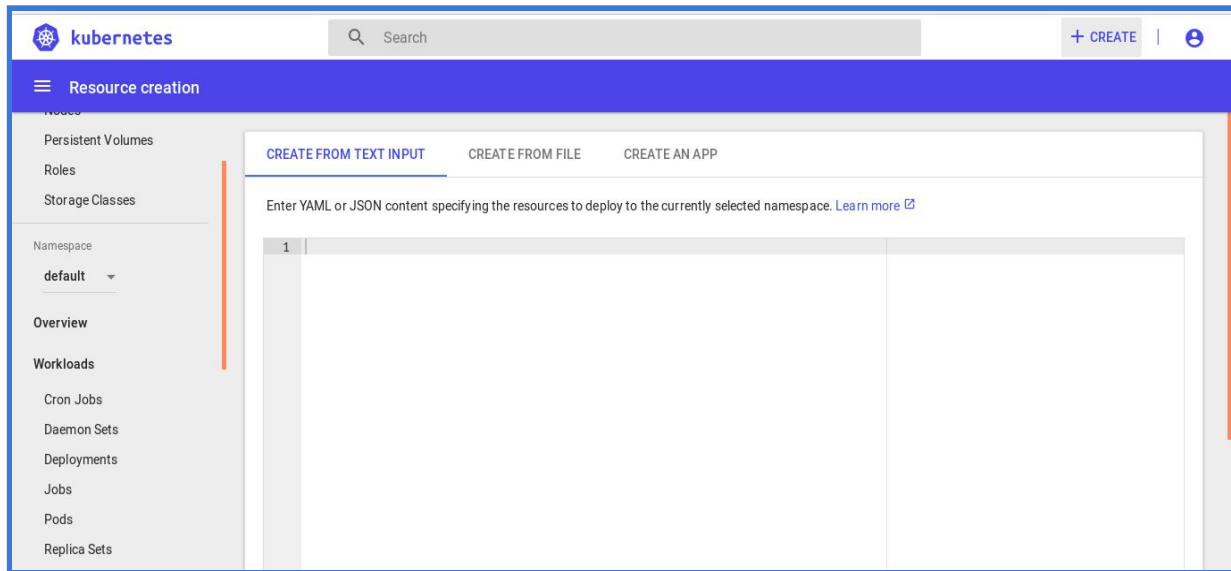


Run a Stateless Application Using a Deployment



Login to the Kubernetes Dashboard from the Link provided in K8s Lab instructions
The default username is admin and the password is provided in K8s Lab instructions

Refer the K8s Lab instructions for commands to generate the TOKEN.

1. Deploy the nginx container by clicking `CREATE` on the top right.

2. Select the third option "CREATE AN APP"

The screenshot shows the 'CREATE AN APP' tab in the Kubernetes Resource creation interface. The form includes the following fields and options:

- App name ***: A text input field with a character count of 0 / 24. A tooltip explains that this value will be added to the Deployment and Service labels.
- Container image ***: A text input field. A tooltip instructs the user to enter the URL of a public image on a registry or a private image hosted on Docker Hub or Google Container Registry.
- Number of pods ***: A numeric input field set to 1. A tooltip states that a Deployment will be created to maintain the desired number of pods across the cluster.
- Service ***: A dropdown menu currently set to 'None'. A tooltip explains that an internal or external Service can be defined to map an incoming Port to a target Port seen by the container.

At the bottom of the form, there are buttons for 'DEPLOY' and 'CANCEL', and a link to 'SHOW ADVANCED OPTIONS'.

3. Enter the values as follows

App name : <Your-name> (name the app with your name for identification later)

Container name : **nginx** (is the name of the Docker container that will be pulled from the official Docker repository)

Number of Pods : **1** (is the count of the container that you wish to deploy when the app is deployed)

Service: **external** (if the app needs to be accessed from outside the network then this is set to external and the K8s engine will bind the application to an external endpoint so that it can be accessed from outside the container network)

Port: **80**

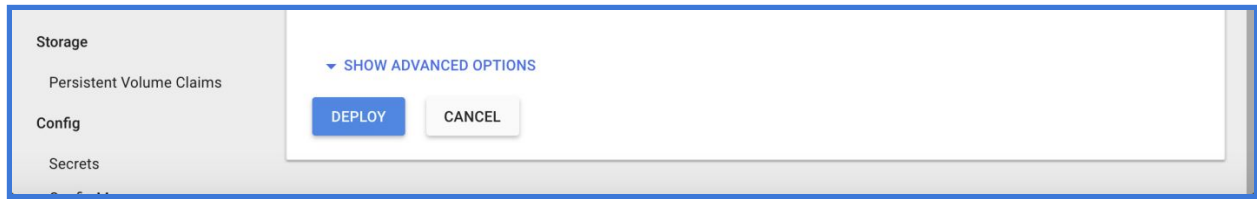
Target Port: **80** (port mapping of the PODS {container} and the host)

This screenshot shows the 'CREATE AN APP' form with the following values entered:

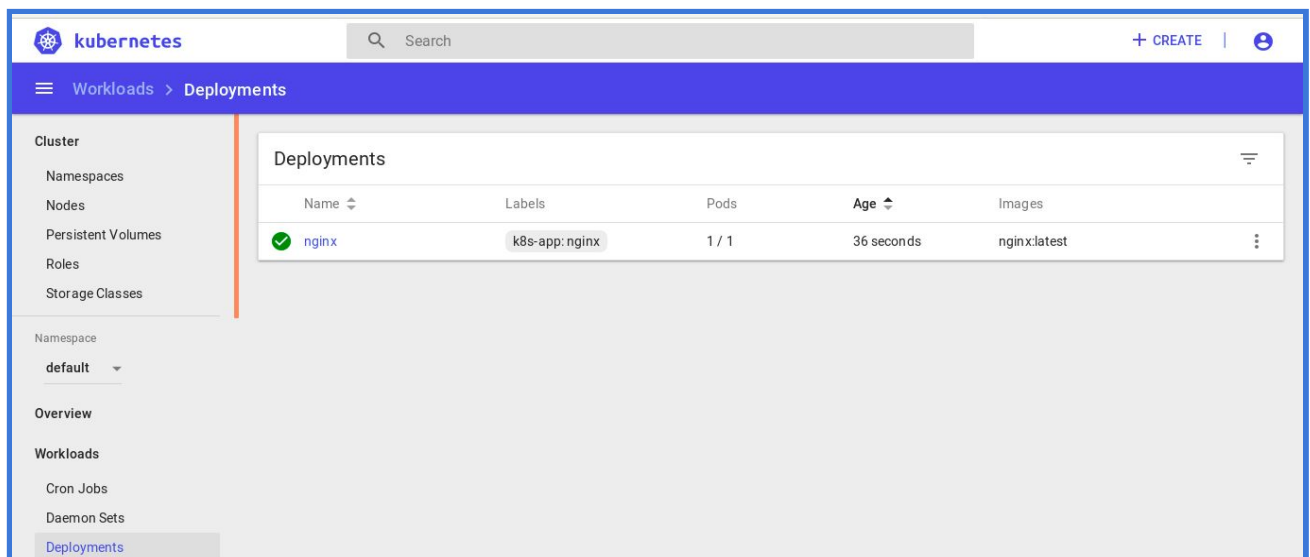
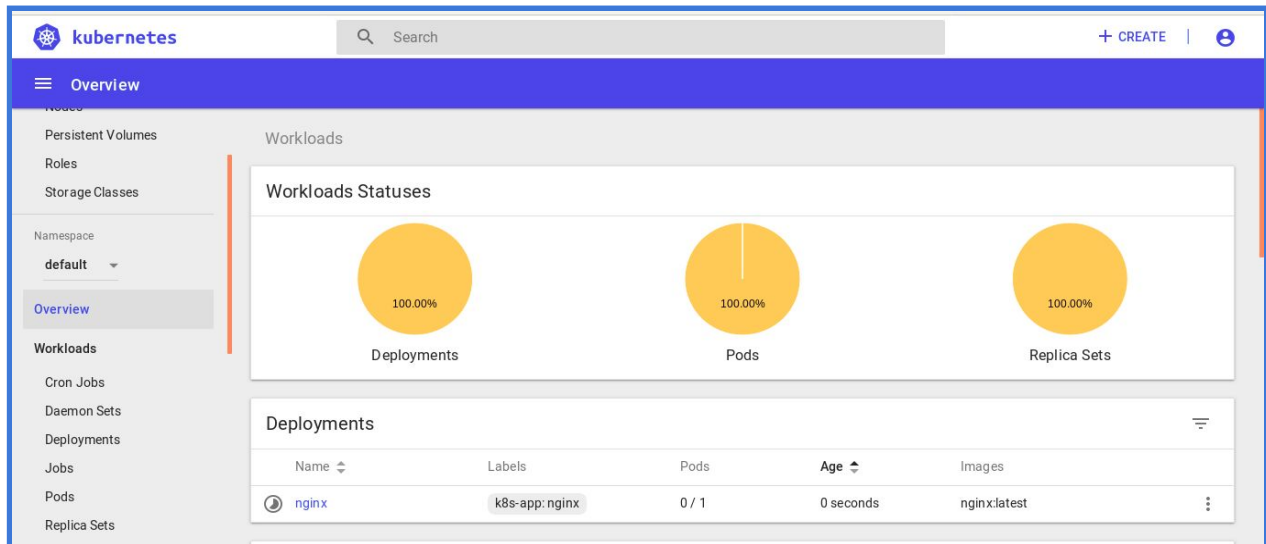
- App name ***: **nginx** (character count: 5 / 24)
- Container image ***: **nginx:latest**
- Number of pods ***: **1**
- Service ***: **External**
- Port ***: **80**
- Target port ***: **80**
- Protocol ***: **TCP**

The 'DEPLOY' button is now visible at the bottom of the form.

4. Click on “**Deploy**” once all the fields are filled.



5. Click on the **Deployments** tab on the left hand side to Review the deployed nginx application details. Your Deployment will be **your-name** as described during step 3.



The screenshot shows the Kubernetes Dashboard Overview page. The left sidebar contains navigation links: Overview, Pods, Replica Sets, Replication Controllers, Stateful Sets, Discovery and Load Balancing (Ingresses, Services), Config and Storage (Config Maps, Persistent Volume Claims, Secrets), Settings, and About. The main content area displays three green circular progress indicators for Deployments, Pods, and Replica Sets, all at 100.00%. Below these, the 'Deployments' table shows one deployment named 'nginx' with labels 'k8s-app: nginx', 1 pod, and an age of 7 minutes. The 'Pods' table shows one pod named 'nginx-67d8778dd9-vx5h5' on node 'ip-172-20-61-116.us-east-2.compute.internal' in a 'Running' state, with 0 restarts and an age of 4 minutes.

| Name | Labels | Pods | Age | Images |
|-------|----------------|-------|-----------|--------------|
| nginx | k8s-app: nginx | 1 / 1 | 7 minutes | nginx:latest |

| Name | Node | Status | Restarts | Age |
|------------------------|---------------------------------------------|---------|----------|-----------|
| nginx-67d8778dd9-vx5h5 | ip-172-20-61-116.us-east-2.compute.internal | Running | 0 | 4 minutes |

6. Click on the Services tab on the left hand and click on the nginx Services to find the external endpoints. The Service will be **your-name** as described during step 3.

The screenshot shows the Kubernetes Dashboard Services page. The left sidebar is updated with 'Services' selected under 'Discovery and Load Balancing'. The main content area displays the 'Services' table with two entries: 'nginx' and 'kubernetes'. The 'nginx' service has labels 'k8s-app: nginx', Cluster IP '100.70.189.16', internal endpoints 'nginx:80 TCP' and 'nginx:32007 TCP', and an external endpoint 'ad05abac1d6a111e8a'. The 'kubernetes' service has labels 'component: apiserver' and 'provider: kubernetes', Cluster IP '100.64.0.1', and internal endpoint 'kubernetes:443 TCP'. Below the Services table, the 'Secrets' table shows one secret named 'https://18.191.1.77/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#/l/service?namespace=default' with a type of 'kubernetes.io/service-account-token' and an age of 3 hours.

| Name | Labels | Cluster IP | Internal endpoints | External endpoints | Age |
|------------|----------------------------------------------|---------------|---------------------------------|--------------------|-----------|
| nginx | k8s-app: nginx | 100.70.189.16 | nginx:80 TCP nginx:32007 TCP | ad05abac1d6a111e8a | 7 minutes |
| kubernetes | component: apiserver provider: kubernetes | 100.64.0.1 | kubernetes:443 TCP | - | 3 hours |

| Name | Type | Age |
|----------------------------------------------------------------------------------------------------------------------------|-------------------------------------|---------|
| https://18.191.1.77/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#/l/service?namespace=default | kubernetes.io/service-account-token | 3 hours |

You can see that the app is mapped to an AWS load-balancer since this K8s cluster is running on AWS and is using Kubernetes Operation services.

The screenshot shows the Kubernetes dashboard interface. On the left is a sidebar menu with categories: Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Discovery and Load Balancing (selected), Ingresses, Services, Config and Storage, Config Maps, Persistent Volume Claims, and Secrets. The main content area is titled 'Discovery and load balancing > Services > nginx'. It features a 'Details' section and an 'Endpoints' table.

Details

| | | | |
|-------------------|----------------------|---------------------|---------------------------------------------------------------------------|
| Name: | nginx | Connection | |
| Namespace: | default | Cluster IP: | 100.70.189.16 |
| Labels: | k8s-app:nginx | Internal endpoints: | nginx:80 TCP nginx:32007 TCP |
| Creation Time: | 2018-10-23T08:58 UTC | External endpoints: | ad05abac1d6a111e8aa57021a64c42e3-873242590.us-east-2.elb.amazonaws.com:80 |
| Label selector: | k8s-app:nginx | | |
| Type: | LoadBalancer | | |
| Session Affinity: | None | | |

Endpoints

| Host | Ports (Name, Port, Protocol) | Node | Ready |
|------------|------------------------------|---------------------------------------------|-------|
| 100.96.2.5 | tcp-80-80-2hqm6, 80, TCP | ip-172-20-61-116.us-east-2.compute.internal | true |

NOTE: It takes around 5 Minutes for the service to get exposed and accessible through the load-balancer endpoint. Please try to access the service after 5 minutes.

7. Click on the **external endpoints** to access the application and if all the steps are followed correctly you will be able to see the nginx welcome page.

The screenshot shows a web browser window with the address bar displaying 'ad05abac1d6a111e8aa57021a64c42e3-873242590.us-east-2.elb.amazonaws.com'. The page content is as follows:

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

