hello-world2> <hello-world3> --dump-logs	# Cancel multiple builds with build logs print
oc delete buildconfigs.build.openshift.io <hello-world>	# Deletes a build configuration

oc rollout <cancel>/< history>/< latest>/< pause>/< restart>/ <resume>/ <retry> /<status>/< undo> (app)

# Manages rollouts of application

oc rollback <deployment config>	# Rollback to a specific deployment
oc rollback <deployment config> --to-version=3 --dry-run	# Rollback to version 3 but do not perform the rollback
oc tag openshift/<image>:2.0 project/<image>:thinknyx	# Tags an image in the local image registry

OpenShift

OpenShift Cheatsheet

Thinknyx®

Service Management

oc create service clusterip <service name> --tcp=[5678]:[8080]	# Creates a new cluster ip service
oc create service externalname <service name> --external-name <name>	# Create a new ExternalName service
oc create service nodeport <service nmae> --tcp=[5678]:[8080]	# Create a new NodePort service
oc expose service <service name>	# Create a route to expose a service externally
oc expose service <service name> -l name=<label name> --name=<route name>	# Create a route with label and route name
oc delete service <service name>	# Deletes a service
oc get service <service name>	# Retrieves information about services
oc get service <service name> -o yaml	# Retrieves information about services on YAML format
oc describe service <service name>	# Displays detailed information about a service
oc edit service <service name>	# Modifies a service

OpenShift

OpenShift Cheatsheet

Thinknyx®

POD Management

oc get pods	# Retrieves information about all pods in the current project
oc get pods -o wide -A	# List all pods in ps output format with node name and more details
oc get pods -A	# Retrieves information about all pods in the current project
oc describe pod <pod-name>	# Displays detailed about a specific pod with containers and volumes
oc get pods <pod-name>	# Retrieves information about a specific pod
oc logs <pod-name>	# Displays logs from a running pod
oc logs -f <pod-name>	# Streams logs from a running pod
oc exec <pod-name> -- <command>	# Executes a command in a running container in a pod
oc attach <pod-name>	# Attaches to a running container in a pod
oc rsh <pod-name>	# Runs a shell in a running container in a pod
oc delete pod <pod-name>	# Deletes a specific pod and its associated containers and volumes
oc delete pod --all	# Deletes all pods in the current project
oc get -o json pod <pod name>	# List a single pod in JSON output format
oc get -o yaml pod <pod name>	# List a single pod in yaml output format
oc port-forward <pod-name> <local-port>:<remote-port>	# Forwards traffic from a local port to a port on a running pod
oc scale --replicas=3 rs/<resource name>	# Scales a deployment to a specified number of replicas

OpenShift Cheatsheet

POD and Project Network Management

oc adm pod-network join-projects --to=<p1> <p2>

# Allow project p2 to use project p1 network using redhat/openshift-ovs-multitenant network plugin

oc adm pod-network join-projects --to=<p1> --selector='name=thinknyx'

# Allow all projects with label name=thinknyx to use project p1 network

oc adm pod-network make-projects-global <p1>

# Allow project p1 to access all pods in the cluster and vice versa

oc adm pod-network make-projects-global --selector='name=thinknyx'

# Allow all projects with label name=thinknyx to access all pods in the oc cluster and vice versa

adm pod-network isolate-projects <p1>

# Allows projects to isolate their network from other projects using redhat/openshift-ovs-multitenant network plugin

oc adm pod-network isolate-projects --selector='name=thinknyx'

# Allow all projects with label name=thinknyx to have their own isolated project network

oc port-forward pod/mypod 5000 6000

# Forwards traffic ports 5000 and 6000 locally, forwarding data to/from ports 5000 and 6000 in the pod

oc port-forward pod/<pod name> :5000

# Forwards traffic from a local port to 5000 in the pod

oc port-forward --address localhost,<xx.xx.xx.xx> pod/mypod 8888:5000

# Listen traffic on port 8888 on localhost and selected IP, forwarding to 5000 in the pod

OpenShift

OpenShift Cheatsheet

Thinknyx®

Monitoring and Logging

```
oc logs <pod name>                                          # Streams logs from a specific pod
oc logs -follow dc/<deployment config name>                 # Streaming the logs of latest deployment config
oc logs -follow bc/<build config name>                      # Streaming the logs of latest build config
oc logs <pod name>  --since-time='<time stamp>’             # Streams logs from a specific pod with time stamp
oc logs <pod name> -c <container name> / oc logs -f pod/<pod name> -c <container name>
# Streams container logs from pod
oc get events                                               # See all the OCP cluster activities
oc get events.events.k8s.io                                 # See all the OCP cluster activities with more information
```

Storage Management

```
oc create -f <storageclass>.yaml                            # Creates a new storage class based on the requirment like glusterfs,
oc create -f <name_of_endpoint_file>                        # To create the endpoints of static volume provisioning
oc get endpoints                                            # To check the endpoint IP Address and Name
oc delete storageclasses.storage.k8s.io                     # Deletes a storage class
oc get pv -o wide                                           # Retrieves information about persistent volumes
oc describe pv                                              # Displays detailed information about a persistent volume
oc get pvc -o wide                                          # Retrieves information about persistent volume claims
oc describe pvc                                             # Displays detailed information about a persistent volume claim
```

OpenShift Cheatsheet

Scheduling and Scaling

oc scale --replicas=3 rs/<replica name>	# Scale a replica set to 3
oc scale --current-replicas=2 --replicas=3 deployment/<dc name>m	# Existing deployment size is 2 and scale it to 3 nos
oc autoscale deployment <deployment name> --min=2 --max=10	# Manages pod autoscalers
oc autoscale rc <deployment name> --max=5 --cpu-percent=80	# Manages pod autoscalers with target CPU utilization at 80%

Machine Config

oc get mc	# Retrieves information about machine configurations
oc get machineconfigpool	# Check the number of MCO-managed nodes available on your cluster
oc describe machineconfigpool <name>	# Retrieves detailed information about a specific machine configuration
oc describe machineconfig <node name>	# Retrieves detailed information about a specific machine configuration
oc edit machineconfig <node name>	# Opens the specified machine configuration in an editor to modify its contents
oc delete machineconfig <node> / oc delete -f ./myconfig.yaml	# Deletes a machine configuration

OpenShift

OpenShift Cheatsheet

Thinknyx®

Configuration and Secret Management

```
oc get configmap <configmap name>                                   # Retrieves information about config maps in the current project
oc describe configmap <configmap name>                              # Retrieves detailed information about a specific config map
oc create configmap <config-name> --from-file=<file path>           # Creates a new config map from a file
oc create configmap <config-name> --from-literal=key1=config1 --from-literal=key2=config2
# Create a config map with key1=config1 and key2=config2
oc get secret                                                       # Retrieves information about secrets in the current project
oc describe secret <secret name>                                    # Retrieves detailed information about a specific secret
oc create secret generic <secret-name> --from-file=<file path>      # Creates a new secret from a file, directory, or literal value

oc create secret docker-registry <secret-name> --from-file=.dockerconfigjson=<path/to/.docker/config.json>
# Create a new secret from ~/.docker/config.json
oc create secret tls <secret-name> --cert=<path/to/tls.cert> --key=<path/to/tls.key>     # Create a new secret using key pair
oc set env pods --all --list                                        # List the environment variables defined on all pods
oc set env --from=secret/<secret name> dc/<app name>                # Import environment from a secret
oc set env rc --all ENV=prod                   # Update all containers in all replication controllers in the project to have ENV=prod
oc set volume dc --all                         #  In the current project list volumes for all dc
oc set volume dc/<app> --add --mount-path=<dir path>     # Add a new empty dir volume to deployment config will be mount to the dir
oc set image dc/<nginx> <busybox>=<busybox> <nginx>=<nginx>:<1.9.1>
#Set a dc nginx container image to 'nginx:1.9.1', and its busybox container image to 'busybox'
```

OpenShift Cheatsheet

Administration Management

oc adm must-gather	# Gathers troubleshooting information for the cluster
oc adm must-gather --dest-dir=</local/directory path>	# Gathers troubleshooting information for the cluster to a directory
oc adm top node	# Displays resource usage for nodes
oc adm top images	# Displays image resource, Registry path and utilization
oc adm top is	# Displays image size and layers
oc adm top pod	# Displays metrics for the pod
oc adm upgrade	# Review the available cluster updates
oc adm upgrade --to-latest=true	# Update to the latest version
oc adm upgrade channel "stable-4.xx"	# Update to the specific version by setting the channel
oc adm cordon <node name>	# Marks a node as unschedulable
oc adm drain <node name>	# Drains a node of its pods
oc adm drain <node name> --force	# Drains a node even if there are pods not managed by a replication controller, replica set, job, daemon set
oc adm taint nodes <node name> <key>:NoSchedule-	# Taint remove from node with key 'key' and effect 'NoSchedule'
oc adm release info	# Information about the cluster's current release

OpenShift

OpenShift Cheatsheet

Thinknyx®

Debugging Cluster & Resources

```
oc adm inspect clusteroperator/openshift-apiserver                       # Collect debugging data for the "openshift-apiserver" clusteroperator
oc adm inspect clusteroperator/openshift-apiserver clusteroperator/kube-apiserver
 # Collect debugging data for the "openshift-apiserver" and " kube-apiserver" clusteroperators
oc adm inspect clusteroperator                                           # Collect debugging data for all clusteroperators
oc adm inspect clusteroperators,clusterversions                          # Collect debugging data for all clusteroperators and clusterversions

oc adm  node-logs <node name>                                            # collect perticular node logs
oc adm node-logs --role master -u kubelet                                # Show kubelet logs from all masters
oc debug                                                                 # Start a shell session into a pod using the OpenShift tools image
oc debug deploy/thinknyx                                                 # Debug a currently running deployment by creating a new pod
oc debug node/<node-name/IP address>                                     # Debug pod on a specific node for troubleshooting

oc debug node/<node-name> --image=<debug-image>                          # Debug pod with a custom debug image on a specific node for troubleshooting
oc debug deployment/<deployment-name>                                    # Debug pod on a specific deployment for troubleshooting
oc debug pod/<pod-name>                                                  #  Debug pod for a specific pod for troubleshooting
oc debug <resource>/<resource-name>                                      #  Debug pod for a specific resource for troubleshooting
oc debug --image=<debug-image>                                           #  Debug pod with a custom debug image for troubleshooting
oc debug job/test --as-user=1000000                                      # Test running a job as a non-root user
oc debug --as-root                                                       #  Debug pod with root privileges for troubleshooting
oc debug --uid=<user-id>                                                 # Debug pod with a specific user ID for troubleshooting
```

OpenShift Cheatsheet

Policy Management for user and Identity

oc adm policy add-cluster-role-to-user <role> <user> # Add cluster role to a exsisting user

oc adm policy add-role-to-user <role> <user> # Add normal role to a existing user

oc adm policy add-scc-to-user <scc policy> <user1> <user2> # Add scc policy to a existing user

oc adm policy remove-user <user name> # Remove user from cluster

oc adm policy remove-cluster-role-from-user <role> <user name> # Remove cluster role from user

oc get sa List all service accounts

oc adm policy scc-review -z <service account> -f res.yaml

# Check whether service accounts service account can admit a pod with a template pod spec specified in res.yaml

oc adm policy scc-subject-review -u <user> -f res.yaml # Check whether user can create a pod specified in res.yaml

Thinknyx®

YOU TRUST, WE DELIVER

Reach out to us at:

www.thinknyx.com

support@thinknyx.com

+91 9810344919/9717917973