

Exploring Jenkins X Garbage Collection

This lesson will discuss CronJobs in Jenkins X which will execute periodically to remove unused resources used in the preview environment.

We'll cover the following

- Getting Kubernetes CronJobs
- Getting the installed previews
- Running the garbage collector

It would be silly for each preview environment to run forever. We need to figure out how to remove everything deployed as part of the pull request process for those that are already merged or closed. We could manually delete previews, but that would result in too much human involvement in something repetitive and predictable. We could create a script that would do that for us that would be executed periodically, but before we do that we should check whether Jenkins X already has some Kubernetes CronJobs.

Getting Kubernetes CronJobs

```
kubectl get cronjobs
```

The output is as follows.

NAME	SCHEDULE	SUSPEND	ACTIVE	LAST	SCHEDULE	AGE
jenkins-x-gcactivities	0/30 */3 * * *	False	0	1h		3h
jenkins-x-gcpods	0/30 */3 * * *	False	0	1h		3h
jenkins-x-gcpreviews	0 */3 * * *	False	0	2h		3h

At the time of this writing, there are three Jenkins X CronJobs. All three are garbage collectors (**gc**) that are executed periodically (every three hours). The names of the CronJobs should give you an idea of their purpose.

- **jenkins-x-gcactivities** : Removes old Jenkins X activities.
- **jenkins-x-gcpods** : Deals with completed or failed Pods.
- **jenkins-x-gcpreviews** : Eliminates preview environments associated with

- **jenkins-x-gcp-previews**: Eliminates preview environments associated with merged or closed pull requests.

Given that we have a CronJob that deletes completed pull requests, the most we would wait before they are removed from the system is three hours. If that's too long (or too short) of a wait, we can change the schedule of the CronJob to some other value. In some cases, we might not want to wait at all. For that, we have a **jx** command that runs the same process as the CronJob. But, before we run it, we'll confirm that the preview is indeed still available.

Getting the installed previews

```
jx get previews
```

The output is as follows.

PULL REQUEST	NAMESPACE	APPLICATION
https://github.com/vfarcic/go-demo-6/pull/33	jx-vfarcic-go-demo-6-pr-33	http://go-demo-6...nip.io

Running the garbage collector

If in your case the output is empty, it means that an iteration of the CronJob was executed since you merged the pull request and, as a result, the preview was removed from the cluster. If that's not the case and the preview is still available, we'll run the garbage collector manually.

```
jx gc previews
```

The output is as follows:

```
Deleting helm release: jx-vfarcic-go-demo-6-pr-34
Deleting preview environment: vfarcic-go-demo-6-pr-34
Deleted environment vfarcic-go-demo-6-pr-34
```

We can see from the output that both the Helm chart of the preview as well as the Namespace were removed. We can confirm that by listing the previews one more time.

```
jx get previews
```

This time the output should be empty.

In other cases, we might want to remove only a specific preview and leave the others intact. We could have used the `jx delete preview` command for that.



In most cases, there is no need for us to run Jenkins X garbage collectors manually. That's what the CronJobs are for. I showed you that it can be done, but that does not mean that it should. I'm sure you have more exciting things to do than to waste your time removing unused applications, given that the system already does that for us.

Before we proceed, we'll exit the `go-demo-6` directory.

```
cd ..
```



Now, let's wrap up this discussion and remove the used resources.