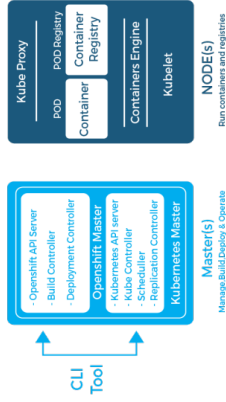


# KUBERNETES CHEAT SHEET

## KUBERNETES

- It is an open source platform for automating deployment and scaling of containers across clusters of hosts providing container centric infrastructure.
- It is a container orchestrator and can run Linux containers:
  - Launch container.
  - Maintain and monitor container site.
  - Performs container-oriented networking



## Key Concepts

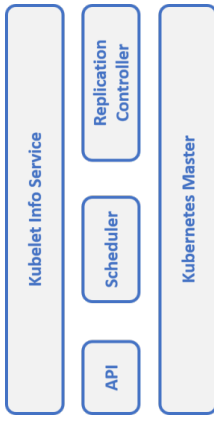
Now let's discuss the key points of this architecture.

- Pod:** These are the group of containers.
- Labels:** These are used to identify the pods.
- Kubelet:** They are container agents, responsible for maintaining the set of pods.
- Proxy:** They are the Load balancer for pods, helping in distributing tasks across the pods.
- ETCD:** A Metadata service.
- Cadviser:** For resource usage and performance stats.
- Replication controller:** It manages pod replication.
- Scheduler:** Used for pod scheduling in worker nodes.
- API server:** Kubernetes API server.

Now let's understand the role Master and Node play in the Kubernetes Architecture.

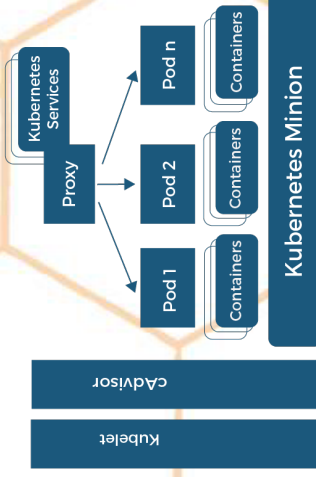
## Master

- It is responsible for maintaining the desired state for the cluster you are working on.
- "Master" indicates a set of processes that are used to manage the cluster.
- Contains info, API, scheduler, replication controllers, and master.



## Worker Nodes / Minions

- Also called as a minion. It contains the services necessary to run the pods that are managed by the master.
- Some services include: container runtime, Kubelet, kube-proxy.
- Contains: Kubelet, cadvisor, services, pods and containers.



## Features

- Automated scheduling:** provides an advanced scheduler that helps launch container on cluster nodes
- Self healing:** reschedule, replace and restart dead containers.
- Automated rollouts and rollbacks:** supports rollback for systems incase of a failure. Enables rollout and rollback for the desired state.
- Horizontal scaling:** can scale up and down the app as per required. Can also be automated wrt CPU usage.
- Service discovery and load balancing:** uses unique ip and dns name to containers. This helps identify them across different containers.

## Kubectl Command List

### Pods and Container Introspection

COMMANDS	FUNCTION	clusterrolebindings	clusterroles
Kubectl get pods	Lists all current pods	controllerrevisions	crd=custom resource definition
Kubectl describe pod<name>	Describes the pod names		csr= certificate signing requests
Kubectl get rc	List all replication controllers	cs=component status	ep=end points
Kubectl get rc -- namespace="namespace"	Lists replication controllers in namespace	ds= daemon sets	ing= ingress
Kubectl describe rc <name>	Shows the replication controller name	hpa= autoscaling	limits=limit ranges
Kubectl get svc	Lists the services	limits=limit ranges	Netpol- network policies
Kubectl describe svc<name>	Shows the service name	ns= namespaces	pdb= pod
Kubectl delete pod<name>	Deletes the pod	Pod preset	Pod templates
Kubectl get nodes -w	Watch nodes continuously	Pvc= pod security volumes	pvc= persistent volume claims
		rc= replication controllers	Role bindings
		roles	sa=service account
		sc= storage classes	sts= stateful sets

### Cluster Introspection

FUNCTION	COMMAND
Execute command on service by selecting container.	Kubectl exec<service><commands>[-c \$container]
Get logs from service for a container	Kubectl logs -f<name>>[-c \$container]
Watch the kubelet logs	Watch -n 2 cat /var/log/kublet.log
Show metrics for node	Kubectl top node
Show metrics for pods	Kubectl top pod

## Other Quick Commands

- Launch a pod with a name an image:** Kubectl run<name> --image=<image>name>
- Create a service in <manifest.yaml>:** Kubectl create -f <manifest.yaml>
- Scale replication counter to count the number of instances:** Kubectl scale --replicas=<count>

- Map external port to internal replication port:** Expose rc<name> --port=<external>-target-port=<internal>
- To stop all pod in <ns>:** Kubectl drain<ns> --delete-local-data--force--ignore-daemonset
- Allow master nodes to run pods:** Kubectl taint nodes --all node-role.kubernetes.io/master-