

Solutions

1. `helm install --name my-release stable/jenkins`
2. `kubectl get svc my-release-jenkins -o=jsonpath='{.spec.ports[?(@.port==8080)].nodePort}'`
3. `printf $(kubectl get secret --namespace default my-release-jenkins -o jsonpath="{.data.jenkins-admin-password}" | base64 --decode);echo`
- 4.

```
def label = "mypod-${UUID.randomUUID().toString()}"
podTemplate(label: label, containers: [
    containerTemplate(name: 'ubuntu', image: 'ubuntu', ttyEnabled: true, command: 'cat'),
]) {

    node(label) {
        stage('Disks Usage') {
            sh "df -h"
        }
    }
}
```

```
Started by user admin
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] podTemplate
[Pipeline] {
[Pipeline] node
Still waiting to schedule task
'jenkins-slave-jjndb-k3qzf' is offline
Agent jenkins-slave-jjndb-k3qzf is provisioned from template Kubernetes Pod Template
Agent specification [Kubernetes Pod Template] (mypod-04b416c7-19a7-4700-b933-6c0ba2fafced):
* [ubuntu] ubuntu
```

```
Running on jenkins-slave-jjndb-k3qzf in /home/jenkins/workspace/my-pipeline
```

```
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Disks Usage)
[Pipeline] sh
+ df -h
Filesystem              Size      Used Available Use% Mounted on
overlay                  17.0G      6.9G       9.0G  43% /
tmpfs                    64.0M        0      64.0M   0% /dev
tmpfs                     1.9G        0       1.9G   0% /sys/fs/cgroup
/dev/sda1                17.0G      6.9G       9.0G  43% /home/jenkins
```

5. Jenkins fired up a jnlp pod with two containers inside, one for the agent and another one for the ubuntu container that we used to run the `df -h` command.