

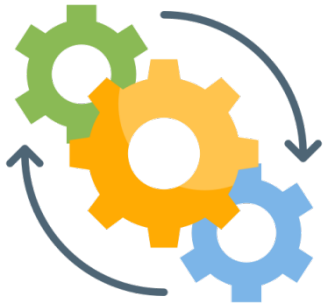


Kubernetes **Survival** **Guide**

@cicdtrail



Overview of Key *Kubernetes* Commands



**Operational
Commands**



**Troubleshooting
Commands**



**Advance
Debugging**



Author: Deepak Choudhary

LinkedIn: <https://www.linkedin.com/in/cdeepak-gb/>

@cicdtrail



Cluster Information

Kubernetes Cluster Management Commands

View Cluster Information:

\$ kubectl cluster-info

Displays the kubernetes control plane and services' addresses.





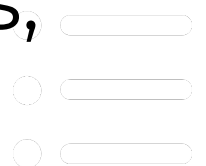
Cluster Information

Kubernetes Cluster Management Commands

List All Nodes:

\$ kubectl get nodes -o wide

Display a detailed view of all nodes, including status, roles and IP addresses.





Cluster Information

Kubernetes Cluster Management Commands

Check Node Health:

\$ kubectl describe node <node name>

Display a detailed information about a specific node.



Pod Management

Kubernetes Cluster Management Commands



List All Pods:

\$ kubectl get pods -A

List all pods across all namespaces.

@cicdtrail



Pod Management

Kubernetes Cluster Management Commands



Check Pod Details:

***\$ kubectl describe pod <pod name> -n
<namespace>***

Gives detailed information about the pod, including events.

@cicdtrail



Pod Management

Kubernetes Cluster Management Commands



List All Pods:

```
$ kubectl logs <pod-name> -n  
                <namespace>
```

View logs for a specific pod. Use -f to follow logs in real time.





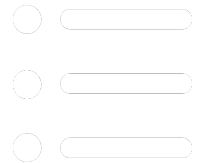
Deployment and Replica Set Management

Kubernetes Deployment and Replica Set Management
Commands

List Deployments:

\$ kubectl get deployments -A

Shows all deployments across namespaces.





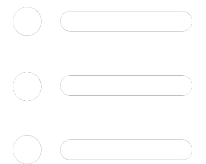
Deployment and Replica Set Management

Kubernetes Deployment and Replica Set Management
Commands

View Deployment Details:

```
$ kubectl describe deployment  
<deployment-name> -n  
<namespace>
```

Provides details like replicas,
strategy, and conditions.



@cicdtrail

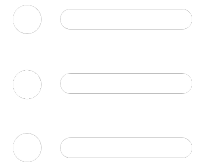


Deployment and Replica Set Management

Kubernetes Deployment and Replica Set Management
Commands

Scale a Deployment:

```
$ kubectl scale deployment  
<deployment-name> --  
replicas=<number> -n  
<namespace>
```





Deployment and Replica Set Management

Kubernetes Deployment and Replica Set Management
Commands

Restart a Deployment:

```
$ kubectl rollout restart  
deployment <deployment-name> -  
n <namespace>
```



Deployment and Replica Set Management

Kubernetes Deployment and Replica Set Management
Commands

Check Rollout Status:

```
$ kubectl rollout status  
deployment <deployment-name> -  
n <namespace>
```

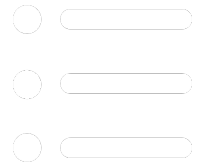


Deployment and Replica Set Management

Kubernetes Deployment and Replica Set Management
Commands

Rollback Deployment:

```
$ kubectl rollout undo deployment  
<deployment-name> -n  
<namespace>
```



Namespace Management

Kubernetes Namespace Management Commands



List Namespaces:

\$ kubectl get namespaces

@cicdtrail

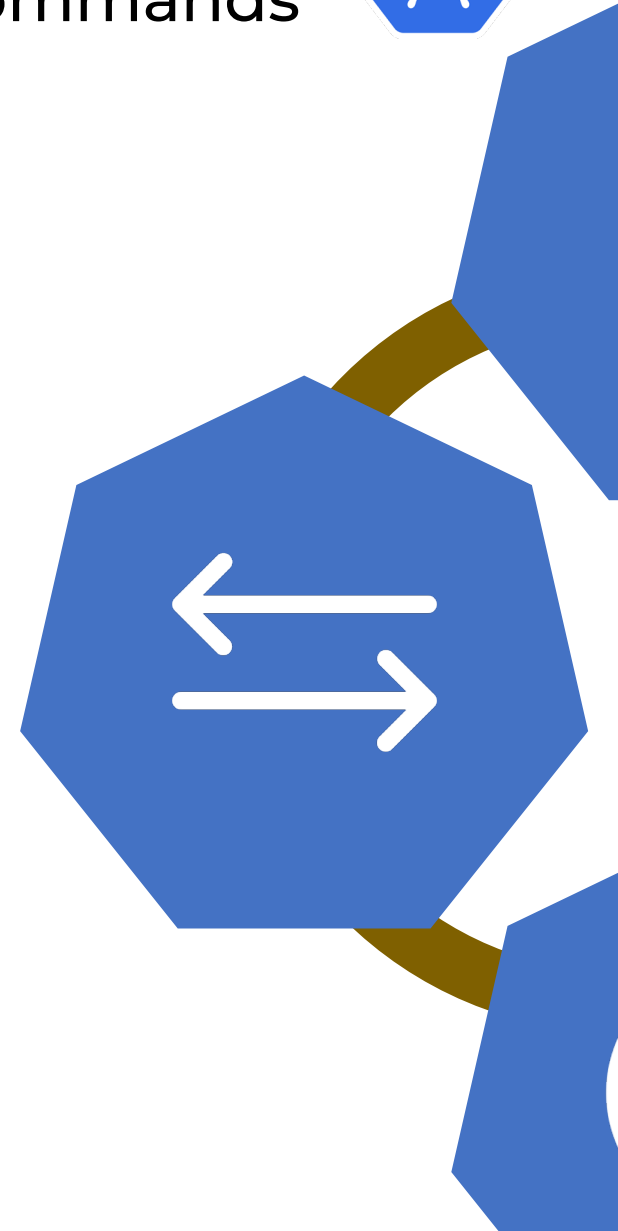
Namespace Management

Kubernetes Namespace Management Commands



Switch Namespace:

```
$ kubectl config set-context --current  
--namespace=<namespace>
```



@cicdtrail

Namespace Management

Kubernetes Namespace Management Commands



Create a Namespace:

```
$ kubectl create namespace  
<namespace>
```

@cicdtrail



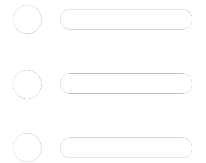


Service and Endpoint Management

Kubernetes Service and Endpoint Management Commands

List Services:

\$ kubectl get services -A



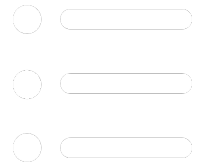


Service and Endpoint Management

Kubernetes Service and Endpoint Management Commands

Inspect Service Details:

\$ kubectl describe service <service-name> -n <namespace>



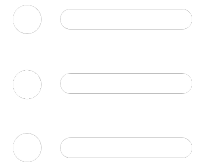


Service and Endpoint Management

Kubernetes Service and Endpoint Management Commands

Debug a Service Endpoint:

\$ kubectl get endpoints <service-name> -n <namespace>





Service and Endpoint Management

Kubernetes Service and Endpoint Management Commands

Port Forward a Service:

***\$ kubectl port-forward
service/<service-name> <local-
port>:<service-port> -n <namespace>***



ConfigMaps and Secrets

Kubernetes ConfigMaps and Secrets Commands



List ConfigMaps:

```
$ kubectl get configmaps -n  
  <namespace>
```

@cicdtrail



ConfigMaps and Secrets

Kubernetes ConfigMaps and Secrets Commands



Inspect a ConfigMap:

```
$ kubectl describe configmap  
<configmap-name> -n <namespace>
```

@cicdtrail



ConfigMaps and Secrets

Kubernetes ConfigMaps and Secrets Commands



List Secrets:

\$ kubectl get secrets -n <namespace>



ConfigMaps and Secrets

Kubernetes ConfigMaps and Secrets Commands



Decode a Secret:

```
$ kubectl get secret <secret-name> -n  
    <namespace> -o  
    jsonpath='{.data.<key>}' | base64 --  
    decode
```

@cicdtrail





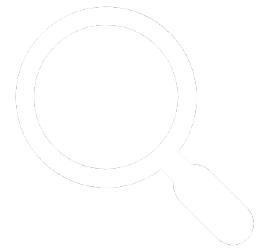
Troubleshooting

Kubernetes Troubleshooting Commands

Check Events in a Namespace:

\$ kubectl get events -n <namespace>

Displays recent events for troubleshooting issues.



@cicdtrail

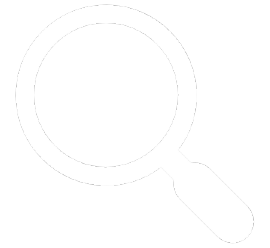


Troubleshooting

Kubernetes Troubleshooting Commands

Debug a Node:

\$ kubectl describe node <node-name>



@cicdtrail

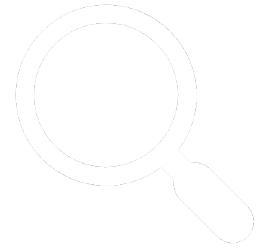


Troubleshooting

Kubernetes Troubleshooting Commands

Debug Pod Resource Usage:

\$ kubectl top pod -n <namespace>



@cicdtrail

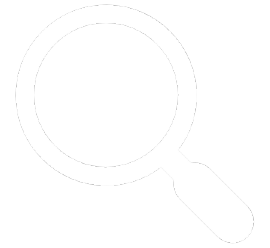


Troubleshooting

Kubernetes Troubleshooting Commands

Debug Node Resource Usage:

\$ kubectl top node



@cicdtrail



Troubleshooting

Kubernetes Troubleshooting Commands

Identify Pending Pods:

\$ kubectl get pods --field-selector=status.phase=Pending -A

@cicdtrail



Troubleshooting

Kubernetes Troubleshooting Commands

Check Pod Scheduling Issues:

```
$ kubectl describe pod <pod-name> -  
n <namespace>
```

@cicdtrail

Persistent Volume (PV) and Persistent Volume Claim (PVC)



Kubernetes Persistent Volume (PV) and Persistent Volume Claim (PVC) Commands

List Persistent Volumes:

\$ kubectl get pv



Persistent Volume (PV) and Persistent Volume Claim (PVC)



Kubernetes Persistent Volume (PV) and Persistent Volume Claim (PVC) Commands

List Persistent Volume Claims:

```
$ kubectl get pvc -n <namespace>
```



Persistent Volume (PV) and Persistent Volume Claim (PVC)



Kubernetes Persistent Volume (PV) and Persistent Volume Claim (PVC) Commands

Inspect PVC Details:

```
$ kubectl describe pvc <pvc-name> -n  
  <namespace>
```



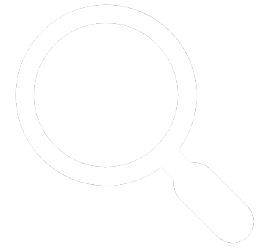


Managing Kubernetes Configurations

Kubernetes Managing Kubernetes Configurations Commands

View Current Context:

\$ kubectl config current-context



@cicdtrail

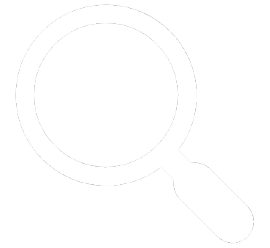


Managing Kubernetes Configurations

Kubernetes Managing Kubernetes Configurations Commands

List All Context:

\$ kubectl config get-contexts



@cicdtrail

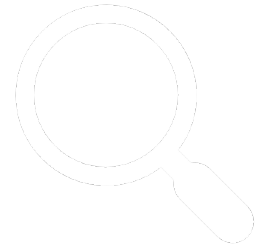


Managing Kubernetes Configurations

Kubernetes Managing Kubernetes Configurations Commands

Switch Context:

***\$ kubectl config use-context
<context-name>***



@cicdtrail

Deleting Resources

Kubernetes Deleting Resources Commands



Delete a Pod:

```
$ kubectl delete pod <pod-name> -n  
  <namespace>
```



Deleting Resources

Kubernetes Deleting Resources Commands



**Delete All Pods in a
Namespace:**

***\$ kubectl delete pods --all -n
<namespace>***



@cicdtrail

Deleting Resources

Kubernetes Deleting Resources Commands



Delete a Deployment:

*\$ kubectl delete deployment
<deployment-name> -n <namespace>*



@cicdtrail

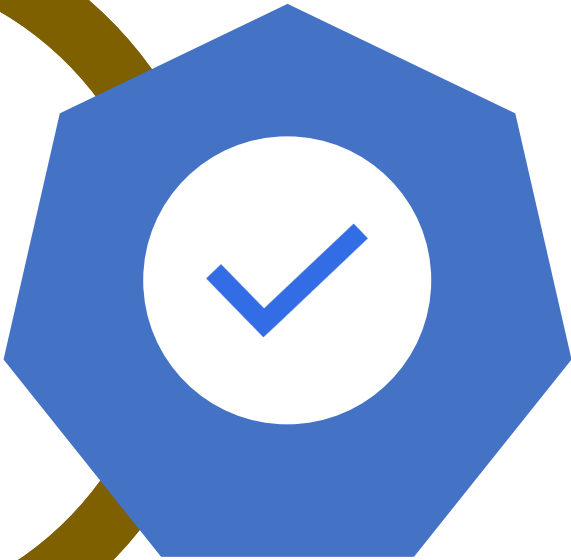


YAML Files

Kubernetes YAML Files Commands

**Apply Configuration from
YAML:**

\$ kubectl apply -f <file.yaml>



@cicdtrail



YAML Files

Kubernetes YAML Files Commands

Delete Resource from YAML:

\$ kubectl delete -f <file.yaml>



@cicdtrail

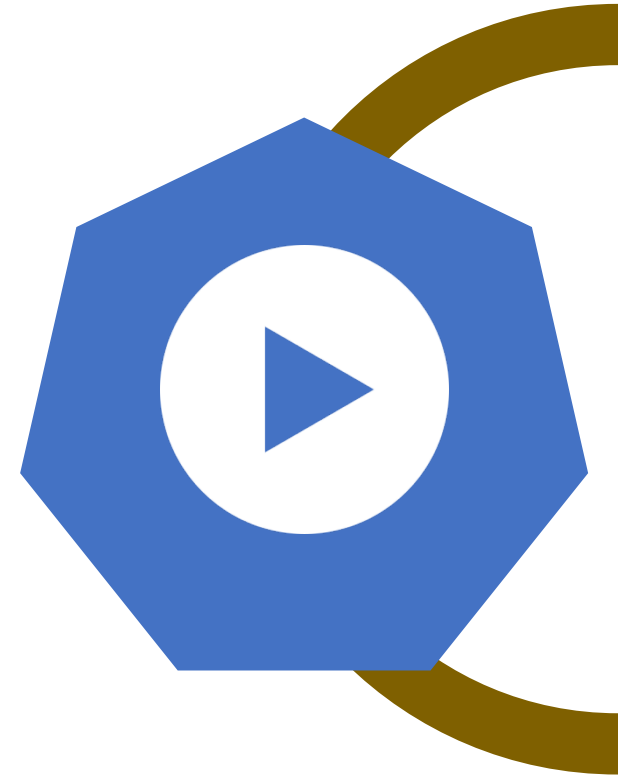
Miscellaneous

Kubernetes Miscellaneous Commands



Run a Pod (One-Off):

```
$ kubectl run <pod-name> --  
image=<image> -n <namespace>
```



@cicdtrail

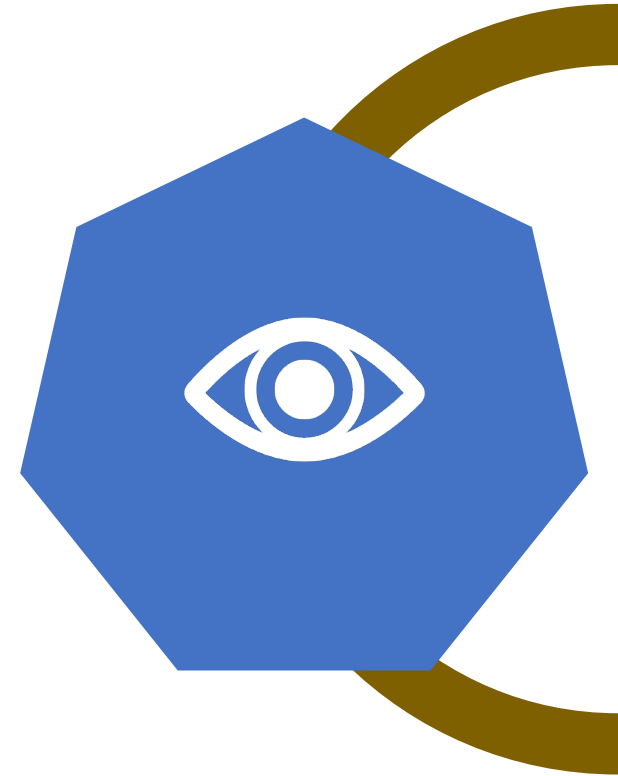
Miscellaneous

Kubernetes Miscellaneous Commands



Expose a Deployment as a Service:

```
$ kubectl expose deployment  
  <deployment-name> --  
type=LoadBalancer --port=<port> -n  
  <namespace>
```



@cicdtrail



Advanced Debugging

Kubernetes Advanced Debugging Commands



Debug a Pod Using `kubectl debug` (*requires Kubernetes 1.18+*):

```
$ kubectl debug pod/<pod-name> -n  
<namespace> --image=<debug-  
image>
```

@cicdtrail



Advanced Debugging

Kubernetes Advanced Debugging Commands



Debug a Node Using `kubectl debug`:

```
$ kubectl debug node/<node-name>  
-it --image=busybox
```

Start a debugging pod on a specific node.

@cicdtrail

Manage Jobs and Cron Jobs

Kubernetes Manage Jobs and Cron Jobs Commands



List All Jobs:

\$ kubectl get jobs -n <namespace>

@cicdtrail



Manage Jobs and CronJobs

Kubernetes Manage Jobs and Cron Jobs Commands



View CronJobs:

*\$ kubectl get cronjobs -n
<namespace>*

@cicdtrail



Manage Jobs and CronJobs

Kubernetes Manage Jobs and Cron Jobs Commands



Run a CronJob Manually:

```
$ kubectl create job --  
from=cronjob/<cronjob-name> <job-  
name> -n <namespace>
```

@cicdtrail





Working with Labels and Annotations

Kubernetes Working with Labels and Annotations Commands

List Pods with Specific Labels:

```
$ kubectl get pods -l <label-key>=<label-value> -n <namespace>
```



Working with Labels and Annotations

Kubernetes Working with Labels and Annotations Commands

Add a Label to a Resource:

```
$ kubectl label pod <pod-name>  
<label-key>=<label-value> -n  
<namespace>
```

@cicdtrail



Working with Labels and Annotations

Kubernetes Working with Labels and Annotations Commands

Remove a Label:

```
$ kubectl label pod <pod-name>  
<label-key>- -n <namespace>
```

@cicdtrail



Working with Labels and Annotations

Kubernetes Working with Labels and Annotations Commands

Add an Annotation:

```
$ kubectl annotate pod <pod-name>  
<annotation-key>=<annotation-  
value> -n <namespace>
```

@cicdtrail

Helm Commands (if using Helm)

Kubernetes Helm Commands



List Installed Helm Charts:

\$ helm list -A

@cicdtrail



Helm Commands (if using Helm)

Kubernetes Helm Commands



Install a Helm Chart :

```
$ helm install <release-name> <chart-name> --namespace <namespace>
```

@cicdtrail



Helm Commands (if using Helm)

Kubernetes Helm Commands



Upgrade a Helm Release :

```
$ helm upgrade <release-name>  
<chart-name> -n <namespace>
```



@cicdtrail

Helm Commands (if using Helm)

Kubernetes Helm Commands



Rollback a Helm Release:

```
$ helm rollback <release-name>  
<revision> -n <namespace>
```

@cicdtrail





Working with Network Policies

Kubernetes Working with Network Policies Commands

List Network Policies:

***\$ kubectl get networkpolicies -n
<namespace>***

@cicdtrail



Working with Network Policies

Kubernetes Working with Network Policies Commands

Inspect a Network Policy:

***\$ kubectl describe networkpolicy
<networkpolicy-name> -n
<namespace>***

@cicdtrail

Monitoring and Metrics

Kubernetes Monitoring and Metrics Commands



**View Horizontal Pod
Autoscaler (HPA) Details:**

\$ kubectl get hpa -n <namespace>

@cicdtrail

Monitoring and Metrics

Kubernetes Monitoring and Metrics Commands



Inspect Resource Utilization for HPA:

```
$ kubectl describe hpa <hpa-name> -  
n <namespace>
```

@cicdtrail



Monitoring and Metrics

Kubernetes Monitoring and Metrics Commands



Check Cluster Metrics *(if Metrics Server is installed):*

\$ kubectl top pod -n <namespace>
\$ kubectl top node

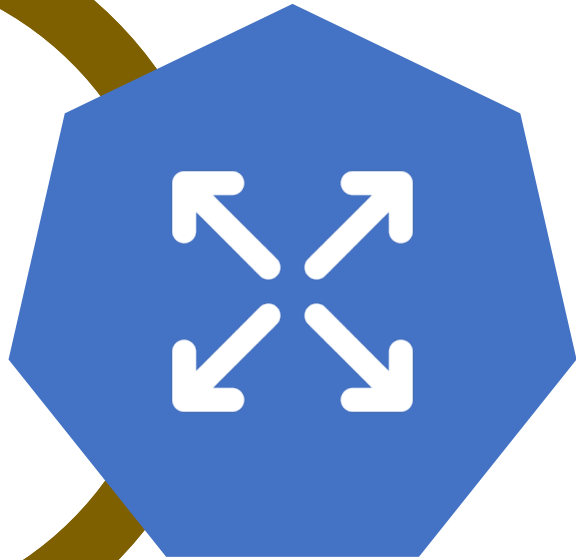
@cicdtrail



Storage Management

Kubernetes Storage Management Commands

Resize a PVC:



```
$ kubectl patch pvc <pvc-name> -n  
<namespace> -p '{"spec":  
{"resources": {"requests": {"storage":  
"10Gi"}}}]'
```

@cicdtrail



Storage Management

Kubernetes Storage Management Commands



Delete Persistent Volume and Reclaim Space:

```
$ kubectl delete pvc <pvc-name> -n  
<namespace>
```

View and Manage Logs

Kubernetes View and Manage Logs Commands



View Multiple Container Logs:

```
$ kubectl logs <pod-name> -c  
<container-name> -n <namespace>
```



View and Manage Logs

Kubernetes View and Manage Logs Commands



Tail Logs in Real-Time:

```
$ kubectl logs -f <pod-name> -n  
  <namespace>
```



@cicdtrail

View and Manage Logs

Kubernetes View and Manage Logs Commands



View Previous Logs:

```
$ kubectl logs <pod-name> --previous  
-n <namespace>
```

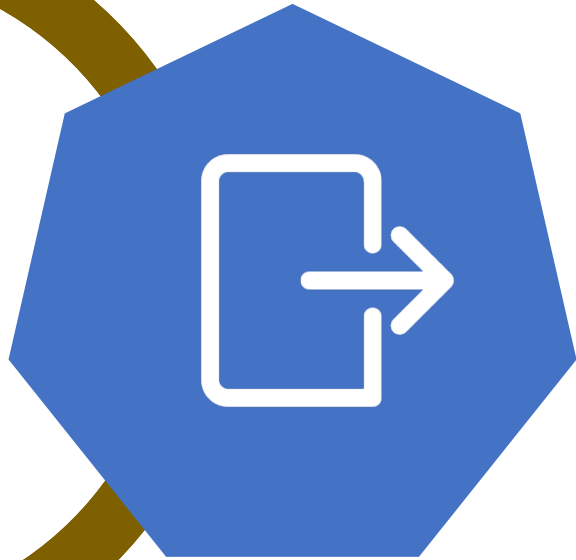




Customizing Output

Kubernetes Customizing Output Commands

Output Resources as YAML/JSON:



```
$ kubectl get pods -n <namespace> -  
o yaml  
$ kubectl get pods -n <namespace> -  
o json
```

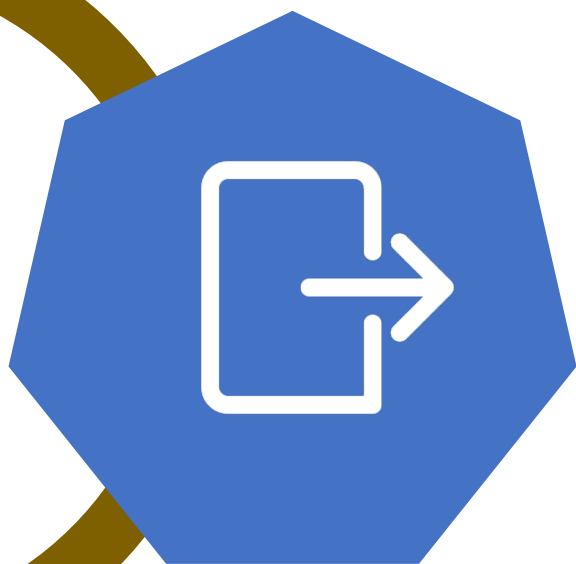
@cicdtrail



Customizing Output

Kubernetes Customizing Output Commands

Wide Output for More Details:



```
$ kubectl get pods -o wide -n  
<namespace>
```

@cicdtrail

Working with CRDs (Custom Resource Definitions)

Kubernetes Working with CRDs (Custom Resource Definitions) Commands

List CRDs:

\$ kubectl get crds

@cicdtrail



Working with CRDs (Custom Resource Definitions)

Kubernetes Working with CRDs (Custom Resource Definitions) Commands

Describe a CRD:

\$ kubectl describe crd <crd-name>

@cicdtrail



Working with CRDs (Custom Resource Definitions)

Kubernetes Working with CRDs (Custom Resource Definitions) Commands

Apply a CRD Definition:

```
$ kubectl apply -f <crd-definition.yaml>
```

@cicdtrail



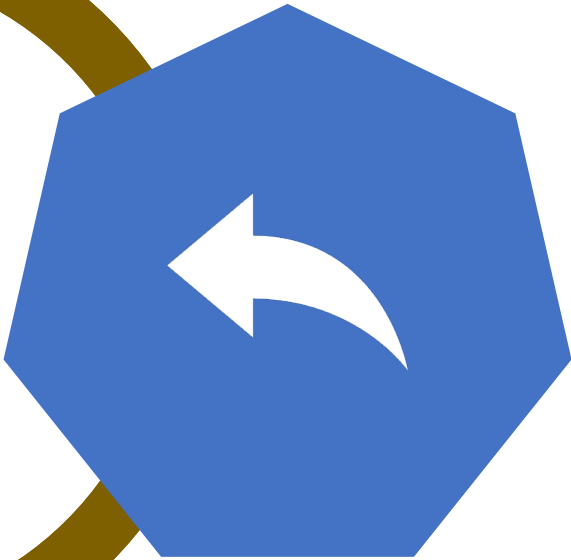


Accessing the Kubernetes API

Kubernetes API Commands

Retrieve the Cluster API
Server URL:

\$ kubectl cluster-info



@cicdtrail

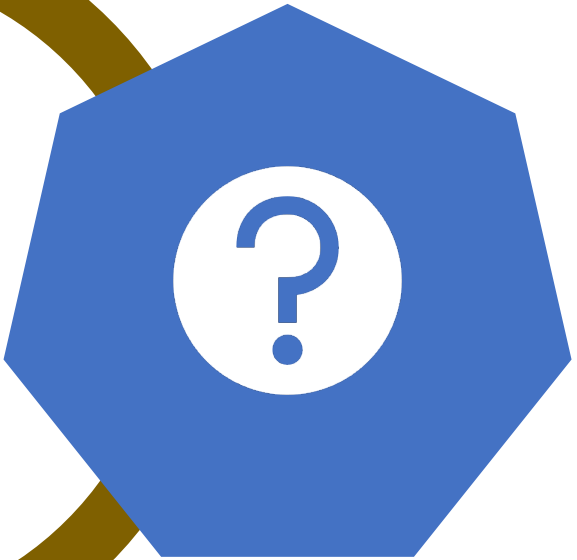


Accessing the Kubernetes API

Kubernetes API Commands

Proxy Requests to the API
Server:

\$ kubectl proxy



@cicdtrail

Troubleshooting and Debugging Commands

Kubernetes Troubleshooting and Debugging Commands



Test DNS Resolution:

```
$ kubectl exec -it <pod-name> -n  
<namespace> -- nslookup <service-  
name>
```

@cicdtrail



Troubleshooting and Debugging Commands

Kubernetes Troubleshooting and Debugging Commands



Identify Failing Pods:

\$ kubectl get pods --field-selector=status.phase=Failed -A

@cicdtrail



Troubleshooting and Debugging Commands

Kubernetes Troubleshooting and Debugging Commands



Check Pod Events:

```
$ kubectl get events -n <namespace>  
--sort-  
by='.metadata.creationTimestamp'
```

@cicdtrail





***Share it with your team
because
the **best engineers** are
the ones who **lift others**
up.***

@cicdtrail

Author: Deepak Choudhary

LinkedIn: <https://www.linkedin.com/in/cdeepak-gb/>



@cicdtrail



LinkedIn:

<https://www.linkedin.com/groups/12742133/>

<https://www.linkedin.com/company/cicdtrail/>

@cicdtrail