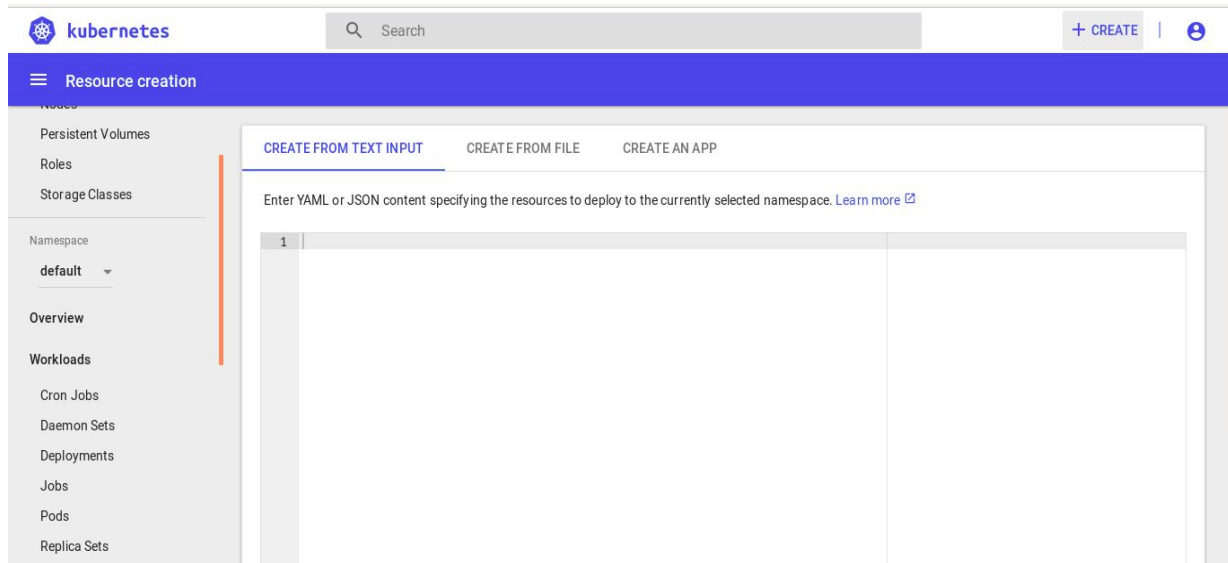


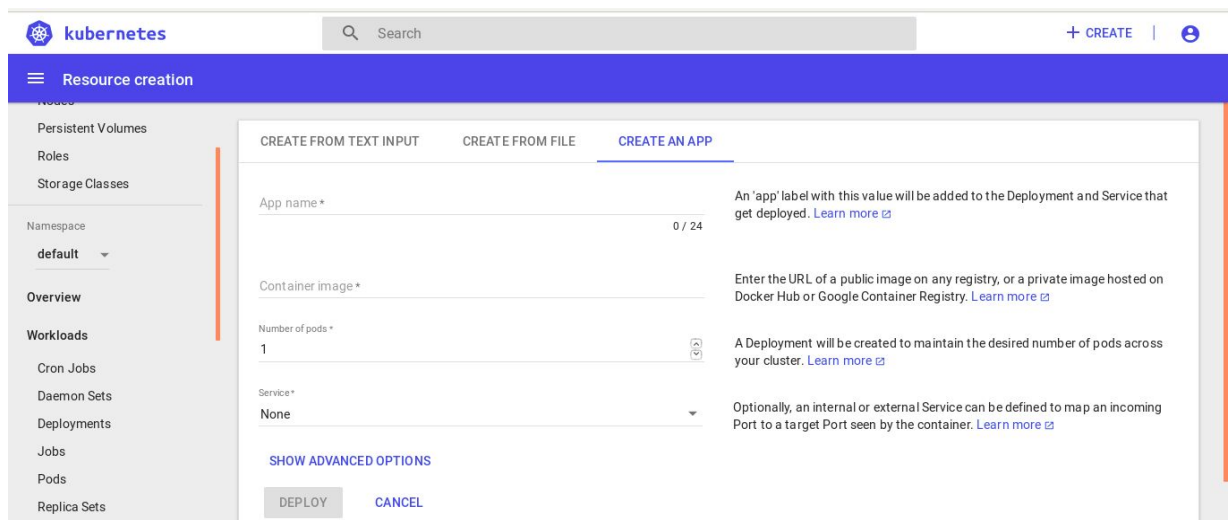
Run a Stateless Application Using a Deployment

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
Click SKIP when asked for the TOKEN.

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



2. Select the third option “CREATE AN APP”



3. Enter the values as follows

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

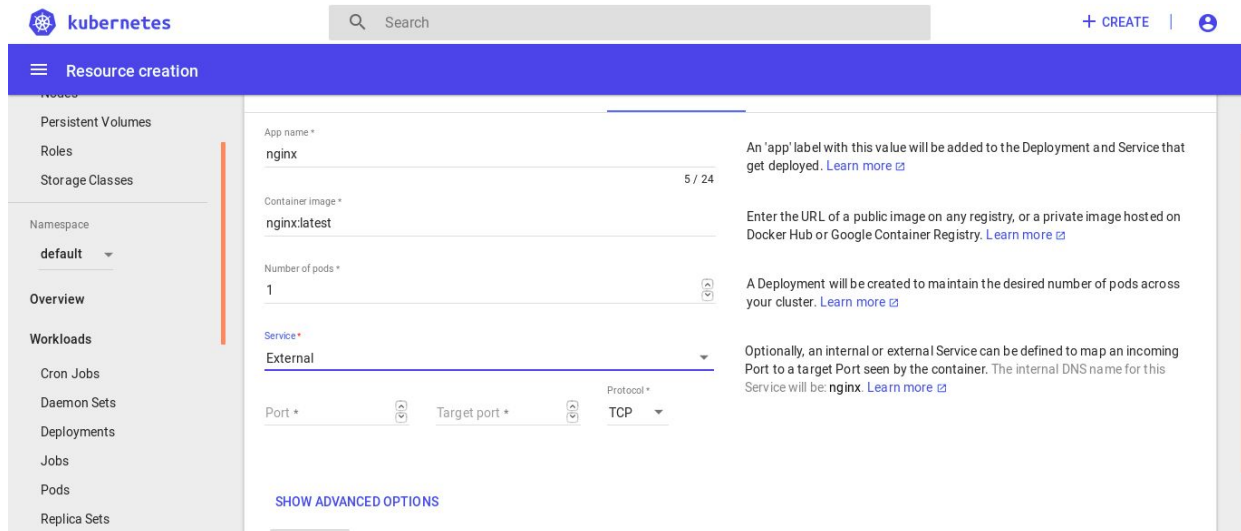
Container name : **nginx** (is the name of the Docker container that will be pulled from the official Docker repository)

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)




The screenshot shows the 'Resource creation' page in the Kubernetes dashboard. The left sidebar contains a navigation menu with 'Overview' selected. The main form is for creating a Deployment. The 'App name' field is filled with 'nginx'. The 'Container image' field is filled with 'nginx:latest'. The 'Number of pods' field is filled with '1'. The 'Service' dropdown is set to 'External'. The 'Port' field is filled with '80', the 'Target port' field is filled with '80', and the 'Protocol' dropdown is set to 'TCP'. There are informational text blocks on the right side of the form. At the bottom of the form, there is a 'SHOW ADVANCED OPTIONS' link.

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




The screenshot shows the bottom of the 'Resource creation' page. The left sidebar contains a navigation menu with 'Storage' selected. The main form is for creating a Deployment. The 'App name' field is filled with 'nginx'. The 'Container image' field is filled with 'nginx:latest'. The 'Number of pods' field is filled with '1'. The 'Service' dropdown is set to 'External'. The 'Port' field is filled with '80', the 'Target port' field is filled with '80', and the 'Protocol' dropdown is set to 'TCP'. There are informational text blocks on the right side of the form. At the bottom of the form, there is a 'SHOW ADVANCED OPTIONS' link. Below the form, there are two buttons: 'DEPLOY' and 'CANCEL'.

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.

 **kubernetes**

Search

+ CREATE | 

Overview

Nodes

Persistent Volumes

Roles

Storage Classes

Namespace

default

Overview

Workloads

Cron Jobs

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Workloads

Workloads Statuses

100.00%

Deployments


100.00%


Pods

100.00%


Replica Sets

Deployments

Name	Labels	Pods	Age	Images
 nginx	k8s-app: nginx	0 / 1	0 seconds	nginx:latest

 **kubernetes**

Search

+ CREATE | 

Workloads > Deployments

Cluster

Namespaces

Nodes

Persistent Volumes

Roles

Storage Classes

Namespace

default

Overview


Workloads

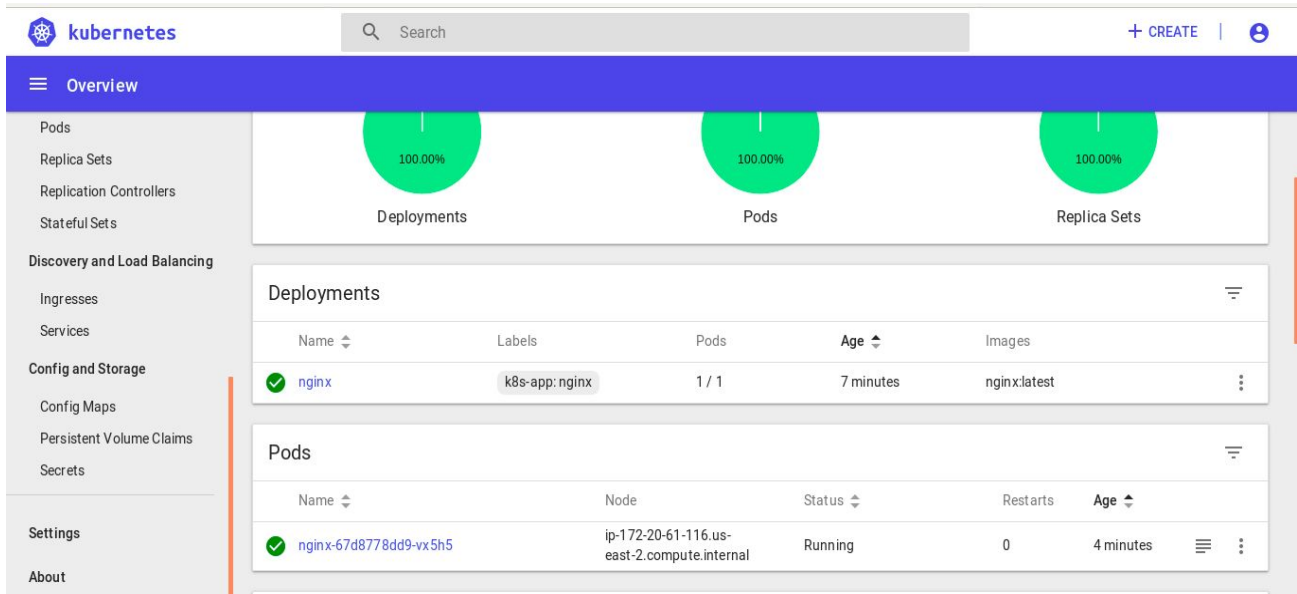
Cron Jobs

Daemon Sets

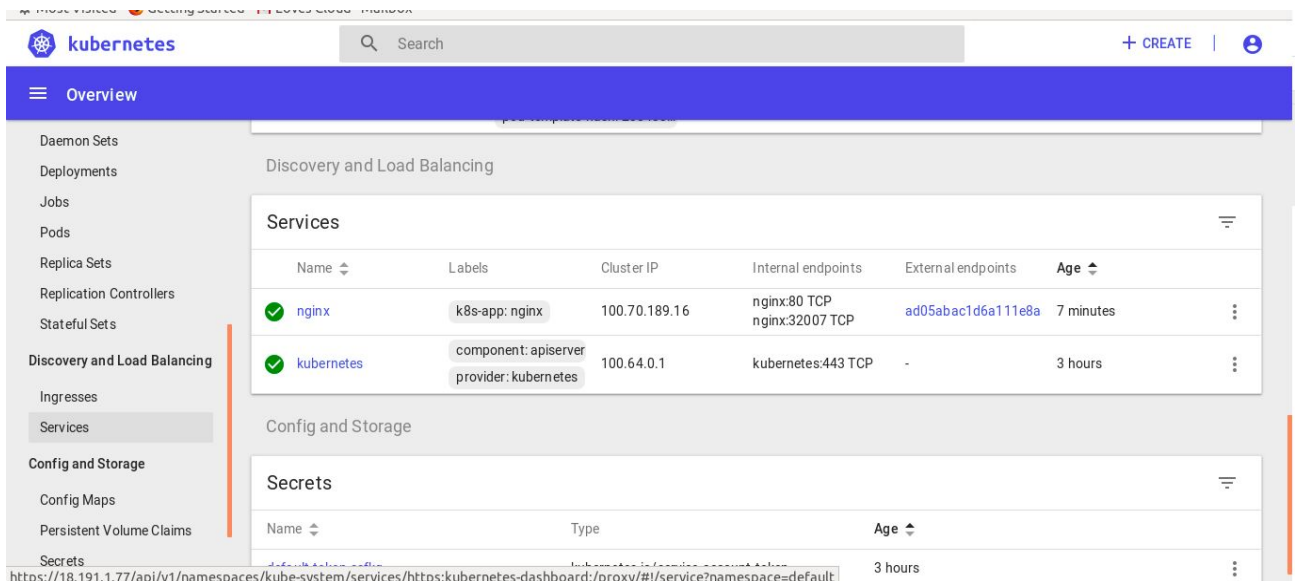
Deployments

Deployments

Name	Labels	Pods	Age	Images
 nginx	k8s-app: nginx	1 / 1	36 seconds	nginx:latest



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.



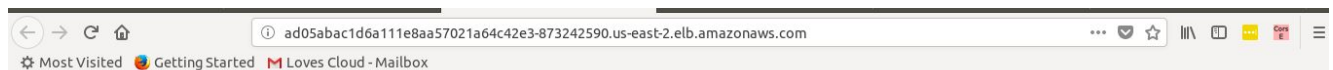
You can see that the app is mapped to an AWS load-balancer since this K8s cluster is running on AWS and is using Kubernetes Operation services.

The screenshot shows the Kubernetes dashboard interface. On the left is a sidebar with navigation links: Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Discovery and Load Balancing (selected), Ingresses, Services, Config Maps, Persistent Volume Claims, and Secrets. The main content area is titled 'Discovery and load balancing > Services > nginx'. It features a 'Details' section with the following information: Name: nginx, Namespace: default, Labels: k8s-app: nginx, Creation Time: 2018-10-23T08:58 UTC, Label selector: k8s-app: nginx, Type: LoadBalancer, and Session Affinity: None. The 'Connection' section shows Cluster IP: 100.70.189.16, Internal endpoints: nginx:80 TCP and nginx:32007 TCP, and External endpoints: ad05abac1d6a111e8aa57021a64c42e3-873242590.us-east-2.elb.amazonaws.com:80. Below this is an 'Endpoints' table with one entry.

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.



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.