



kubernetes

Kubernetes Cheat Sheet & Kubernetes Troubleshooting

Kubernetes Errors & Remediation

Node Memory Pressure

- Node is running out of memory resources

ANS: Check and fine-tune pod memory requests and limits "**kubectl describe pod <pod_name>**"

Utilize profiling tools to spot memory leaks

OOMKilled

- Container is killed due to out-of-memory

ANS: Increase memory

Identify and Kill memory-hogging processes, use "**top**"

Optimize application and caching

ImagePullBackOff

- Unable to pull container image from registry
- ANS: Check image availability and authentication

Use correct image name and tag

Verify Registry Quotas and Rate Limits

Check Network connectivity, firewalls, proxies

CrashLoopBackOff

- Container repeatedly crashes after starting

ANS: Inspect container logs for issues "**kubectl logs <pod_name> - c <container_name>**"

Adjust resource limits if necessary

PodPending

- Pods remain in the Pending state

ANS: Check node availability "**kubectl get nodes**"

Investigate Pod scheduling issues "**kubectl describe pod <pod_name>**" / "**kubectl get events**"

NodeNotReady

- Node is not ready for scheduling

ANS: Investigate node health "**kubectl describe node <node_name> & kubectl get nodes**"

Check kubelet logs "**/var/log/kubelet.log**"

NamespaceNotFound

- Trying to work with a non-existent namespace

ANS: Create the namespace using "**kubectl create namespace <name>**" before using it in your manifests

EvictedPods

- Pods are evicted from nodes due to resource constraints

ANS: List Evicted Pods "**kubecttl get pods --field-selector=status.phase==Failed**"

Get evicted pod details "**kubecttl describe pod <pod_name>**" and adjust resources

SecretDecryptionError

- Secrets can't be decrypted for pods
- ANS: Ensure the correct secret is referenced in your pod configuration and The secret is created correctly

Inconsistent State

- Desired state of a resource does not match actual state

ANS: Inspect Resource Status "**kubecttl describe & kubecttl get**"

Reapply - **kubecttl delete** followed by **kubecttl apply** with the corrected configuration

Kubernetes Cheat Sheet

Creating Objects: Create resource

kubecttl apply -f ./<file_name>.yaml

Create from multiple files

kubecttl apply -f ./<file_name_1>.yaml -f ./<file_name_2>.yaml

Create all files in directory

kubecttl apply -f ./<directory_name>

Create from url

kubecttl apply -f https://<url>

Create pod

kubecttl run <pod_name> --image <image_name>

Create pod, then expose it as service

kubecttl run <pod_name> --image <image_name> --port <port> --expose

Create pod yaml file

kubecttl run <pod_name> --image image_name --dry-run=client -o yaml > <file_name>.yaml

Create deployment

kubecttl create deployment <deployment_name> --image <image_name>

Create deployment yaml file

kubecttl create deployment <deployment_name> --image <image_name> --dry-run=client -o yaml > <file_name>.yaml

Create service

kubecttl create service <service-type> <service_name> --tcp=<port:target_port>

Create service yaml file

kubecttl create service <service-type> <service_name> --tcp=<port:target_port> --dryrun=client -o yaml > <file_name>.yaml

Expose service from pod/deployment

kubectrl expose deployment <pod/deployment_name> -type=<service-type> --port <port> --target-port <target_port>

Create config map from key-value

kubectrl create configmap <configmap_name> --fromliteral=<key>:<value> --from-literal=<key>:<value>

Create config map from file

kubectrl create configmap <configmap_name> --fromfile=<file_name>

Create config map from env

kubectrl create configmap <configmap_name> --from-envfile=<file_name> file

Create secret from keyvalue

kubectrl create secret generic <secret_name> --fromliteral=<key>:<value> --from literal=<key>:<value>

Create secret from file

kubectrl create secret generic <secret_name> --fromfile=<file_name>

Create job

kubectrl create job <job_name> --image=<image_name>

Create job from cronjob

kubectrl create job <job_name> -from=cronjob/<cronjob-name>

Create cronjob

kubectrl create cronjob --image=<image_name> -schedule='<cron-syntax>' -- <command> <args>

Monitoring Usage: Get node cpu and memory utilization

kubectrl top node <node_name>

Get pod cpu and memory utilization

kubectrl top pods <pod_name>

Node Commands: Describe node

kubectrl describe node <node_name>

Get node in yaml

kubectrl get node <node_name> -o yaml

Get node Drain node

kubectrl drain node <node_name>

Cordon node

kubectrl cordon node <node_name>

Uncordon node

kubectrl uncordon node <node_name>

Pod Commands: Get pod Command

kubectrl get pod <pod_name>

Get pod in yaml

kubectrl get pod <pod_name> -o yaml

Get pod wide

kubectrl get pod <pod_name> -o wide

Information Get pod with watch

kubectrl get pod <pod_name> -w

Edit pod

kubectrl edit pod <pod_name>

Describe pod

kubectrl describe pod <pod_name>

Delete pod

kubectrl delete pod <pod_name>

Log pod

kubectrl logs pod <pod_name>

Tail -f pod

kubectrl logs pod -f <pod_name>

Execute into pod

kubectrl exec -it pod <pod_name> /bin/bash

Running Temporary

kubectrl run <pod_name> --image=curlimages/curl --rm -it -Image restart=Never -- curl <destination>

Deployment Commands: Get deployment

kubectrl get deployment <deployment_name>

Get deployment in yaml

kubectrl get deployment <deployment_name> -o yaml

Get deployment wide information

kubectrl get deployment <deployment_name> -o wide

Edit deployment

kubectrl edit deployment <deployment_name>

Describe deployment

kubectrl describe deployment <deployment_name>

Delete deployment

kubectrl delete deployment <deployment_name>

Log deployment

kubectrl logs deployment/deployment_name -f

Update image

kubectrl set image deployment <deployment_name> <container_name>=<new_image_name>

Scale deployment with replicas

kubectrl scale deployment <deployment_name> --replicas <replicas>

Service Commands: Get service

kubectrl get service <service>

Get service in yaml

kubectl get service <service> -o yaml

Get service wide information

kubectl get service <service> -o wide

Edit service

kubectl edit service <service>

Describe service

kubectl describe service <service>

Delete service

kubectl delete service <service>

Endpoints Commands: Get endpoints

kubectl get endpoints <endpoints_name>

Ingress Commands: Get ingress

kubectl get ingress

Get ingress in yaml

kubectl get ingress -o yaml

Get ingress wide information

kubectl get ingress -o wide

Edit ingress

kubectl edit ingress <ingress_name>

Describe ingress

kubectl describe ingress <ingress_name>

Delete ingress

kubectl delete ingress <ingress_name>

DaemonSet Commands: Get daemonset

kubectl get daemonset <daemonset_name>

Get daemonset in yaml

kubectl get daemonset <daemonset_name> -o yaml

Edit daemonset

kubectl edit daemonset <daemonset_name>

Describe daemonset

kubectl describe daemonset <daemonset_name>

Delete daemonset

kubectl delete deployment <daemonset_name>

StatefulSet Commands: Get statefulset

kubectl get statefulset <statefulset_name>

Get statefulset in yaml

kubectl get statefulset <statefulset_name> -o yaml

Edit statefulset

kubecttl edit statefulset <statefulset_name>

Describe statefulset

kubecttl describe statefulset <statefulset_name>

Delete statefulset

kubecttl delete statefulset <statefulset_name>

ConfigMaps Commands: Get configmap

kubecttl get configmap <configmap_name>

Get configmap in yaml

kubecttl get configmap <configmap_name> -o yaml

Edit configmap

kubecttl edit configmap <configmap_name>

Describe configmap

kubecttl describe configmap <configmap_name>

Delete configmap

kubecttl delete configmap <configmap_name>

Secret Commands: Get secret

kubecttl get secret <secret_name>

Get secret in yaml

kubecttl get secret <secret_name> -o yaml

Edit secret

kubecttl edit secret <secret_name>

Describe secret

kubecttl describe secret <secret_name>

Delete secret

kubecttl delete secret <secret_name>

Rollout Commands: Restart deployment

kubecttl rollout restart deployment <deployment_name>

Undo deployment with the latest revision

kubecttl rollout undo deployment <deployment_name>

Undo deployment with specified revision

kubecttl rollout undo deployment <deployment_name> --to-revision <revision_number>

Get all revisions of deployment

kubecttl rollout history deployment <deployment_name>

Get specified revision of

kubecttl rollout history deployment <deployment_name> -deployment revision=<revision_number>

Job Commands: Get job

kubecttl get job <job_name>

Get job in yaml

kubectrl get job <job_name> -o yaml

Edit job in yaml

kubectrl edit job <job_name>

Describe job

kubectrl describe job <job_name>

Delete job

kubectrl delete job <job_name>

Cronjob Commands: Get cronjob

kubectrl get cronjob cronjob_name

Get cronjob in yaml

kubectrl get cronjob <cronjob_name> -o yaml

Edit cronjob

kubectrl edit cronjob <cronjob_name>

Describe cronjob

kubectrl describe cronjob <cronjob_name>

Delete cronjob

kubectrl delete cronjob <cronjob_name>

Network Policy Commands: Get networkpolicy

kubectrl get networkpolicy <networkpolicy_name>

Get networkpolicy in yaml

kubectrl get networkpolicy <networkpolicy_name> o yaml

Get networkpolicy wide information

kubectrl get networkpolicy <networkpolicy_name> o wide

Edit networkpolicy

kubectrl edit networkpolicy <networkpolicy_name>

Describe networkpolicy

kubectrl describe networkpolicy <networkpolicy_name>

Delete networkpolicy

kubectrl delete networkpolicy <networkpolicy_name>

Labels and Selectors: Show labels of node,pod and deployment

kubectrl get <node/pod/deployment> --showlabels

Attach labels to <node/pod/deployment>

kubectrl label <node/pod/deployment> <pod_name> <key>=<value>

Remove labels from <node/pod/deployment>

kubectrl label <node/pod/deployment> <pod_name> <key>-

Select node,pod and deployment by using labels

kubectrl get <node/pod/deployment> -l <key>=<value>