

Mini Article – Nov 2022 – DevOps

API Access & RESTFUL & Optimistic Concurrency

Kubernetes has a powerful REST-based API. The entire architecture is API-driven. Knowing where to find resource endpoints and understanding how the API changes between versions can be important to ongoing administrative tasks, as there is much ongoing change and growth. Starting with v1.16 deprecated objects are no longer honored by the API server.

As we learned in the Kubernetes Architecture chapter, the main agent for communication between cluster agents and from outside the cluster is the kube-apiserver. A curl query to the agent will expose the current API groups. Groups may have multiple versions, which evolve independently of other groups, and follow a domain-name format with several names reserved, such as single-word domains, the empty group, and any name ending in .k8s.io.

RESTFUL

kubectl makes API calls on your behalf, responding to typical HTTP verbs (GET, POST, DELETE). You can also make calls externally, using curl or another program. With the appropriate certificates and keys, you can make requests, or pass JSON files to make configuration changes. See the following command:

```
$ curl --cert userEd.pem --key userEd-key.pem \
--cacert /path/to/ca.pem \
https://k8sServer:6443/api/v1/pods
```

The ability to impersonate other users or groups, subject to RBAC configuration, allows a manual override authentication. This can be helpful for debugging authorization policies of other users.

Optimistic Concurrency

The default serialization for API calls must be JSON. Google's **protobuf serialization** is another option, but this remains experimental. While we may work with files in a YAML format, they are converted to and from JSON.

Kubernetes uses the resourceVersion value to determine API updates and implement optimistic concurrency. In other words, an object once it is read is not locked until the object is written. During an updated call to an object, the resourceVersion is checked, and a 409 CONFLICT is returned, should the number have changed. The resourceVersion is currently backed via the modifiedIndex parameter in the etcd database, and is unique to the namespace, kind, and server. Operations such as WATCH or GET which do not change an object, do not update this value.