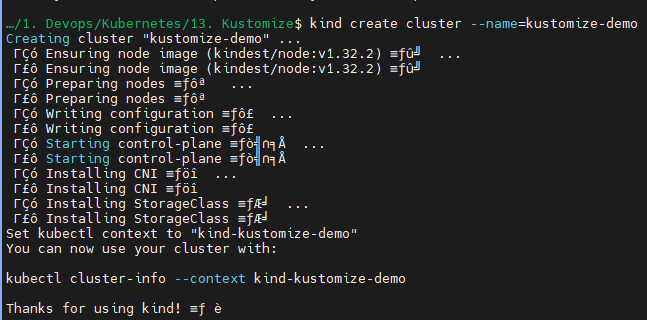
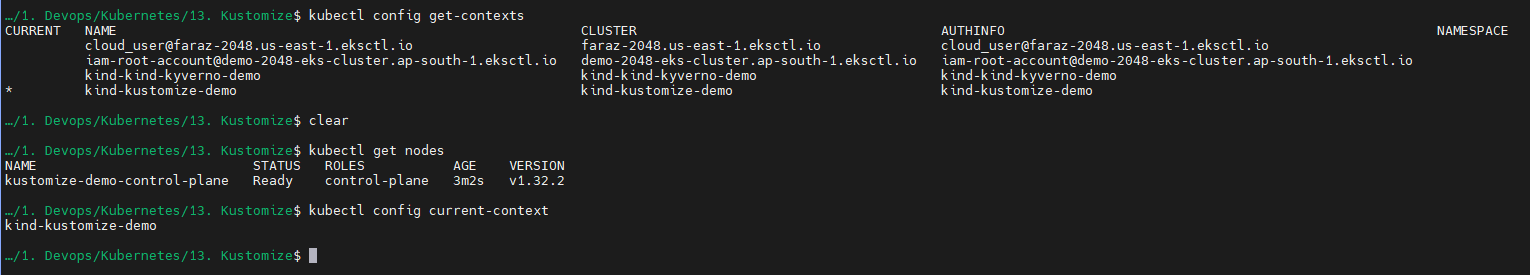
Create kind cluster if you want:



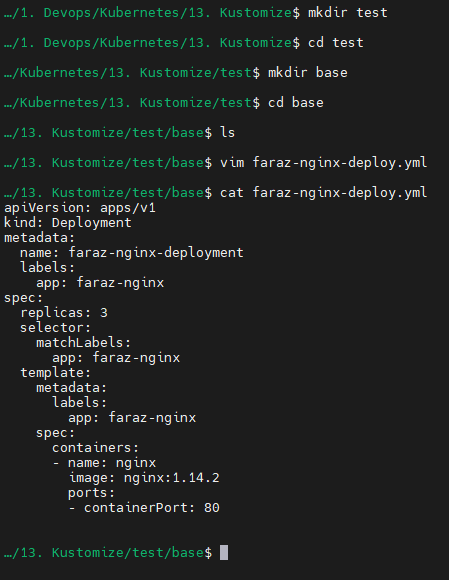
Check the current context, change if required:

* kubectl config get-contexts
* kubectl config current-context

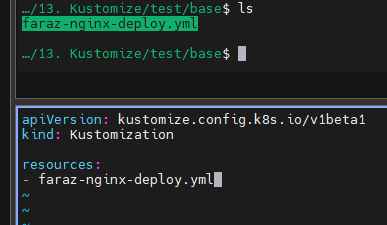


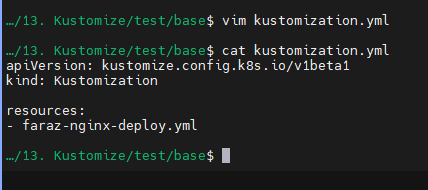
**Using inbuild transformers:**

Now create a deployment in test/base/



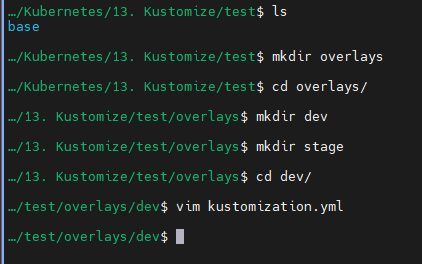
And create kustomization:



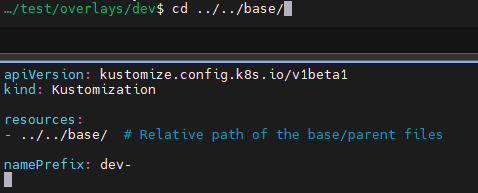


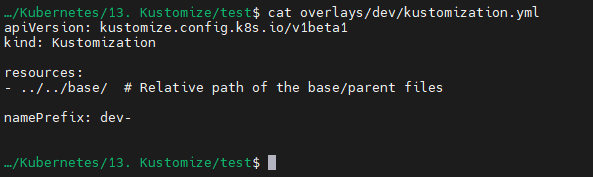
This kustomization file will keep a track of resources we want to keep a track of or modify

Now create overlays folder:



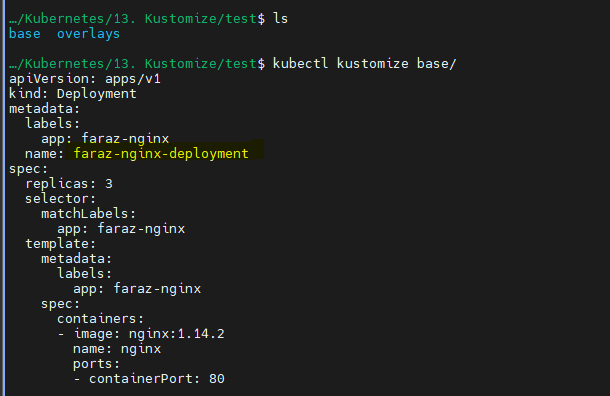
Create a kustomization file:



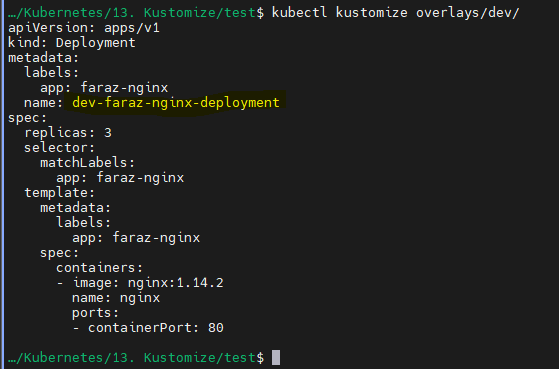


Now we kustomize:

* kubectl kustomize base/

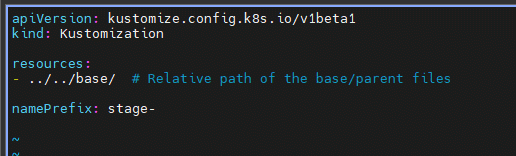


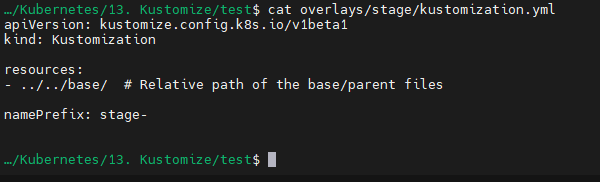
* kubectl kustomize overlays/dev/



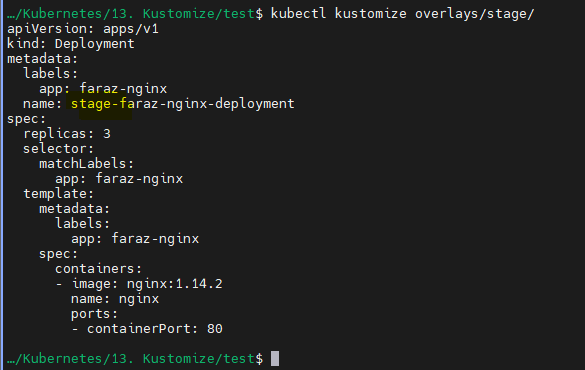
Similarly, we create for stage as well:





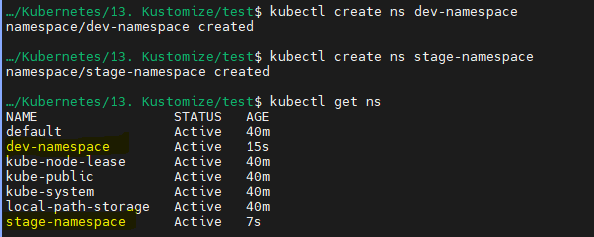


* kubectl kustomize overlays/stage/

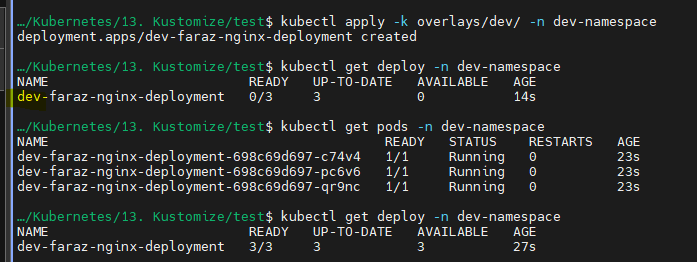


Now create namespaces:

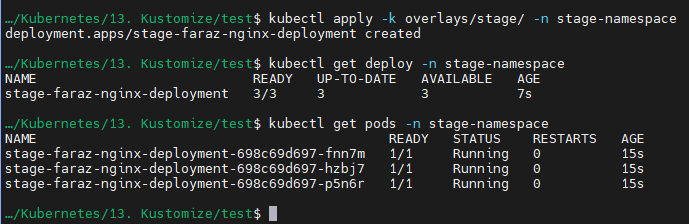
* kubectl create ns dev-namespace
* kubectl create ns stage-namespace



* kubectl apply -k overlays/dev/ -n dev-namespace

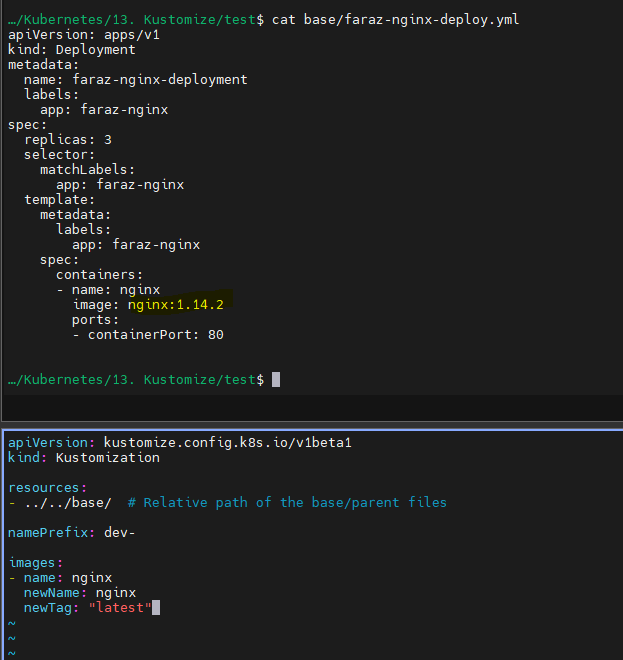


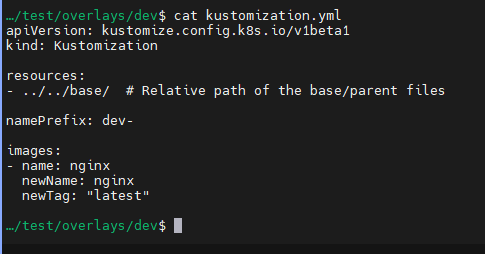
* kubectl apply -k overlays/stage/ -n stage-namespace



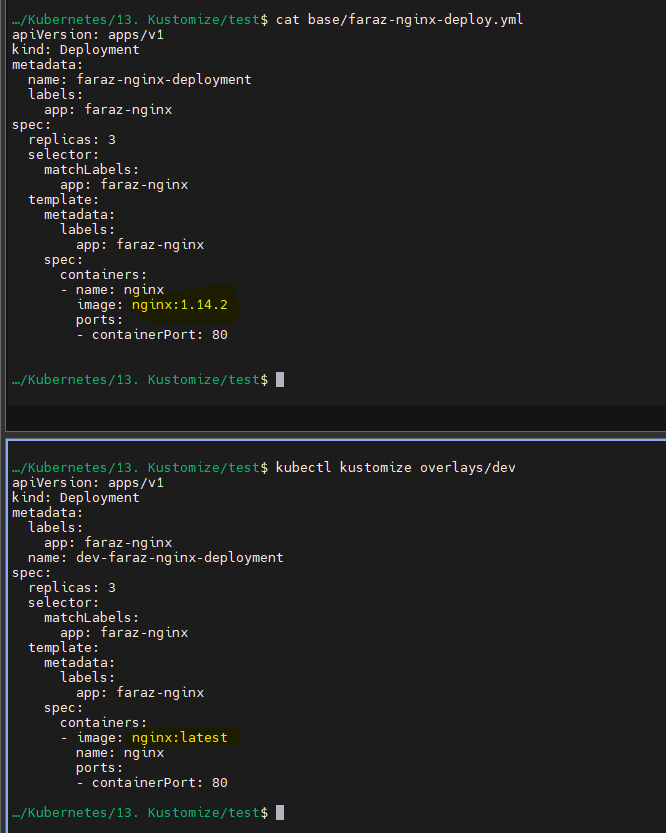
You can also kustomize the image tags:

So instead of 1.14.2 I want to use latest:



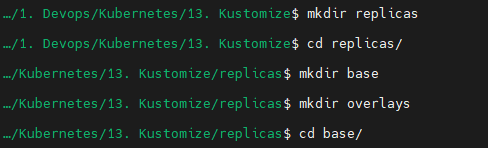


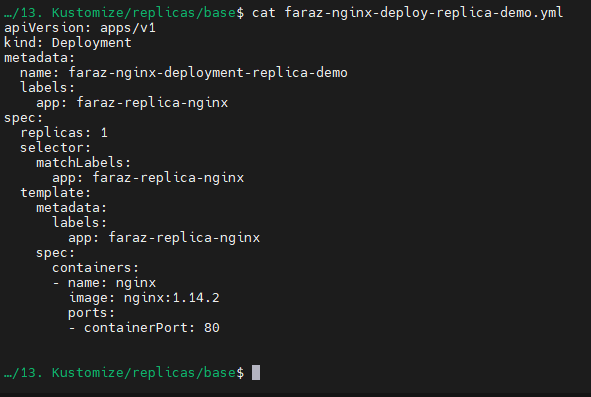
* kubectl kustomize overlays/dev



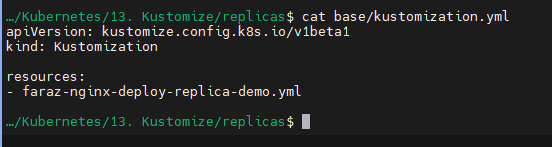
**You can also use your own patches (underlying use of transformers is same but gives more flexibility):**

2.1. We can also perform demo for replicas:



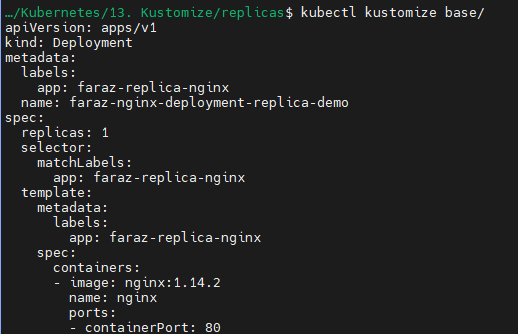


Also create base kustomization file:



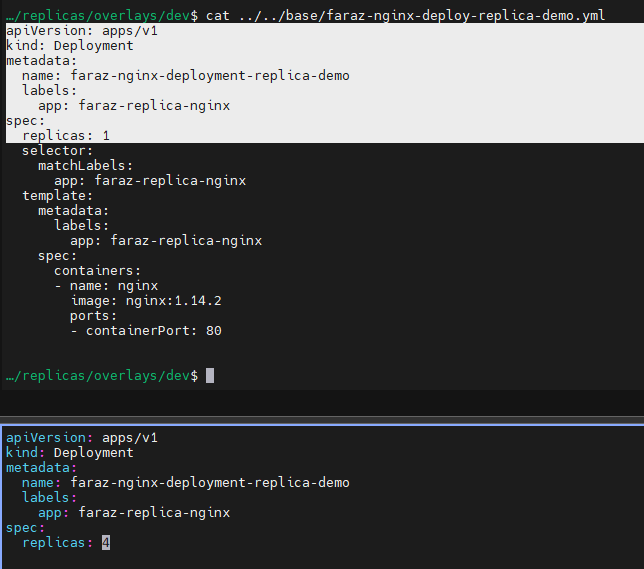
Test it by:

* kubectl kustomize base/

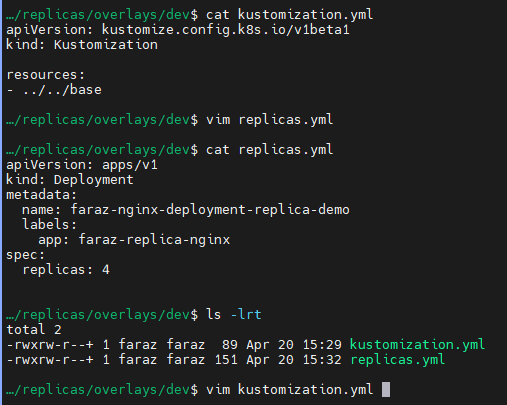


If this is successful, it means that the kubectl and kustomize were able to load the files and these files can be kustomized by anyone by overlays

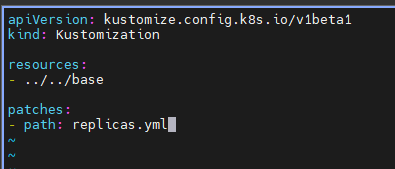
Create a patch:



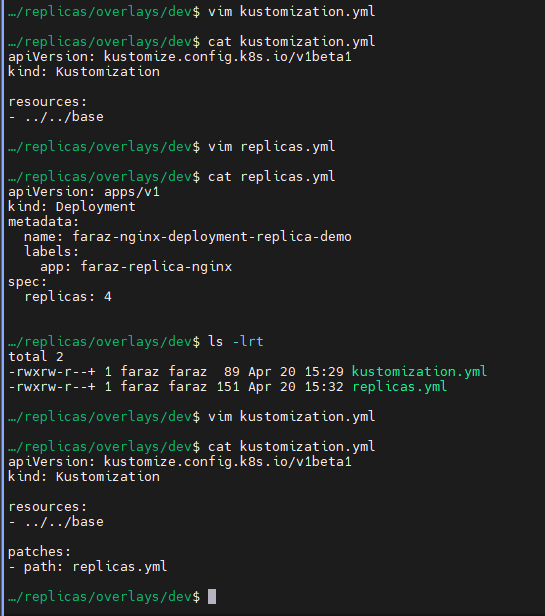
Edit the kustomization file:



Add the patches:

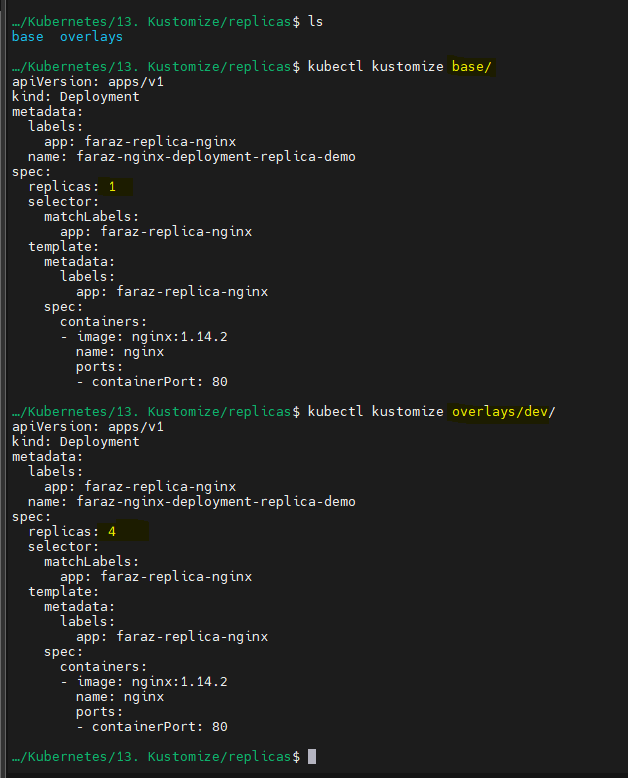


* Your kustomization file will look like this on editing:



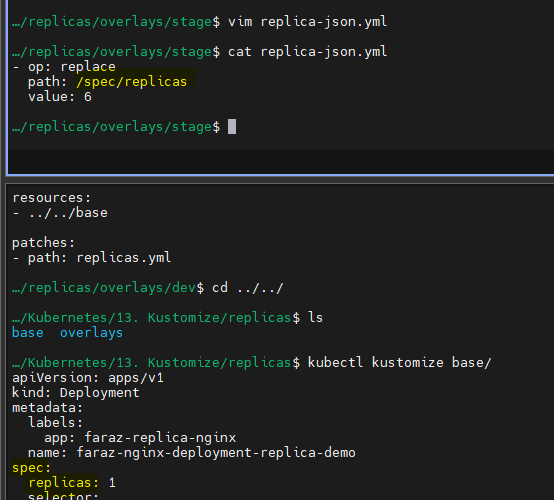
Now if you:

* kubectl kustomize base/
* kubectl kustomize overlays/dev/

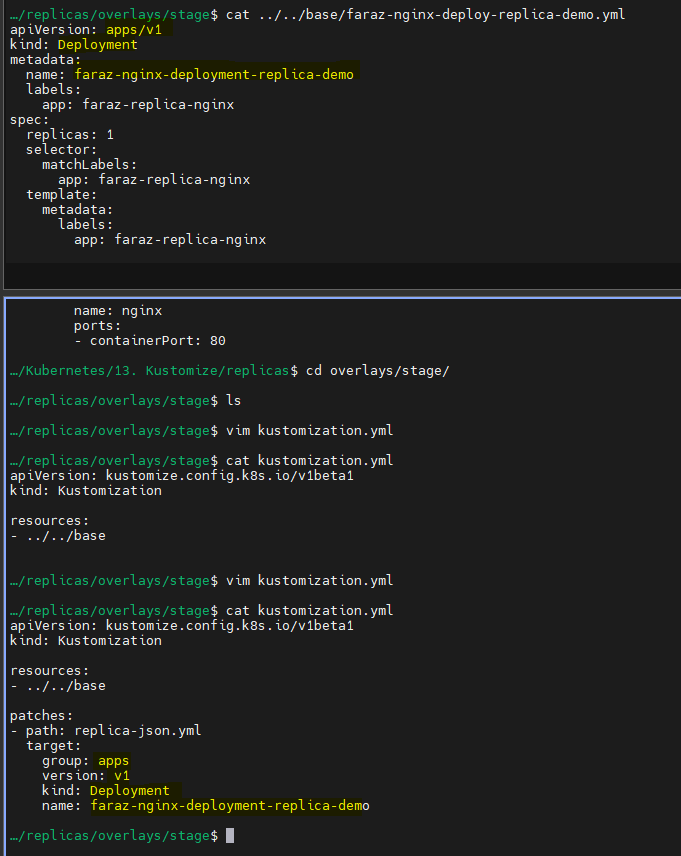


Here you will be able to compare the kustomization which we want to achieve using overlays to base.

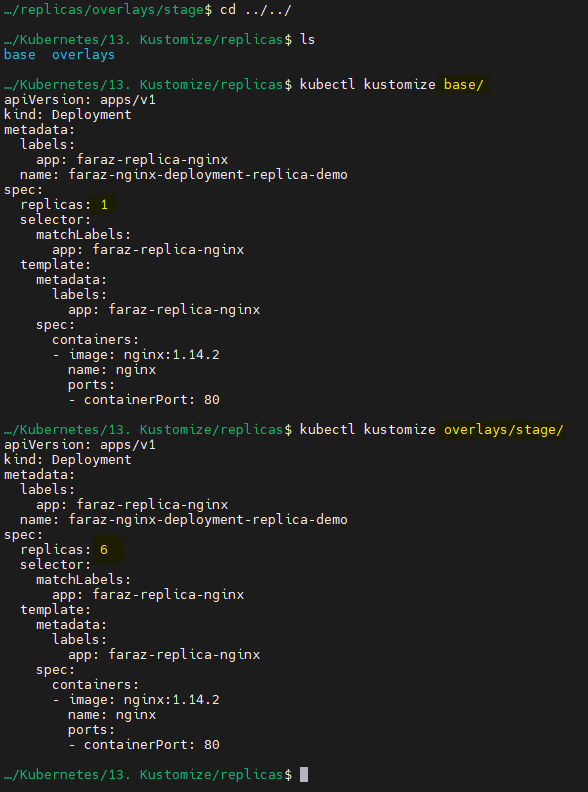
2.2. You can also use JSON yml approach:



Also add the following patches and target:

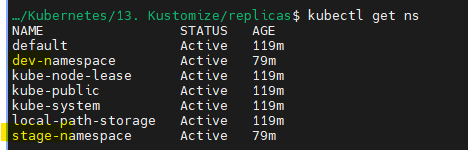


You can verify:

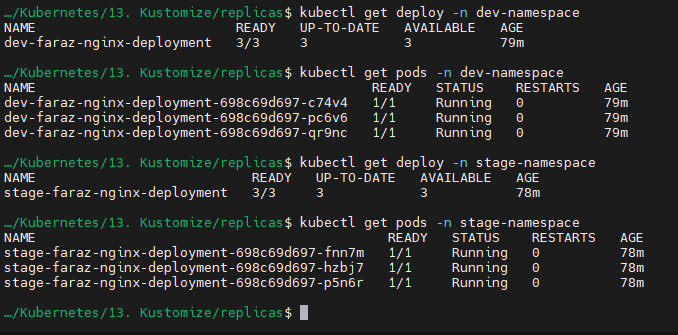


Now test it by running the kustomize:

Namespaces:

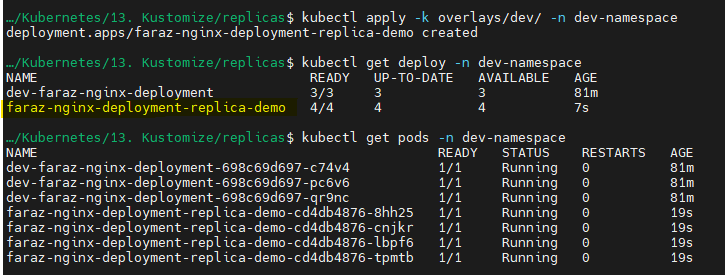


Currently we have:

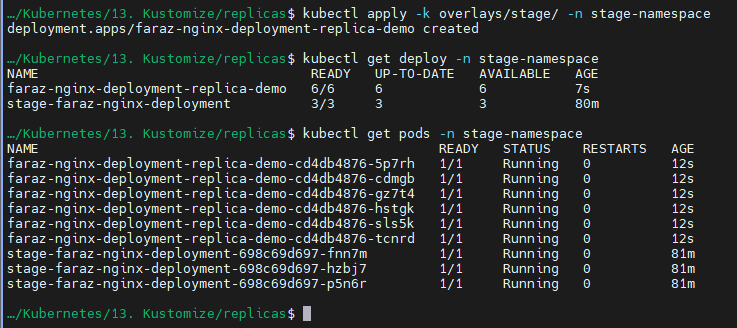


Now on applying:

* kubectl apply -k overlays/dev/ -n dev-namespace



* kubectl apply -k overlays/stage/ -n stage-namespace



For more examples:

<https://github.com/iam-veeramalla/kustomize-zero-to-hero/tree/main>