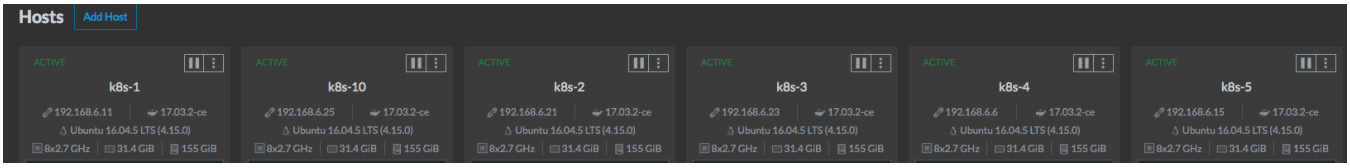
Kubernetes cluster stability

ONAP has grown in Casablanca and requires a lot of resources to run. Although I have a 6 node K8s cluster that I deployed using rancher, I believe the same kubelet flags can be applied to any method of spinning up and maintaining a K8s cluster (kubeadm etc). The nodes are Openstack VMs that I spun up on a private lab network.



The flavour I used:

VCPUs	RAM	Root Disk
8	32GB	160GB

That gives me 42 cores of CPU, 192 GB RAM, 960 GB storage with which to "try" and deploy all of ONAP.

Out of the box, what I found was that after a few days one of my nodes would become unresponsive and disconnect from the cluster. Even accessing the console through Openstack was slow. The resolution for me was a hard shutdown from the Openstack UI which brought it back to life (until it eventually happened again!).

It seems that the node that died was overloaded with pods that starved the core containers that maintain connectivity to the K8s cluster (kubelet, rancher-agent, etc).

The solution I found was to use kubelet eviction flags to protect the core containers from getting starved of resources.

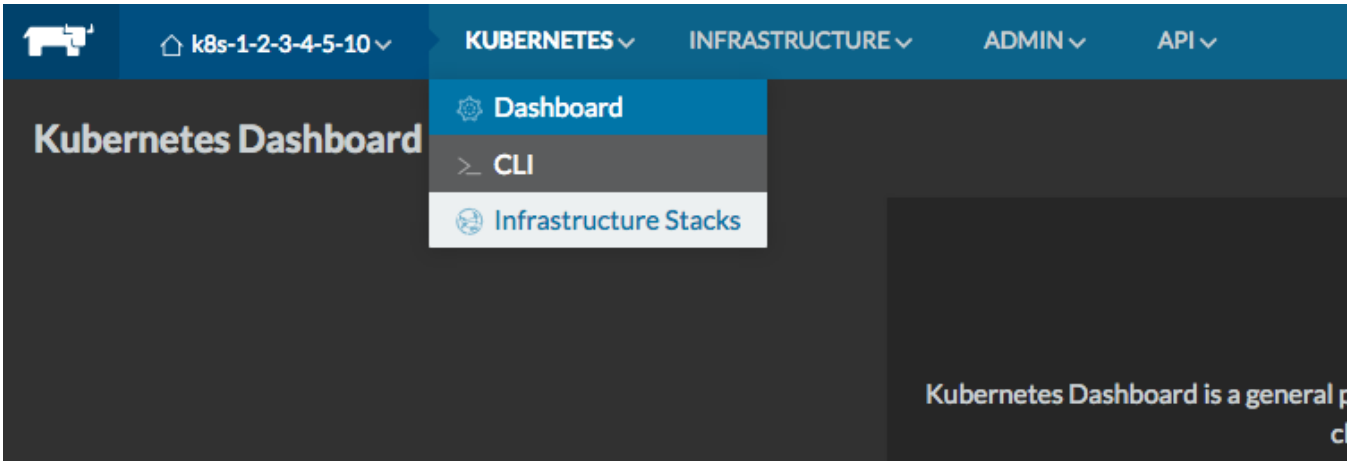
> These flags will not enable you to run all of ONAP on less resources. They are there to simply prevent the overloading of your K8s cluster from crashing. If you attempt to deploy too much the pods will be evicted by K8s.
>
> The resource limit story introduced in Casablanca add minimum requirements to each application. This is a WIP and will help the community better understand how many actual CPU cores and RAM is really required in your K8s cluster to run things.

For more information on the kubelet flags see:

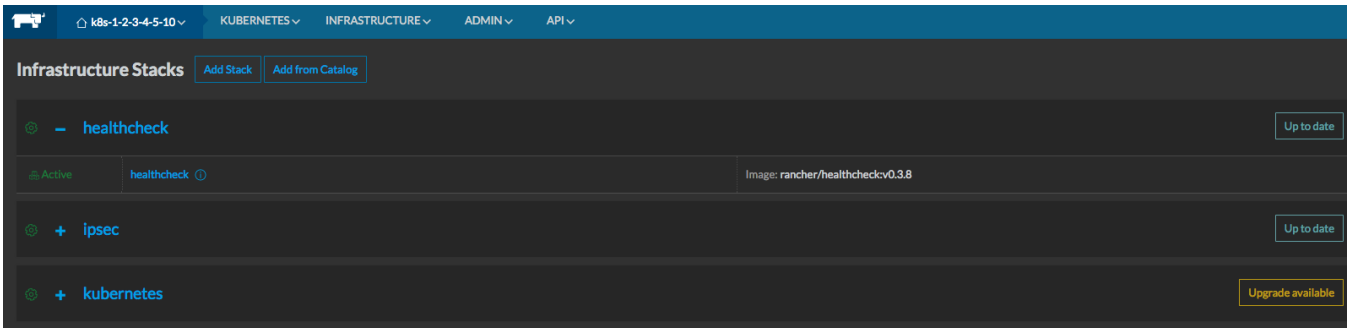
https://kubernetes.io/docs/tasks/administer-cluster/out-of-resource/

Here is how I applied them to my already running K8s cluster using rancher:

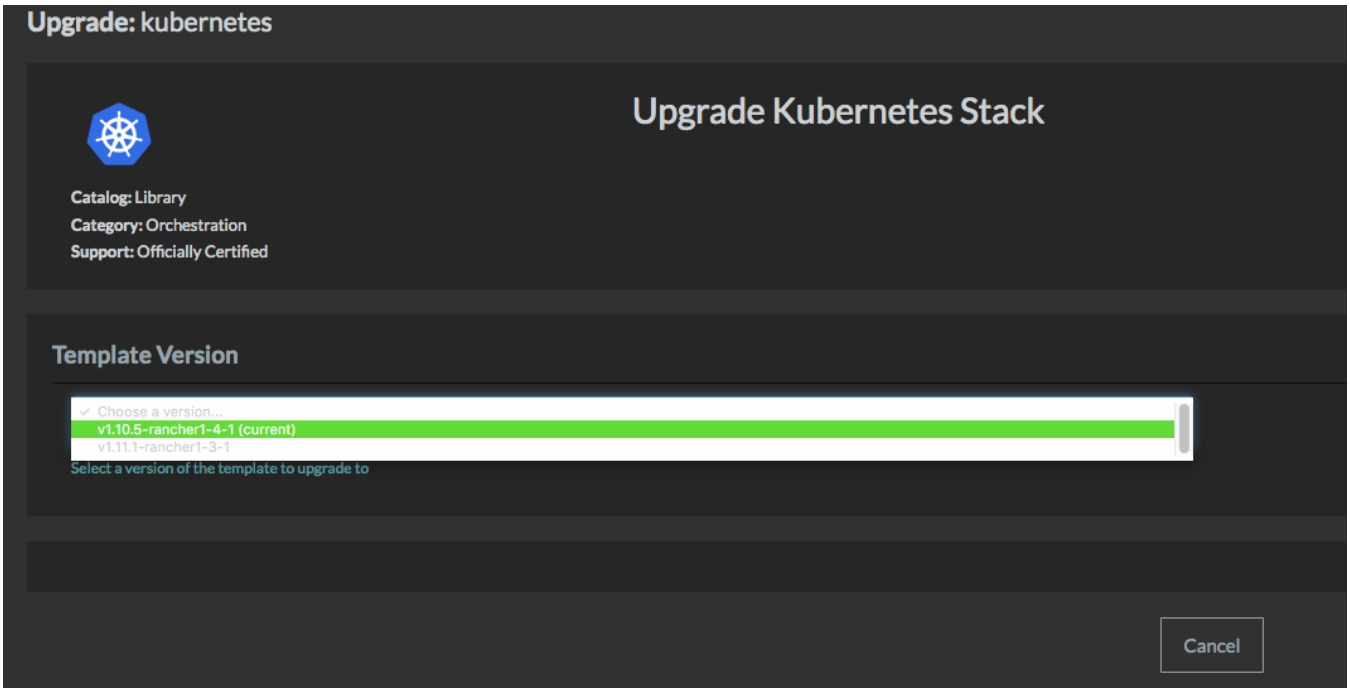
- Navigate to the Kubernetes > "Infrastructure Stacks" menu option



- Depending on the version of rancher and the K8s template version you have deployed, the button on the right side will say either "Up to date" or "Upgrade available".

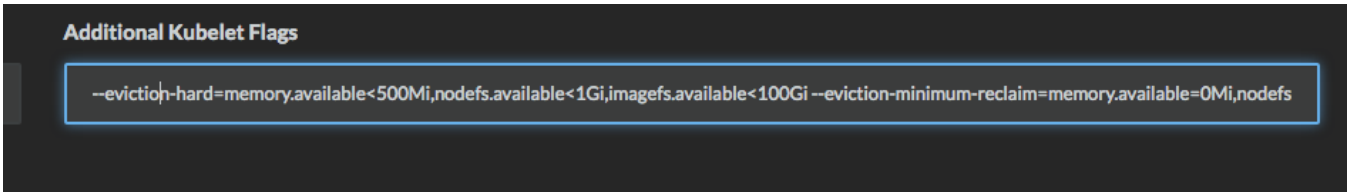


- Click the one beside the "+ kubernetes stack" and it will bring up a menu where you can select the environment template you want to tweak

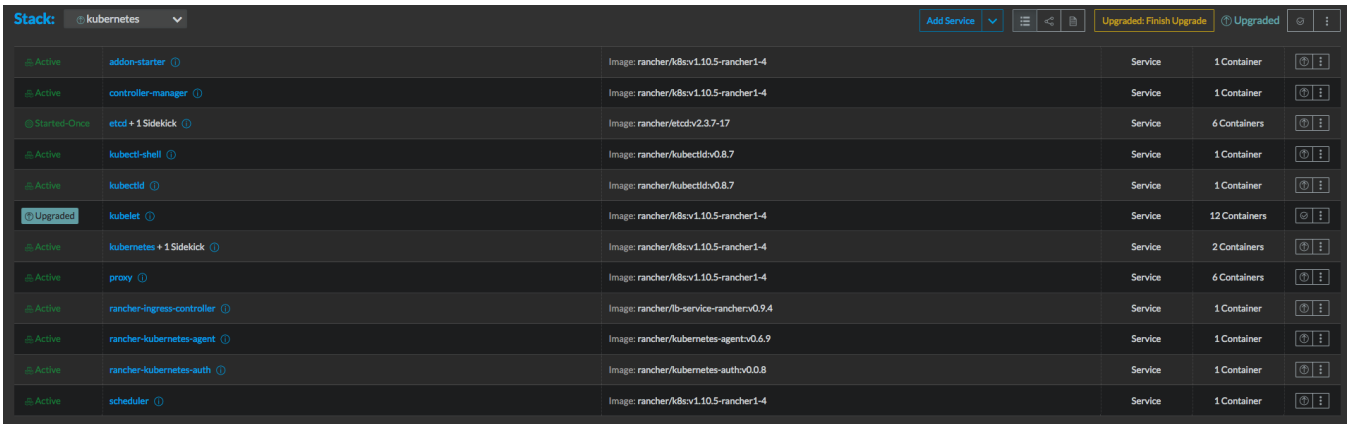


- Find the field for Additional Kubelet Flags and input the desired amount of resources you want to reserve. I used the example from the K8s docs but you should adjust to what makes sense for your environment.

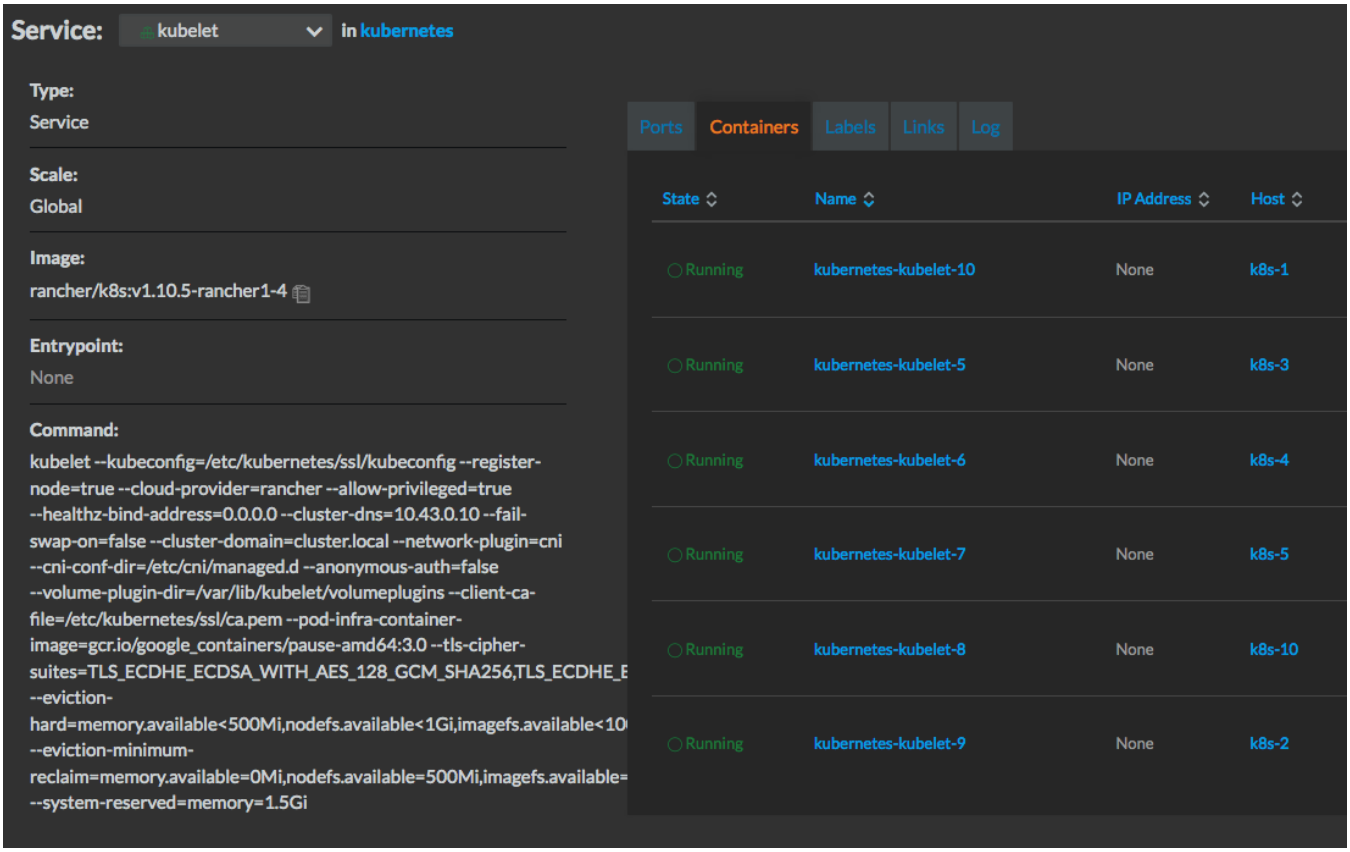
```
--eviction-hard=memory.available<500Mi,nodefs.available<1Gi,imagefs.available<5Gi

--eviction-minimum-reclaim=memory.available=0Mi,nodefs.available=500Mi,imagefs.available=2Gi
 --system-reserved=memory=1.5Gi
```



- The stack will upgrade and it will basically add some startup parameters to the kubelet container and bounce them. Click on "Upgraded: Finish Upgrade" to complete the upgrade.



- Validate the kubelet flags are now passed in as arguments to the container:



- That's it!

Hopefully this was helpful and will save people time and effort reviving dead K8s worker nodes due to resource starvation from containers that do not have any resource limits in place.