**Kubernetes Web UI**

**Recommended Approach**

* If this is a test cluster and you don’t have strict separation of master and worker roles, deploy it on the **master node** for simplicity.
* For production setups, deploy it on a **worker node** to maintain best practices.

1. **Steps to Deploy the Dashboard**

kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.7.0/aio/deploy/recommended.yaml

1. **Expose the Dashboard**

To access the dashboard, expose it via a **proxy** or **NodePort**:

**Expose as a NodePort (For External Access)** Edit the dashboard service to change its type to NodePort:

kubectl -n kubernetes-dashboard edit svc kubernetes-dashboard

Change:

type: ClusterIP

To:

type: NodePort

Save and check the NodePort:

kubectl -n kubernetes-dashboard get svc kubernetes-dashboard

Access it via <Node-IP>:<Node-port. Example :- 192.168.100.85:31911

**3) Create a Service Account with Admin Access**

To log into the dashboard:

1. Create a service account:

kubectl create serviceaccount dashboard-admin -n kubernetes-dashboard

kubectl create clusterrolebinding dashboard-admin \

--clusterrole=cluster-admin \

--serviceaccount=kubernetes-dashboard:dashboard-admin

1. Get the login token:

kubectl -n kubernetes-dashboard describe secret $(kubectl -n kubernetes-dashboard get secret | grep dashboard-admin | awk '{print $1}')

Copy the token and use it to log into the dashboard

.

**4) If Automatically Token is not generated so we have to manually created**

**1. Create the Secret with Proper Annotations**

Use the following command to create a properly annotated secret for the dashboard-admin service account:

kubectl -n kubernetes-dashboard create secret generic dashboard-admin-token \

--type=kubernetes.io/service-account-token \

--from-literal=token=$(openssl rand -base64 21) \

--dry-run=client -o yaml > dashboard-admin-token.yaml

This command generates the secret YAML but does not apply it yet.

**2. Edit the YAML**

Open the dashboard-admin-token.yaml file with a text editor, such as nano:

nano dashboard-admin-token.yaml

Add the kubernetes.io/service-account.name annotation under metadata.annotations, so the YAML looks like this:

apiVersion: v1

kind: Secret

metadata:

name: dashboard-admin-token

namespace: kubernetes-dashboard

annotations:

kubernetes.io/service-account.name: dashboard-admin

type: kubernetes.io/service-account-token

data:

token: <REPLACE\_WITH\_A\_BASE64\_ENCODED\_STRING>

**Replace <REPLACE\_WITH\_A\_BASE64\_ENCODED\_STRING>** with a Base64-encoded string. For simplicity, you can leave it empty, and Kubernetes will generate the token automatically.

**3. Apply the Secret**

Apply the edited YAML:

kubectl apply -f dashboard-admin-token.yaml

**4. Verify the Secret**

Check if the secret is created:

kubectl -n kubernetes-dashboard get secret

Describe the secret to retrieve the token:

kubectl -n kubernetes-dashboard describe secret dashboard-admin-token



