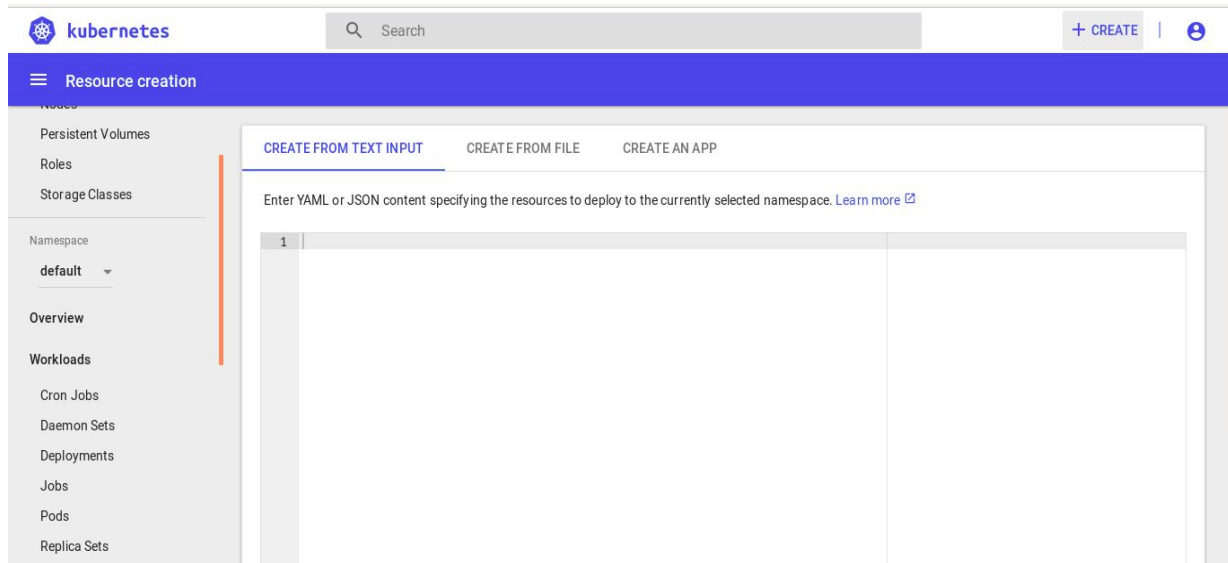


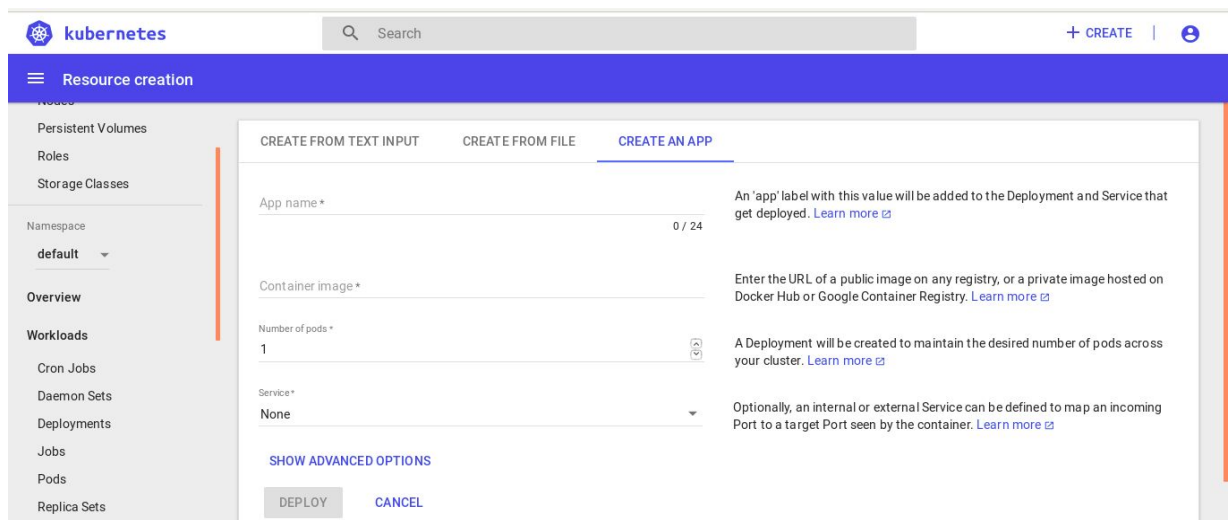
Run a Stateless Application Using a Deployment

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 : <app-name> (can be anything)

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)

Note: Do not Deploy the app over 3 else the deployments will fail as the threshold limit for scaling a Deployment is 3 on the higher side.

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)

Refer <https://kubernetes.io/docs/concepts/cluster-administration/networking/> for more information on cluster networking in Kubernetes.

Port: 80

Target Port: 80 (port mapping of the PODS {container} and the hos

The screenshot shows the 'Resource creation' page in the Kubernetes dashboard. The left sidebar contains a navigation menu with options: Persistent Volumes, Roles, Storage Classes, Namespace (set to 'default'), Overview, Workloads, Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, and Replica Sets. The main area is titled 'Resource creation' and contains the following fields:

- App name ***: nginx (Note: An 'app' label with this value will be added to the Deployment and Service that get deployed. [Learn more](#))
- Container image ***: nginx:latest (Note: Enter the URL of a public image on any registry, or a private image hosted on Docker Hub or Google Container Registry. [Learn more](#))
- Number of pods ***: 1 (Note: A Deployment will be created to maintain the desired number of pods across your cluster. [Learn more](#))
- Service ***: External (Note: Optionally, an internal or external Service can be defined to map an incoming Port to a target Port seen by the container. The internal DNS name for this Service will be: nginx. [Learn more](#))
- Port ***: (Note: Port mapping of the PODS {container} and the host)
- Target port ***: 80
- Protocol ***: TCP

At the bottom of the main area, there is a link to 'SHOW ADVANCED OPTIONS'.

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

The screenshot shows the bottom of the 'Resource creation' page. On the left sidebar, the 'Storage' section is expanded, showing 'Persistent Volume Claims', 'Config', and 'Secrets'. The main area has a 'SHOW ADVANCED OPTIONS' link and two buttons: 'DEPLOY' and 'CANCEL'.

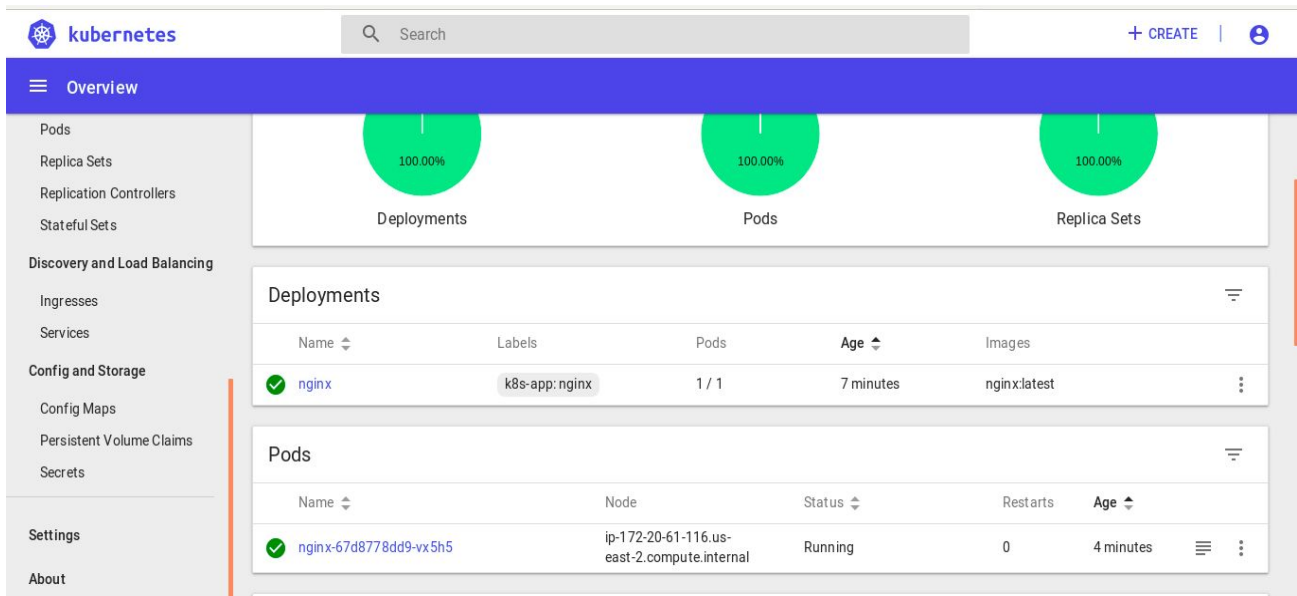
5. Click on the **Deployments** tab on the left hand side to Review the deployed nginx application details.

The screenshot shows the Kubernetes Overview dashboard. The left sidebar contains navigation links for Overview, Nodes, Persistent Volumes, Roles, Storage Classes, Namespace (default), Overview, Workloads, Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, and Replica Sets. The main content area is titled 'Workloads' and features a 'Workloads Statuses' section with three circular progress indicators, each showing 100.00% for Deployments, Pods, and Replica Sets. Below this is a 'Deployments' table with one entry: 'nginx' with label 'k8s-app: nginx', 0 / 1 pods, 0 seconds age, and image 'nginx:latest'.

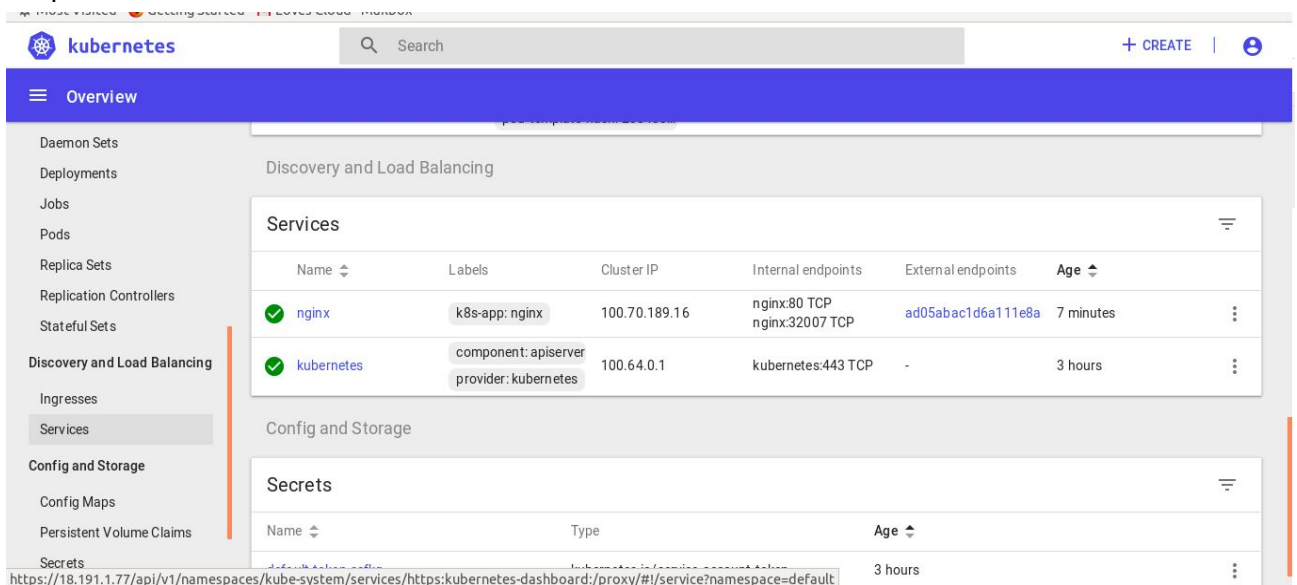
Name	Labels	Pods	Age	Images
nginx	k8s-app: nginx	0 / 1	0 seconds	nginx:latest

The screenshot shows the Kubernetes Workloads > Deployments page. The left sidebar contains navigation links for Cluster, Namespaces, Nodes, Persistent Volumes, Roles, Storage Classes, Namespace (default), Overview, Workloads, Cron Jobs, Daemon Sets, and Deployments. The main content area is titled 'Deployments' and features a table with one entry: 'nginx' with label 'k8s-app: nginx', 1 / 1 pods, 36 seconds age, and image 'nginx:latest'.

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



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

Details

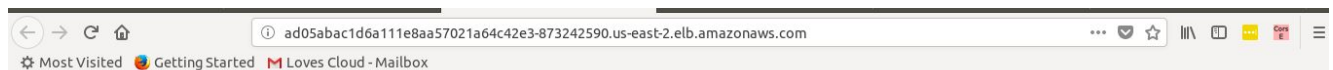
Name:	nginx	Connection	
Namespace:	default	Cluster IP:	100.70.189.16
Labels:	k8s-app: nginx	Internal endpoints:	nginx:80 TCP nginx:32007 TCP
Creation Time:	2018-10-23T08:58 UTC	External endpoints:	ad05abac1d6a111e8aa57021a64c42e3-873242590.us-east-2.elb.amazonaws.com:80
Label selector:	k8s-app: nginx		
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.



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.