

Force Remove namespaces stuck in terminating/Deleting state

vipulmalkoti

May '21

There are multiple scenarios where a K8s admin/user might need to delete an existing namespace for migrating resource, no need for the namespace, some problems, convention gone wrong etc.

Generally we can just run something as simple as

\$ kubectl delete ns mynamespace

to delete this, however in certain situations where an issue might be persisting in your config or a deployment this might not work. This can lead to this namespace being stuck in a loop of never-ending terminating/Deleting stage:

```
> $ kubectl get ns mynamespace
> NAME STATUS AGE mynamespace Terminating 1h
```

This can lead to useless resource consumption and unusability of the same name for a new namespace. One way almost guaranteed to forcefully remove a stuck namespace is as following:

Step 1: Dump the descriptor as JSON to a file

```
kubectl get namespace mynamespace -o json > mynamespace.json
```

2. Open the file where you saved it for editing

```
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "creationTimestamp": "2021-05-14T13:55:20Z",
    "labels": {
      "name": "mynamespace"
    },
    "name": "mynamespace",
    "resourceVersion": "29571918",
    "selfLink": "/api/v1/namespaces/mynamespace",
    "uid": "dsdsdqdd-764f-11e9-9621-dddasdasdasd"
  },
  "finalizers": [
```

[Skip to main content](#)

```

        "kubernetes"
    ]
},
"status": {
    "phase": "Terminating"
}
}

```

3. Remove kubernetes from the finalizers array:

```

> {
>   "apiVersion": "v1",
>   "kind": "Namespace",
>   "metadata": {
>     "creationTimestamp": "2021-05-14T13:55:20Z",
>     "labels": {
>       "name": "mynamespace"
>     },
>     "name": "mynamespace",
>     "resourceVersion": "29571918",
>     "selfLink": "/api/v1/namespaces/mynamespace",
>     "uid": "dsdsdqdd-764f-11e9-9621-dddasdasdasd"
>   },
>   "spec": {
>     "finalizers": [
>     ]
>   },
>   "status": {
>     "phase": "Terminating"
>   }
> }

```

NOTE: Finalizers allow controllers to implement asynchronous pre-delete hooks. Let's say you create an external resource (such as a storage bucket) for each object of your API type, and you want to delete the associated external resource on object's deletion from Kubernetes, you can use a finalizer to do that.

4. Executing our cleanup command

We can now instruct our cluster to get rid of that annoying namespace as we have the setup with finalizers complete via the "replace" command

```

> kubectl replace --raw "/api/v1/namespaces/mynamespace/finalize" -f
./mynamespace.json

```

[Skip to main content](#)

After running that command, the namespace should now be gone from your namespaces list, verify using

```
kubect1 get ns
```

The key thing to note here is the resource you are modify, in our case, it is namespaces, but it could be used forpods, deployments, services, etc.

#harnessLovesOurCommunity

Welcome to the
Harness Community!

Getting Started

[Quick Start Guide](#)[Quick Start](#)

Resources

[Documentation](#)[Videos](#)[Unscripted](#)

Support

[Troubleshooting](#)

Participate

[Community Slack](#)[Join Community Slack](#)[YouTube Channel](#)[ShipTalk Podcast](#)[Unscripted Conference](#)[Privacy](#) [About](#)