*This guide explains the step by step process for setting up Jenkins on a*[*Kubernetes cluster.*](https://devopscube.com/setup-kubernetes-cluster-google-cloud/)

**Setup Jenkins On Kubernetes Cluster**

For setting up a [Jenkins](https://devopscube.com/jenkins-2-tutorials-getting-started-guide/) cluster on Kubernetes, we will do the following.

1. Create a Namespace
2. Create a deployment yaml and deploy it.
3. Create a service yaml and deploy it.
4. Access the Jenkins application on a Node Port.

**Note:** This tutorial doesn’t use persistent volume as this is a generic guide. For using persistent volume for your Jenkins data, you need to create volumes of relevant cloud or on-prem data center and configure it.

**Create a Jenkins Deployment**

1. Create a Namespace for Jenkins. So that we will have an isolation for the CI/CD environment.



|  |  |
| --- | --- |
| 1 | kubectl create ns jenkins |

2. Create a Deployment file named jenkins-deployment.yaml the latest Jenkins Docker Image.

**Note**: The following deployment file doesn’t add any persistent volume for jenkins. For production use cases, you should add a persistent volume for your jenkins data. A sample implementation of persistent volume for Jenkins in Google Kubernetes Engine can be [found here](https://devopscube.com/persistent-volume-google-kubernetes-engine/)



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | apiVersion: extensions/v1beta1 # for versions before 1.7.0 use apps/v1beta1  kind: Deployment  metadata:    name: jenkins-deployment  spec:    replicas: 1    selector:      matchLabels:        app: jenkins    template:      metadata:        labels:          app: jenkins      spec:        containers:        - name: jenkins          image: jenkins:2.60.3          ports:          - containerPort: 8080 |

3. Create the jenkins deployment in jenkins namespace using the following command.



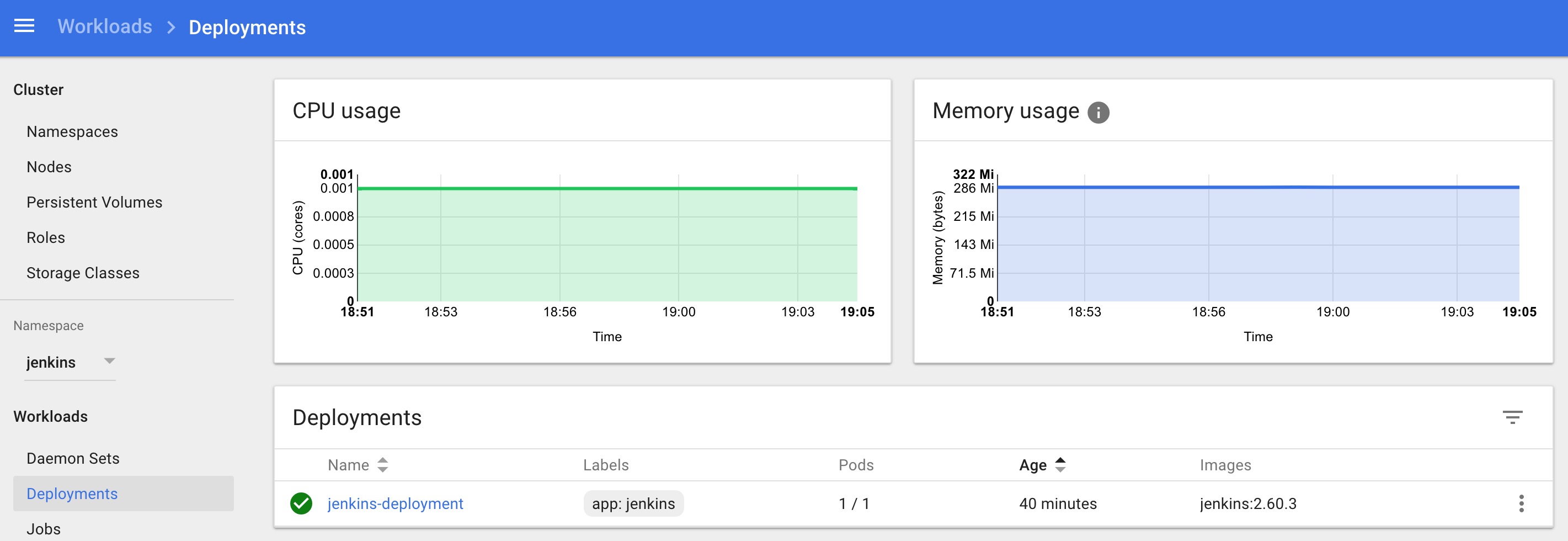
|  |  |
| --- | --- |
| 1 | kubectl create -f jenkins-deployment.yaml --namespace=jenkins |

4. Now, you can get the deployment details using the following command.



|  |  |
| --- | --- |
| 1 | kubectl  describe deployments --namespace=jenkins |

Also, You can get the details from the kubernetes dashboard as shown below.

[](https://devopscube.com/wp-content/uploads/2017/10/Screen-Shot-2017-10-08-at-7.05.25-PM.jpg)

**Create a Jenkins Service**

We have created a deployment. However is not accessible to the outside world. For accessing the Jenkins container from outside world, we should create a service and map it to the deployment.

1. Create a jenkins-service.yaml file with the following contents.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | apiVersion: v1  kind: Service  metadata:    name: jenkins  spec:    type: NodePort    ports:      - port: 8080        targetPort: 8080        nodePort: 30000    selector:      app: jenkins |

**Note:** In this, we are using the type as NodePort which will expose Jenkins on all kubernetes node IP’s. Also, we have mentioned the nodeport as 30000. So you can access the application on port 30000. If you are on Google Cloud or AWS, you can use the type as Loadbalancer which will launch create a Load balancer and points to the jenkins deployment.

[irp posts=”397″ name=”List of DevOps Blogs and Resources for Learning”]

2. Create the jenkins service using the following command.



|  |  |
| --- | --- |
| 1 | kubectl create -f jenkins-service.yaml --namespace=jenkins |

Now if you browse to any one of the Node IP on port 30000, you will be able to access the Jenkins dashboard.



|  |  |
| --- | --- |
| 1 | http://<node-ip>:3000 |

3. Jenkins will ask for initial Admin password. You can get that from the pod logs either from kubernetes dashboard or  CLI. You can get the pod details using the following CLI command.



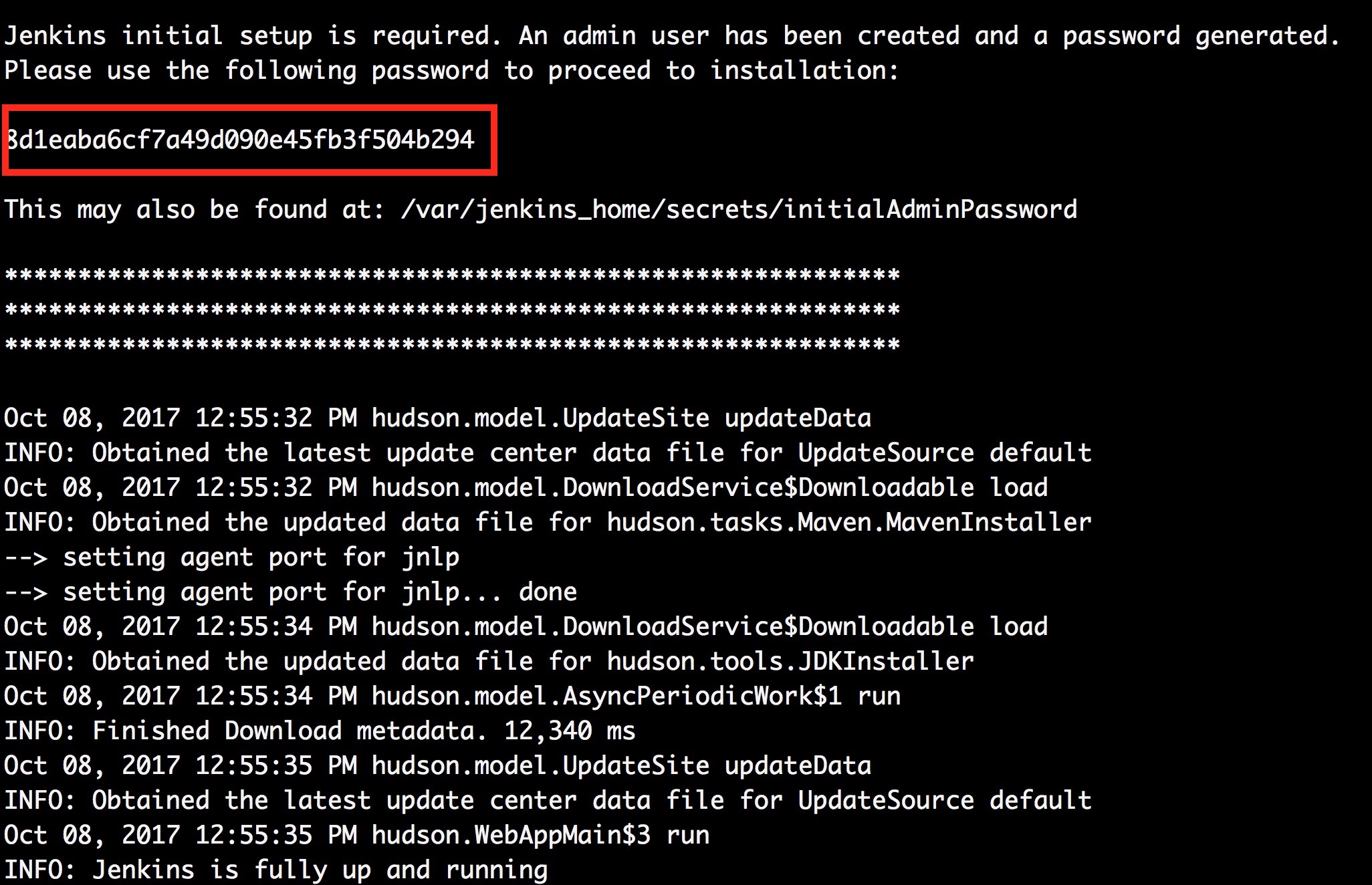
|  |  |
| --- | --- |
| 1 | kubectl get pods --namespace=jenkins |

And with the pod name, you can get the logs as shown below. replace the pod name with your pod name.



|  |  |
| --- | --- |
| 1 | kubectl logs jenkins-deployment-2539456353-j00w5 --namespace=jenkins |

The password can be found at the end of the log as shown below.

[](https://devopscube.com/wp-content/uploads/2017/10/Screen-Shot-2017-10-08-at-8.00.16-PM.jpg)

[](https://devopscube.com/wp-content/uploads/2017/09/Kubernetes2.png)

**ONLINE COURSE:**The Complete Kubernetes Course

Learn how you can run, deploy, manage and maintain containerized Docker applications on Kubernetes

[The Complete Kubernetes Course](https://devopscube.com/recommends/kubernetes-course/)

* Learn to launch kubernetes cluster
* Get started with Containerization of apps
* Deploy applications on kubernetes cluster
* Run stateful and stateless applications on containers

**You'll also like:**

* [How to Install and configure Sonarqube on Linux (RHEL/Centos/ec2)](https://devopscube.com/setup-and-configure-sonarqube-on-linux/)
* [How To Run/Access Jenkins on Port 80 in Linux](https://devopscube.com/access-run-jenkins-port-80/)
* [Setup NFS Server On Google Cloud – Managed Cloud FileStore Service](https://devopscube.com/nfs-servers-google-cloud-filestore/)
* [Understanding Continuous Integration, Delivery and Deployment](https://devopscube.com/continuous-integration-delivery-deployment/)
* [How to Setup and Configure Docker Swarm Cluster](https://devopscube.com/how-to-setup-and-configure-docker-swarm-cluster/)