One of the most secure ways to run a Kubernetes cluster is to avoid using images that have not been vetted for use in your infrastructure by you personally or by trusted members of your team. The best way to do this is by using a private Docker repository. In this way, you know that the image you're using is the same one you pushed.

You're going to need some information about the repo in order to configure your cluster to access it. Start by logging in to your repo using

docker login

which will create (or update) a config.json file in your ~/.docker directory. Have a look at it and note the authentication token for your new repo. You'll need it to create a secret that Kubernetes can use to grab your image from your private repo. Here's example syntax for creating that secret:

kubectl create secret docker-registry regsecret --docker-server=<your-registry-server> --docker-username=<your-name> --docker-password=<your-pword> --docker-email=<your-email>

Next, when you create your yaml for your pod or deployment or whatever, you'll specify the image like in this sample below:

```
apiVersion: v1
kind: Pod
metadata:
  name: private-reg
spec:
  containers:
  - name: private-reg-container
    image: <your-private-image>
  imagePullSecrets:
  - name: regsecret
```

Ref:
https://unofficial-kubernetes.readthedocs.io/en/latest/tasks/configure-pod-container/pull-image-private-registry/

https://linuxacademy.com/howtoguides/posts/show/topic/24043-pulling-images-from-a-private-repository-into-kubernetes