* **Cluster** – Group of physical or virtual servers wherein  Kubernetes is installed
* **Node (Master)** – Physical or virtual server that controls the Kubernetes cluster
* **Node (Worker)** – Physical or virtual servers where  workloads run in a given container technology
* **Pods** – Group of containers and volumes which share the  same network namespace
* **Labels** – User defined Key:Value pair associated to Pods
* **Master** – Control plane components which provide access  point for admins to manage cluster workloads
* **Service** – An abstraction which serves as a proxy for a group  of Pods performing a “service”

Kubernetes objects

**Workloads**

* Container
* CronJob / cronjobs / cj
* DaemonSet / daemonsets / ds
* Deployment / deployments / deploy
* Job / jobs
* Pod / pods / po
* ReplicaSet / replicasets / rs
* ReplicationController / replicationcontrollers / rc
* StatefulSet / statefulsets / sts

**Services**

* Endpoints / endpoints / ep
* EndpointSlice / endpointslices
* Ingress / ingresses / ing
* IngressClass / ingressclasses
* Service / services / svc

**Config & Storage**

* ConfigMap / configmaps / cm
* CSIDriver / csidrivers
* CSINode / csinodes
* Secret / secrets
* PersistentVolumeClaim / persistentvolumeclaims / pvc
* StorageClass / storageclasses / sc
* CSIStorageCapacity
* Volume
* VolumeAttachment / volumeattachments

**Clusters**

* APIService / apiservices
* Binding / bindings
* CertificateSigningRequest / certificatesigningrequests / csr
* ClusterRole / clusterroles
* ClusterRoleBinding / clusterrolebindings
* ComponentStatus / componentstatuses/cs
* FlowSchema / flowschemas
* Lease / leases
* LocalSubjectAccessReview / localsubjectaccessreviews
* Namespace / namespaces/ns
* NetworkPolicy / networkpolicies / netpol
* Node / nodes / no
* PersistentVolume / persistentvolumes / pv
* PriorityLevelConfiguration / prioritylevelconfigurations
* ResourceQuota / resourcequotas / quota
* Role / roles
* RoleBinding / rolebindings
* RuntimeClass / runtimeclasses
* SelfSubjectAccessReview / selfsubjectaccessreviews
* SelfSubjectRulesReview / selfsubjectrulesreviews
* ServiceAccount / serviceaccounts / sa
* StorageVersion
* SubjectAccessReview / subjectaccessreviews
* TokenRequest
* TokenReview / tokenreviews

**Metadata**

* ControllerRevision / controllerrevisions
* CustomResourceDefinition / customresourcedefinitions / crd,crds
* Event / events / ev
* LimitRange / limitranges / limits
* HorizontalPodAutoscaler / horizontalpodautoscalers / hpa
* MutatingWebhookConfiguration / mutatingwebhookconfigurations
* ValidatingWebhookConfiguration / validatingwebhookconfigurations
* PodTemplate / podtemplates
* PodDisruptionBudget / poddisruptionbudgets / pdb
* PriorityClass / priorityclasses / pc
* PodSecurityPolicy / podsecuritypolicies / psp

In general, any of these commands will work with any of these objects. So rather than:

kubectl get pods

You can use:

kubectl get deployments