Hands-on Lab: (Optional) Deploy an application to Kubernetes

**Estimated time needed:** 30 minutes

Lab Overview

In this hands-on lab we'll provision a free Kubernetes cluster and by using the IBM Cloud Shell we'll deploy a sample application.

Objectives

After completing this lab, you will be able to:

1. Spin up a free cluster of the Kubernetes service
2. Manage and connect to your cluster using the IBM Cloud CLI
3. Run kubectl commands from the IBM Cloud Shell

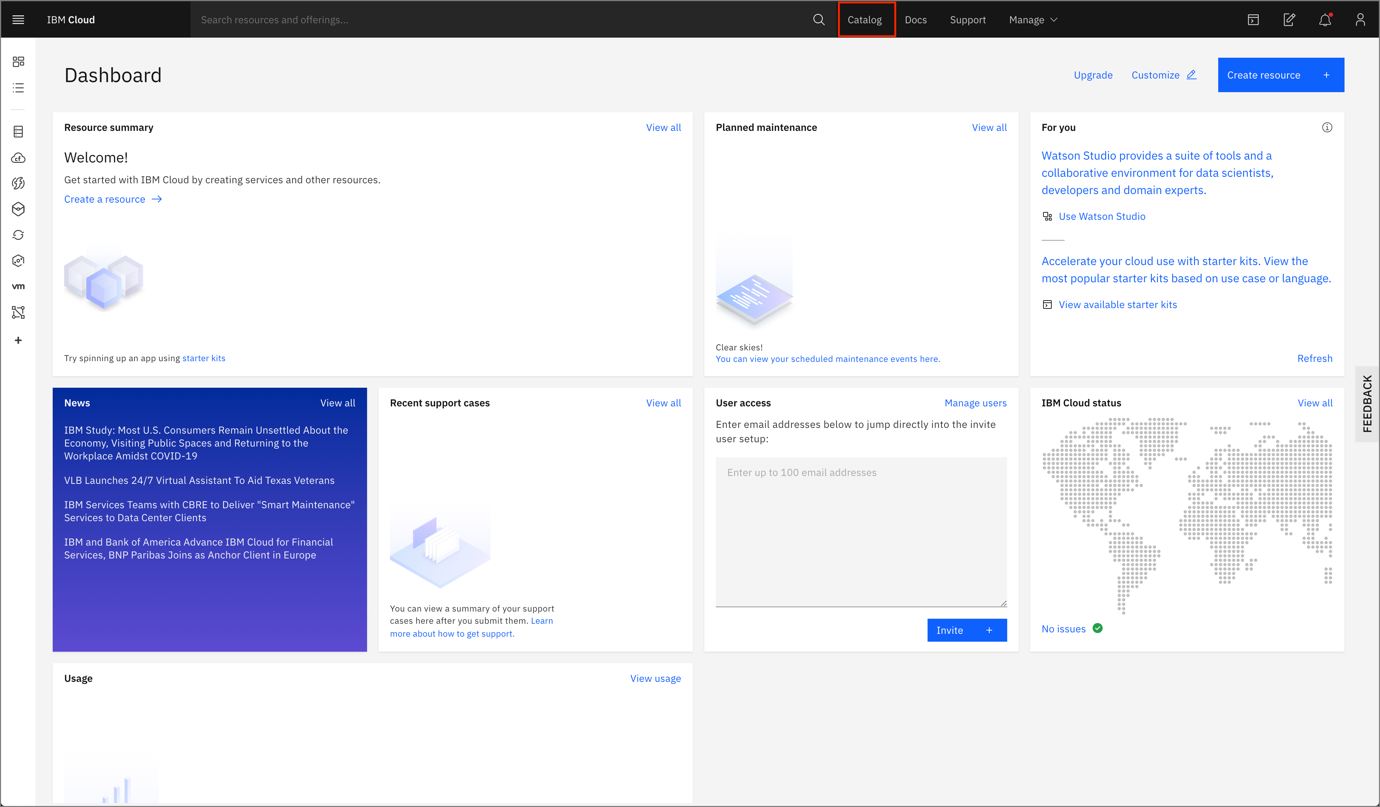
Lab Instructions

This lab is broken up into the following tasks:

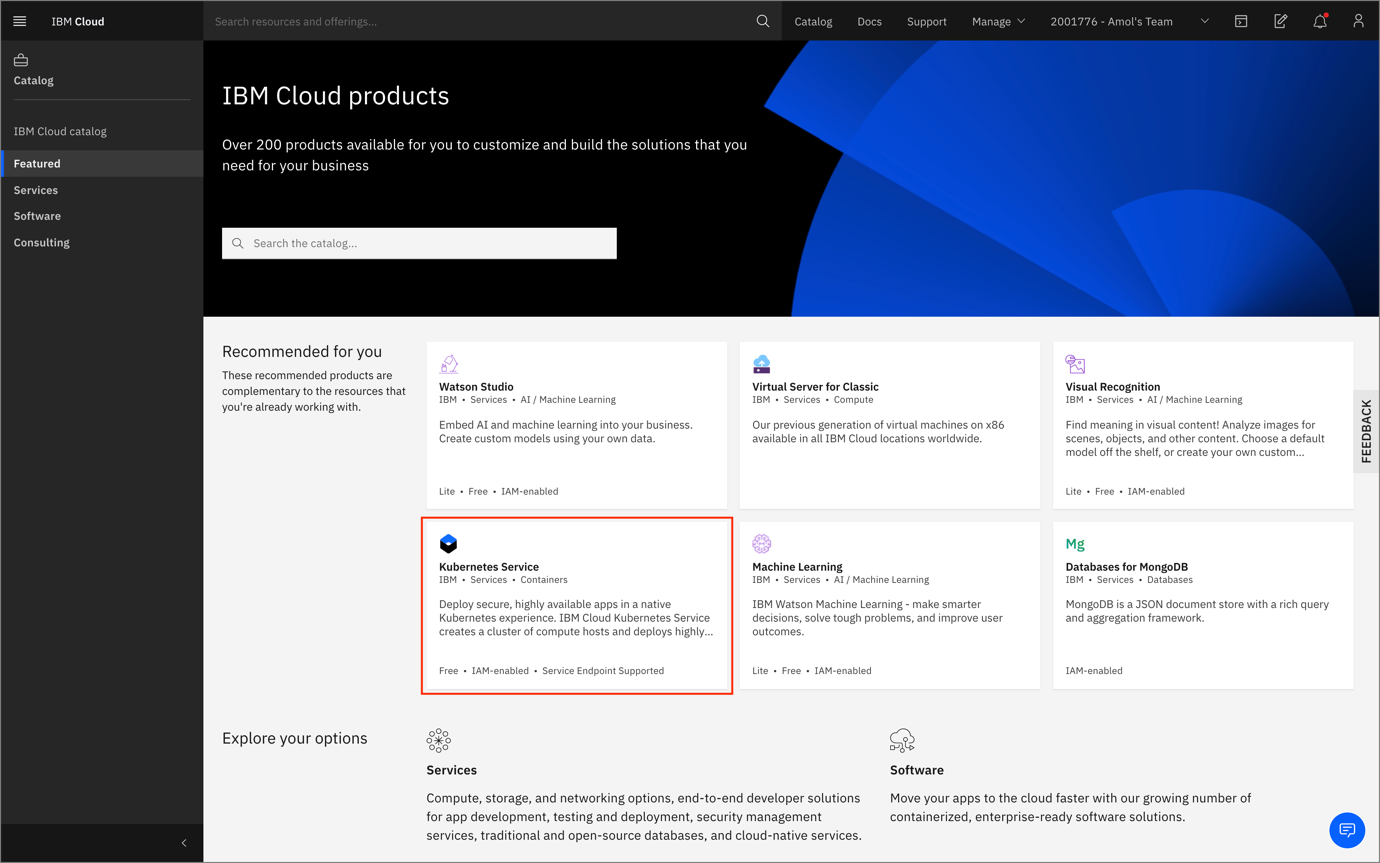
1. [Create a free Kubernetes cluster](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-CC0103EN-SkillsNetwork/labs/module-3-lab-1/index.md.html#1-create-a-free-kubernetes-cluster)
2. [Launch the IBM Cloud Shell](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-CC0103EN-SkillsNetwork/labs/module-3-lab-1/index.md.html#2-launch-the-ibm-cloud-shell)
3. [Connect to your cluster](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-CC0103EN-SkillsNetwork/labs/module-3-lab-1/index.md.html#3-connect-to-your-cluster)
4. [Deploy the application](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-CC0103EN-SkillsNetwork/labs/module-3-lab-1/index.md.html#4-deploy-the-application)
5. [Access the web app](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-CC0103EN-SkillsNetwork/labs/module-3-lab-1/index.md.html#5-access-the-web-app)
6. [Test it out](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-CC0103EN-SkillsNetwork/labs/module-3-lab-1/index.md.html#6-test-it-out)
7. [Clean up](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-CC0103EN-SkillsNetwork/labs/module-3-lab-1/index.md.html#7-clean-up)

1. Create a free Kubernetes cluster

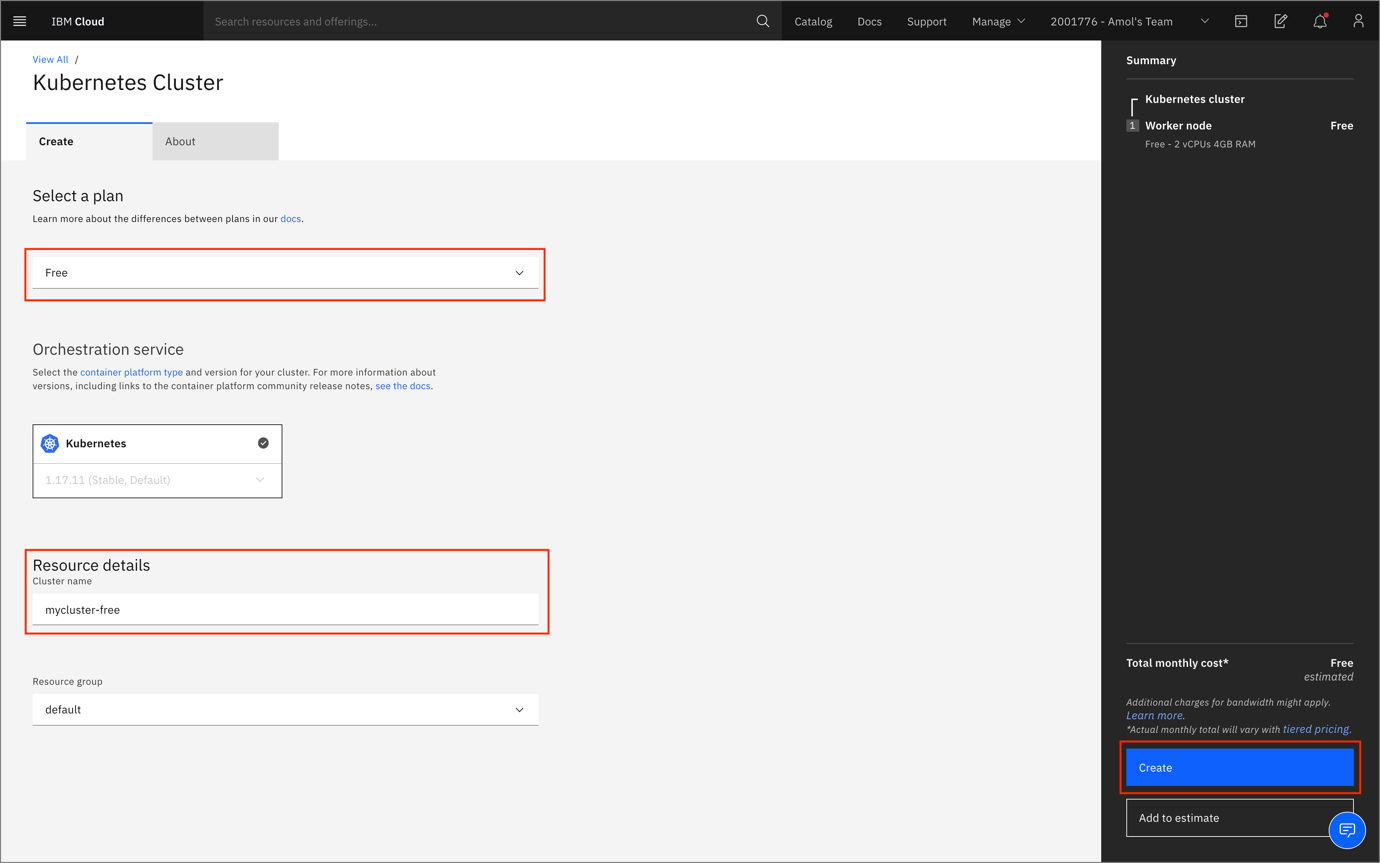
Navigate to [https://cloud.ibm.com](https://cloud.ibm.com/?utm_email=Email&utm_source=Nurture&utm_content=000026UJ&utm_term=10006555&utm_campaign=PLACEHOLDER&utm_id=SkillsNetwork-Courses-IBMDeveloperSkillsNetwork-CC0103EN-SkillsNetwork-19956175) to log into IBM Cloud. You'll be presented with an empty console. Click on the **Catalog** button on the top-right.



From the catalog, choose the **Kubernetes** option.

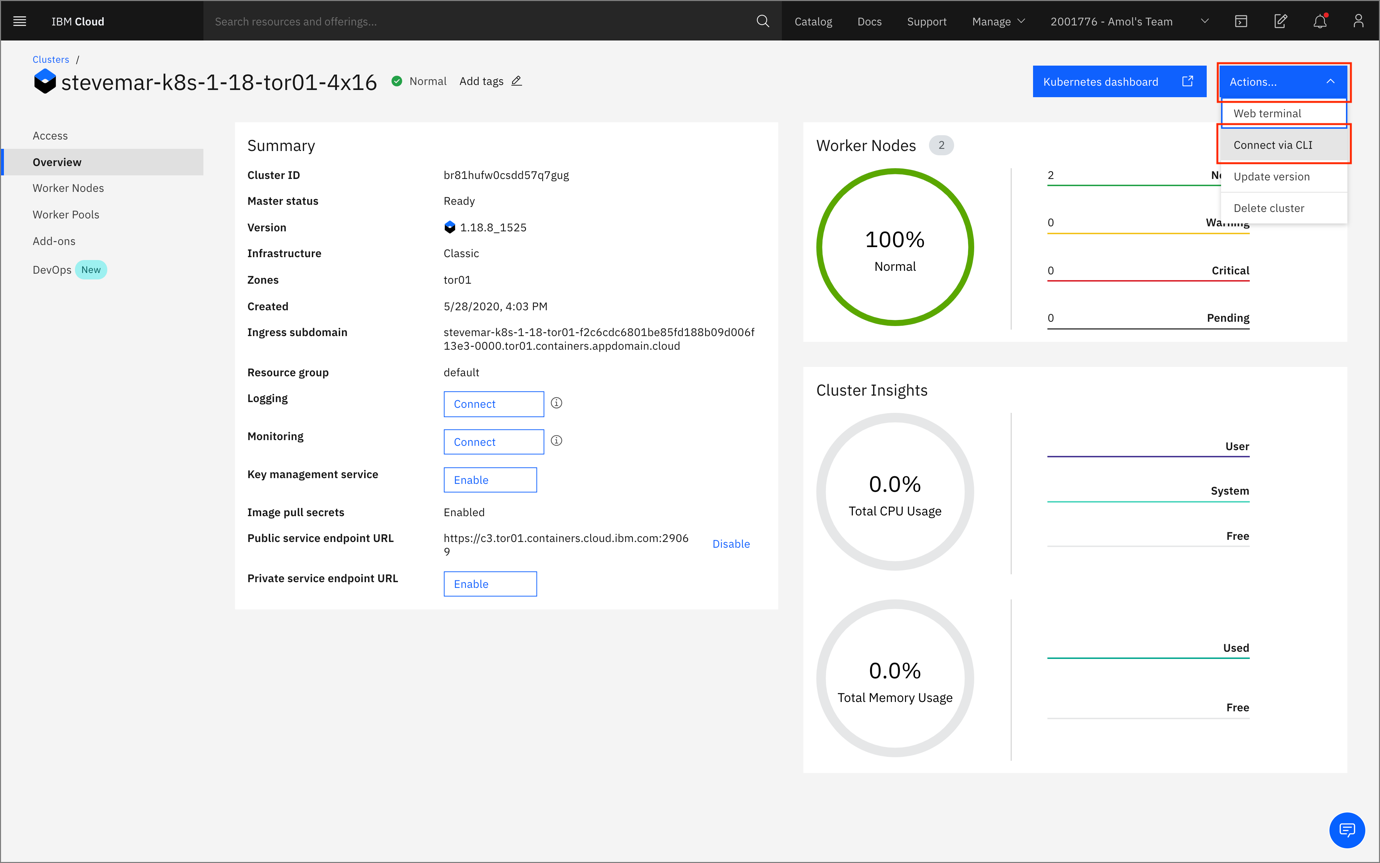


Choose the **Free** tier option, give your cluster a unique name and click **Create**.

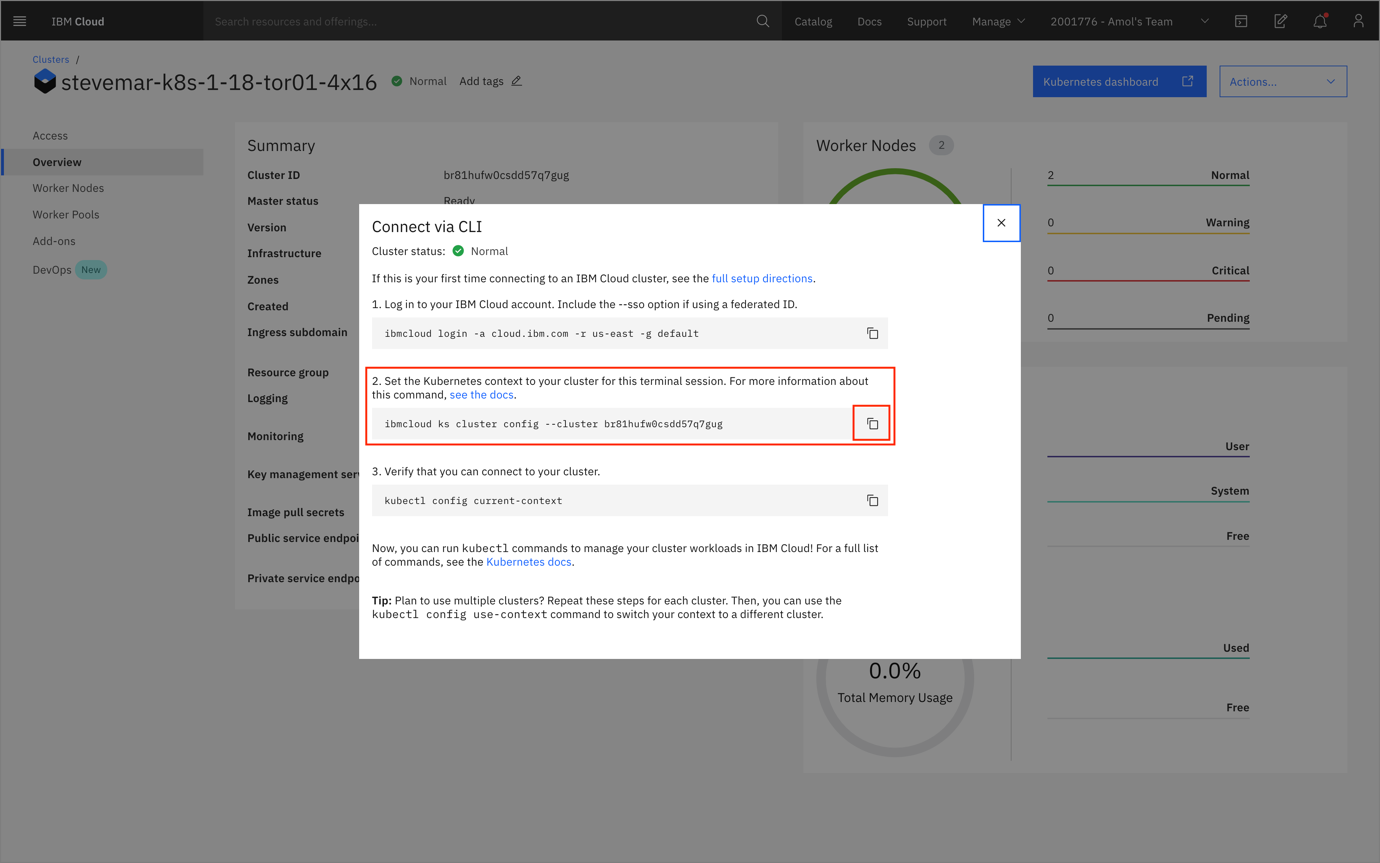


2. Launch the IBM Cloud Shell

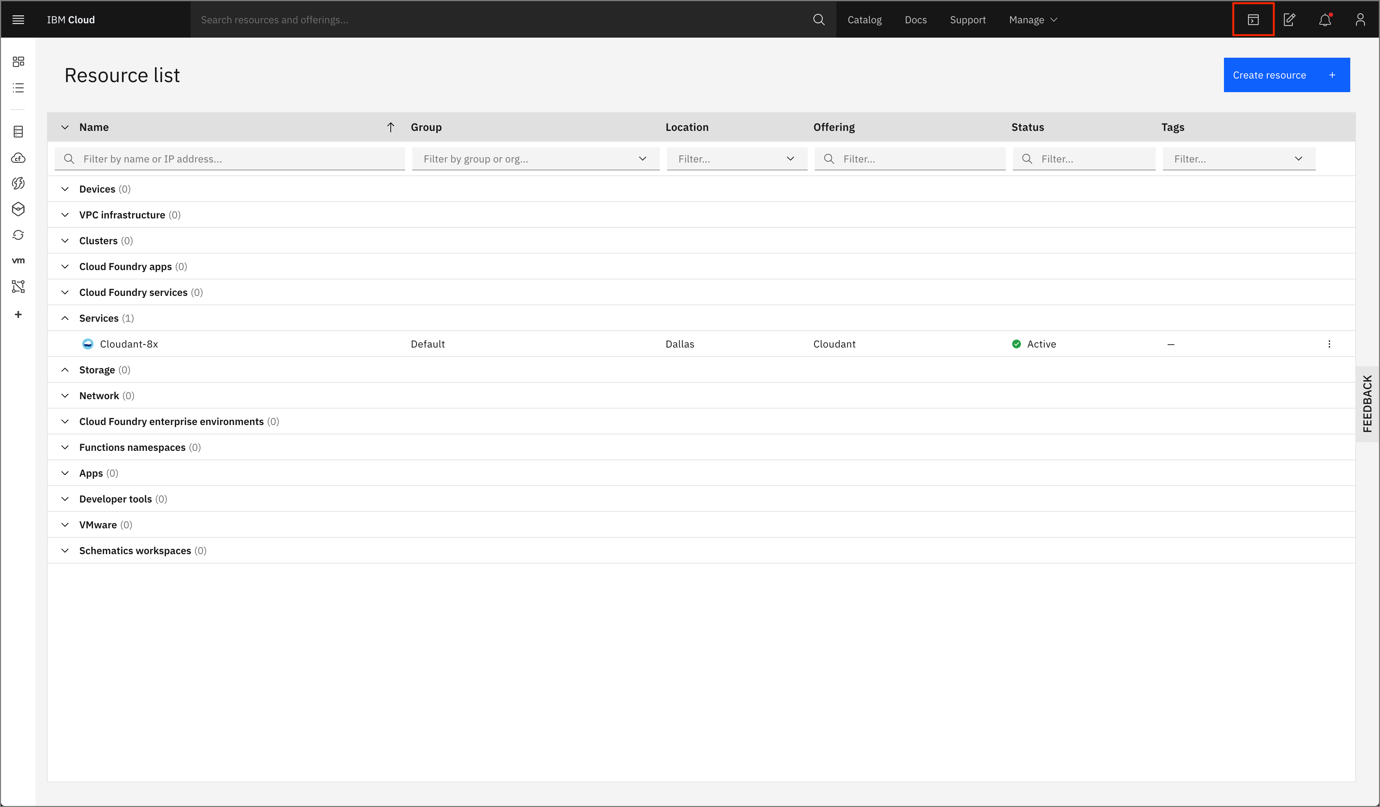
Once your cluster is created click on the **Actions** menu on the top right, and choose the **Connect via CLI** option.



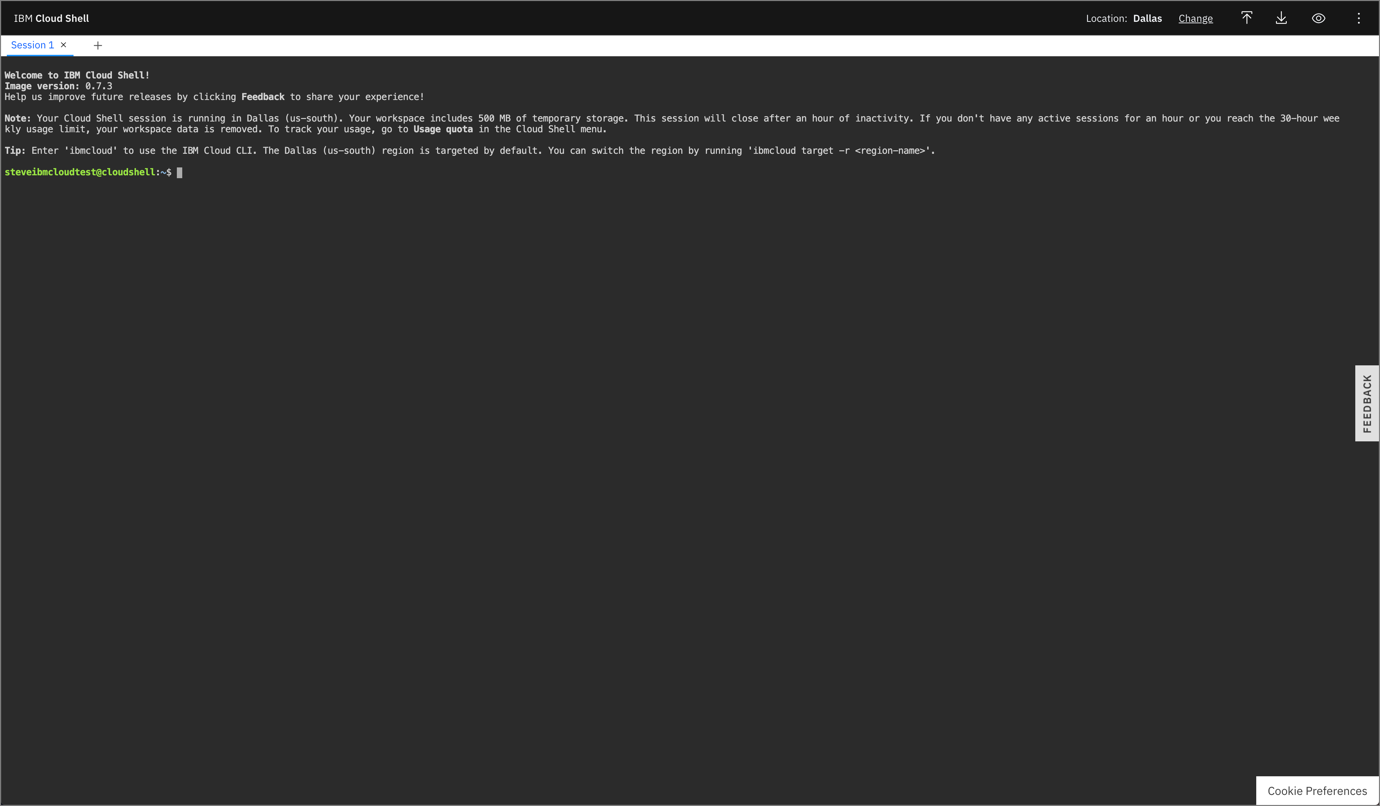
Click on the clipboard to copy the command, the one using the ibmcloud ks cluster config command. This will set any kubectl commands to use your newly created cluster.



Next, we'll be using the *IBM Cloud Shell*, which is available by clicking on the terminal icon on the top right.



Once loaded, you'll be presented with an embedded web terminal that has the ibmcloud and kubectl CLIs available.



3. Connect to your cluster

From the IBM Cloud Shell, run the command that you command, replacing your cluster ID in the command below.

ibmcloud ks cluster config --cluster <YOUR\_CLUSTER\_ID>

You should see output like the example below:

$ ibmcloud ks cluster config --cluster br81hufw0csdd57q7gug

The configuration **for** br81hufw0csdd57q7gug was downloaded successfully.

Added context **for** br81hufw0csdd57q7gug to the current kubeconfig file.

You can now execute 'kubectl' commands against your cluster. For example, run 'kubectl get nodes'.

4. Deploy the application

In this lab we'll run a sample application that detects objects in a picture. Deploy the application by running the command below.

kubectl apply -f https://raw.githubusercontent.com/IBM/MAX-Object-Detector/master/max-object-detector.yaml

You should see output like the example below:

$ kubectl apply -f https://raw.githubusercontent.com/IBM/MAX-Object-Detector/master/max-object-detector.yaml

service/max-object-detector created

deployment.apps/max-object-detector created

5. Access the web app

After a few minutes, the application should be available. To access it we need to find the public IP address of the cluster, and the port the application is running on.

To find the public IP address of the cluster run the command below:

ibmcloud cs workers --cluster <YOUR\_CLUSTER\_ID>

You should see output like the example below:

$ ibmcloud cs workers --cluster br81hufw0csdd57q7gug

ID Public IP Private IP Flavor State Status Zone Version

kube-br81hufw0csdd57q7gug-stevemark8s-default-000001ab 169.55.160.163 10.167.24.59 b3c.4x16.encrypted normal Ready tor01 1.18.3\_1514

*The public IP address for this cluster is 169.55.160.163.*

And to find the port the application is running on, run the following command:

kubectl describe service max-object-detector | grep NodePort

You should see output like the example below:

$ kubectl describe service max-object-detector | grep NodePort

Type: NodePort

NodePort: <unset> 30071/TCP

*The port for this application is 30071.*

6. Test it out

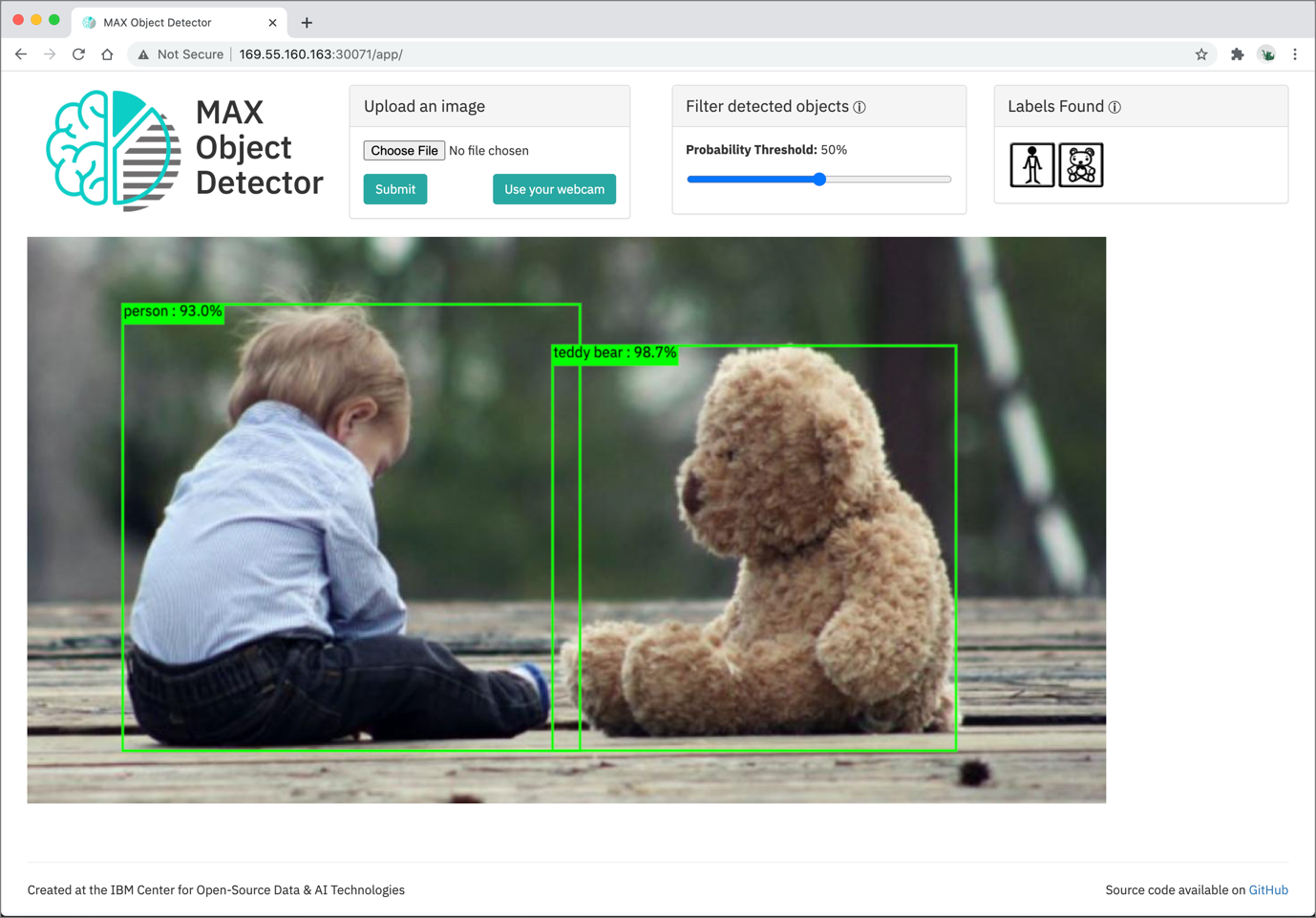
From your browser navigate to the URL

http://<public-ip-address:<port>/app

In the example above, it looked like:

http://169.55.160.163:30071/app

To use the application start by upload a picture. The application will using AI to detect objects in the picture. Here is some example output.



7. Clean up

Should you want to delete your Kubernetes cluster you can run the command below.

ibmcloud cs cluster rm -c <YOUR\_CLUSTER\_ID> -f

**Congratulations!**