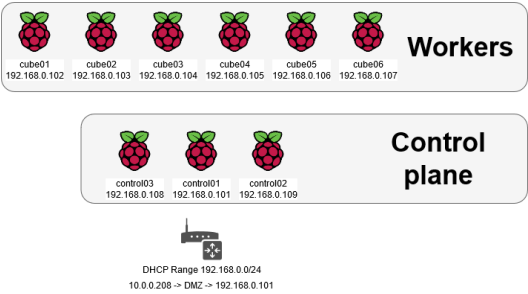


Design goals

More of this later on, but as far as I know there is no consensus ont how exactly a Kubernetes cluster should look, or what should be outside it, like load balancers or storage. However, if you are rocking official OpenShift from Red Hat, you have predefined pretty much everything. So take this whole setup as one of many versions. My goal is to contain everything in one cluster.

Goals

- Keep everything inside the cluster: no external components.
- Use 64bit OS – In our case, it's Ubuntu 20.10 ( You can't boot up easily 20.04 from USB, but 20.10 can be booted up out of the box ). I wanted arm64 because storage solutions for Kubernetes do not support arm32 (more on this later).
- Use distributed block storage for persistent storage for pods.
- End up with FaaS (Function as a Service) platform for python (and other functions).
- Failure of any node control or worker should not affect any running pods or whole cluster.



High Availability

I double in HA with this setup. It's not great, but at least it's something. Control plane contains 3 nodes which is the minimum number for a stacked cluster for the ETCD database. The issue with most guides I can find is that they set up one controller for Kubernetes cluster. This is fine, but when you restart the controller or lose it, the whole cluster goes down. Therefore, I have opted for 3 nodes that share the ETCD database between nodes (each keeps its own copy, synced). This allows one controller to fail and the cluster will run just fine. Same goes for worker nodes: they can fail and if there are still some up, pods can be rescheduled to run on them automatically.

Resource issues

Another thing usually omitted from discussion is the limitation of resources. You need to keep in mind that pods require disk space to exist. Yes, they exist temporarily on worker nodes, not tied to a specific server. Yet, they consume space: the more pods you are running more space you will need. Same shit for RAM: if you have pods that consume more RAM than worker nodes have, it might not run at all. Keep this in mind when designing your cluster. Make sure the workers have enough resources. Control servers can be as simple as possible as they do very little except keeping the workers in check.

🇵🇹 Liked it ? Buy me a drink :)

Last update: May 26, 2022

Comments

2 Comments

https://rpi4cluster.com

Disqus' Privacy Policy


Login

Favorite

Tweet

Share

Sort by Best




Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

**cmonty14** • a year ago

Hi,


why did you decide for 3 master nodes when K3S architecture is saying "Two or more server nodes that will serve the Kubernetes API and run other control plane services" [here](#).

Regards

Thomas

^ | v

• Reply • Share >

**viadoportos** Mod ➔ cmonty14 • a year ago

Well it says "two or more":D It will very well work with 2 ( or even one ) but I decided for 3 as that is what I would do in production. I do work with clusters in my day to day work ( mostly HP Service Guard and some RedHat HA ) and we have two node setups as standard ( which is not that great or even recommended and can cause split brain situations, but it save one server :D ) maybe its just my professional deformation trying to avoid problems with cluster solution that might be unrelated to this one ... hahahah

^ | v

• Reply • Share >

Subscribe

Add Disqus to your siteAdd DisqusAdd

Do Not Sell My Data

