

Application Observability and Maintenance for CKAD

Understand API Deprecations



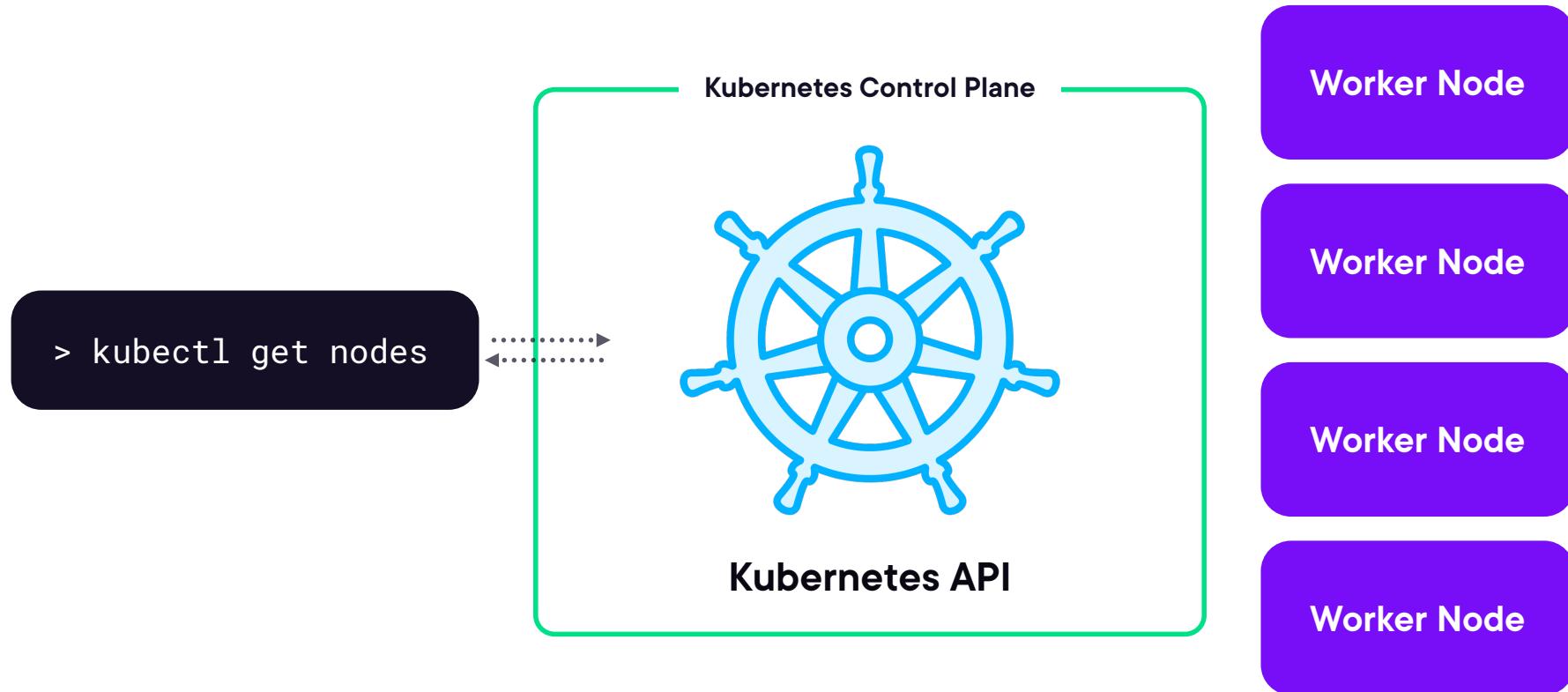
Elle Krout

Principal Course Author, Pluralsight

Kubernetes API Versioning

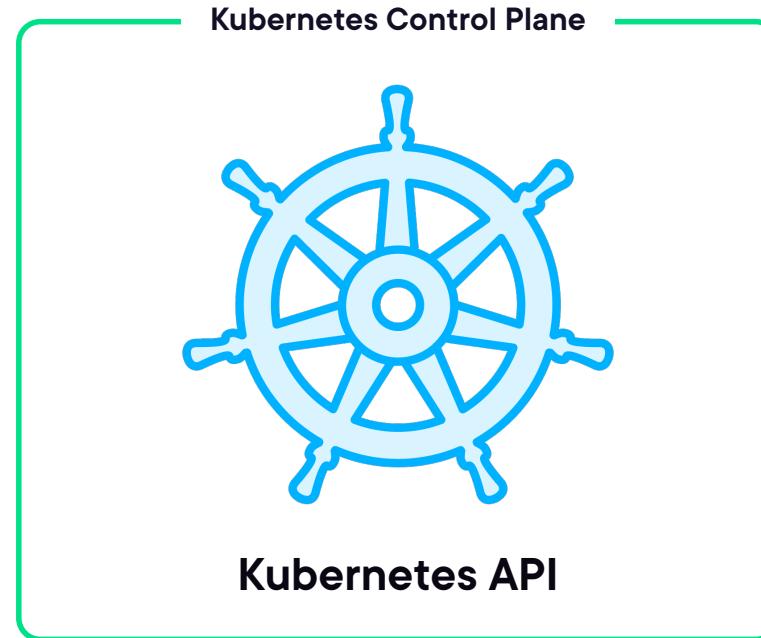


Kubernetes is Driven by RESTful APIs



Kubernetes is Driven by RESTful APIs

```
apiVersion: v1  
kind: Pod
```



Worker Node

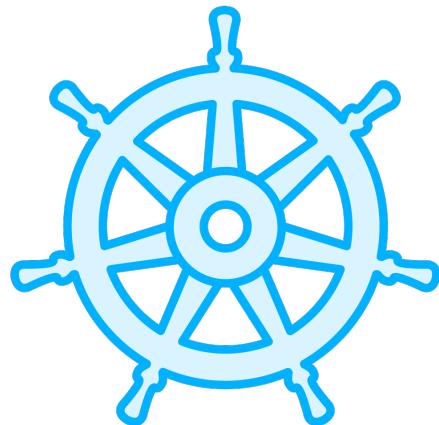
Worker Node

Worker Node

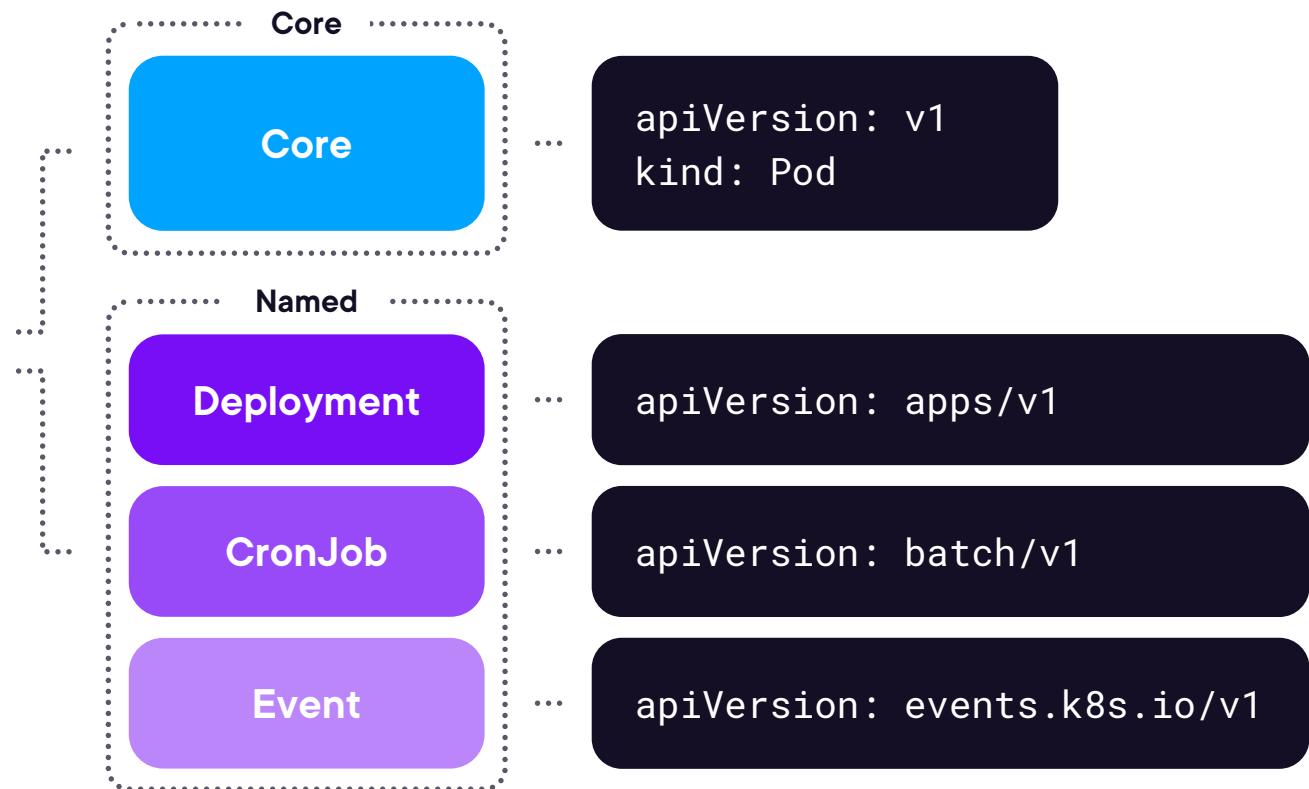
Worker Node



Kubernetes APIs



Kubernetes API



Gather Information About API Resource

```
> kubectl explain deployment
```

GROUP: apps

KIND: Deployment

VERSION: v1

DESCRIPTION:

Deployment enables declarative updates for Pods and ReplicaSets.

FIELDS:

apiVersion <string>

APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info:

...



Gather Information About API Resource

```
> kubectl explain pod
```

KIND: Pod

VERSION: v1

DESCRIPTION:

Pod is a collection of containers that can run on a host. This resource is created by clients and scheduled onto hosts.

FIELDS:

apiVersion <string>

APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info:

...



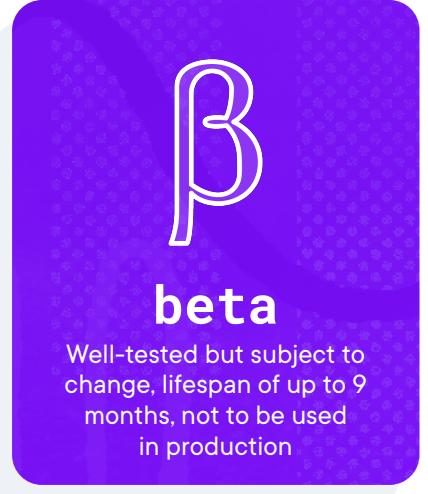
API Versioning



α

alpha

Experimental, subject to change, can be removed, disabled by default



β

beta

Well-tested but subject to change, lifespan of up to 9 months, not to be used in production



stable

Production-ready, cannot be dropped, no breaking changes, enabled by default



View Kubernetes Cluster Version

> `kubectl version`

Client Version: v1.33.1

Kustomize Version: v5.6.0

Server Version: v1.33.1



View Kubernetes Cluster Version

```
> kubectl version -o yaml
```

```
clientVersion:  
  buildDate: "2025-05-15T08:27:33Z"  
  compiler: gc  
  gitCommit: 8adc0f041b8e7ad1d30e29cc59c6ae7a15e19828  
  gitTreeState: clean  
  gitVersion: v1.33.1  
  goVersion: go1.24.2  
  major: "1"  
  minor: "33"  
  platform: darwin/amd64  
kustomizeVersion: v5.6.0  
serverVersion:  
  buildDate: "2025-05-15T08:19:08Z"  
...
```



View API Groups and Versions on Cluster

```
> kubectl api-versions
```

```
admissionregistration.k8s.io/v1
admissionregistration.k8s.io/v1beta1
apiextensions.k8s.io/v1
apiregistration.k8s.io/v1
apps/v1
authentication.k8s.io/v1
authorization.k8s.io/v1
autoscaling/v1
autoscaling/v2
batch/v1
certificates.k8s.io/v1
coordination.k8s.io/v1
coordination.k8s.io/v1alpha2
...
```



View API Groups and Versions on Cluster

```
> kubectl api-versions
```

```
admissionregistration.k8s.io/v1
```

```
admissionregistration.k8s.io/v1beta1
```

```
apiextensions.k8s.io/v1
```

```
apiregistration.k8s.io/v1
```

```
apps/v1
```

```
authentication.k8s.io/v1
```

```
authorization.k8s.io/v1
```

```
autoscaling/v1
```

```
autoscaling/v2
```

```
batch/v1
```

```
certificates.k8s.io/v1
```

```
coordination.k8s.io/v1
```

```
coordination.k8s.io/v1alpha2
```

```
...
```



View Kubernetes Cluster Version

> `kubectl api-resources`

NAME	SHORTNAMES	APIVERSION	NAMESPACED	KIND
bindings		v1	true	Binding
configmaps	cm	v1	true	ConfigMap
endpoints	ep	v1	true	Endpoints
apiservices		apiregistration.k8s.io/v1	false	APIService
daemonsets	ds	apps/v1	true	DaemonSet
deployments	deploy	apps/v1	true	Deployment
replicasets	rs	apps/v1	true	ReplicaSet
statefulsets	sts	apps/v1	true	StatefulSet
tokenreviews		authentication.k8s.io/v1	false	TokenReview
cronjobs	cj	batch/v1	true	CronJob
jobs		batch/v1	true	Job
leases		coordination.k8s.io/v1	true	Lease



View Kubernetes Cluster Version

```
> kubectl api-resources --api-group=batch
```

NAME	SHORTNAMES	APIVERSION	NAMESPACED	KIND
cronjobs	cj	batch/v1	true	CronJob
jobs		batch/v1	true	Job

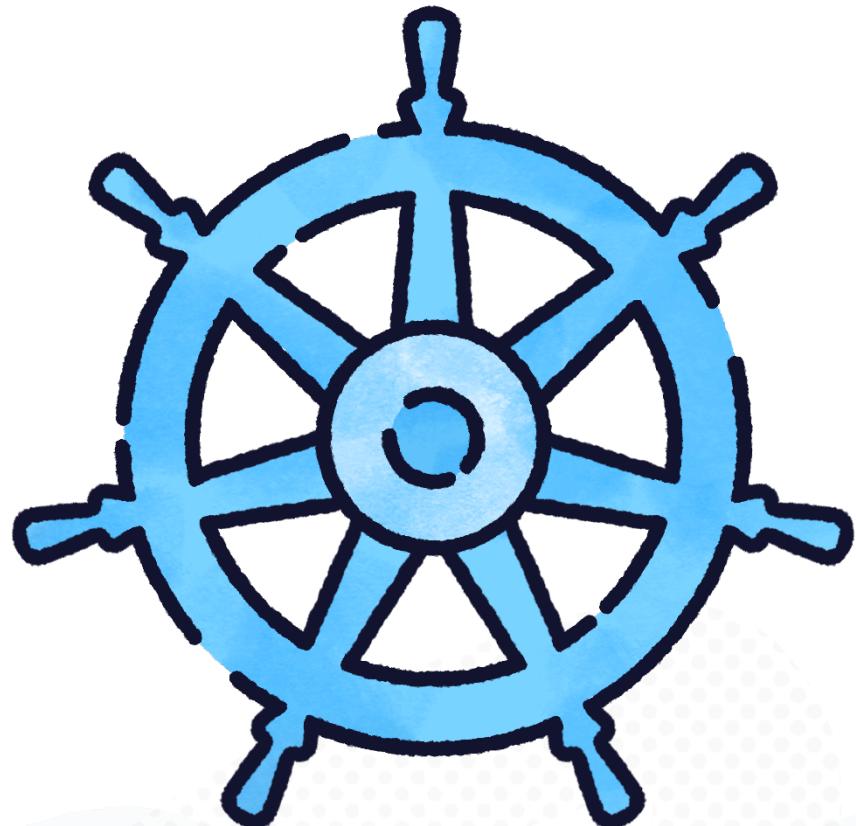


**Understanding API
versioning is essential**

**Avoid using deprecated
resources**

Ensure compatibility

Aid in API deprecation



Kubernetes API Deprecation Policies



API deprecation is the phasing out of older versions of API resources in favor of newer, preferred versions.



Kubernetes API Lifecycle



Rule #1

“API elements may only be removed by incrementing the version of the API group.”



Rule #1

No breaking or definition-altering changes

If you add a feature in alpha, it cannot be removed in the same alpha version



Rule #2

“API objects must be able to round-trip between API versions in a given release without information loss, with the exception of whole REST resources that do not exist in some versions.”



Rule #2

You must be able to convert between different Kubernetes version without data loss

If an API version adds something new, it must be encoded so an older version can work around it

A resource should be able to be converted between versions and look identical



Rule #3

“An API version in a given track may not be deprecated in favor of a less stable API version.”



Rule #3

**A version cannot go from stable to beta or
beta to alpha**



Rule #4a

“API lifetime is determined by the API stability level.”



Rule #4a

API version determine lifecycle

beta versions can exist for no more than 9 months or 3 minor releases

alpha releases can be removed without deprecation notice

stable releases should not be removed



Rule #4b

“The ‘preferred’ API version and the ‘storage version’ for a given group may not advance until after a release has been made that supports both the new version and the previous version.”



Rule #4b

You should always be able to upgrade to a new release of Kubernetes and roll back safely.



The Kubernetes API Policy Governs

REST resources

Fields of REST
resources

Annotations of REST
resources

Enumerated or
content values

Component config
structures



**[https://kubernetes.io/docs/
reference/using-api/
deprecation-policy](https://kubernetes.io/docs/reference/using-api/deprecation-policy)**



Working with Kubernetes Versions



View Enabled Alpha/Beta APIs

```
> sudo $EDITOR /etc/kubernetes/manifests/kube-apiserver.yaml  
...  
- --runtime-config=admissionregistration.k8s.io/v1beta1  
- --secure-port=6443  
- --service-account-issuer=https://kubernetes.default.svc.cluster.local  
- --service-account-key-file=/etc/kubernetes/pki/sa.pub  
...
```



View Enabled Alpha/Beta APIs

```
> sudo $EDITOR /etc/kubernetes/manifests/kube-apiserver.yaml  
  
...  
    - --runtime-config=admissionregistration.k8s.io/v1beta1,coordination.k8s.io/  
v1alpha2  
    - --secure-port=6443  
    - --service-account-issuer=https://kubernetes.default.svc.cluster.local  
    - --service-account-key-file=/etc/kubernetes/pki/sa.pub  
...  
...
```



View Preferred APIs

```
> kubectl proxy 8001 <&>
> curl http://localhost:8001/apis/batch
```

...

```
"versions": [
  {
    "groupVersion": "batch/v1",
    "version": "v1"
  }
],
"preferredVersion": {
  "groupVersion": "batch/v1",
  "version": "v1"
}
```

...



Kill Kubernetes Background Process

```
> pgrep kubectl
```

```
30818 # Your process ID will be different
```

```
> kill <PID>
```

```
> pgrep kubectl
```

```
# No PID should be returned
```



Exam Tip!

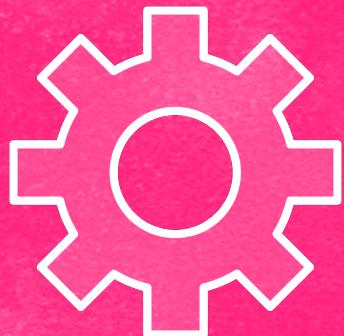
Practice adding and removing unstable APIs to reinforce the location and commands needed to check for deprecated APIs



Demo: Identifying and Updating Kubernetes APIs



Course Environment



Kubernetes 1.33

For this demo, the exact version matters!



Exam Tip!

The CKAD will provide a workstation to run `kubectl` from and multiple clusters to work with — be sure you're using the right cluster!



Exam Scenario



Update the Kubernetes cluster to use the v1beta1 admissionregistration.k8s.io API and v1alpha2 for the Lease API

Confirm the APIs were updated by viewing available versions

Confirm the preferred version for the updated APIs

