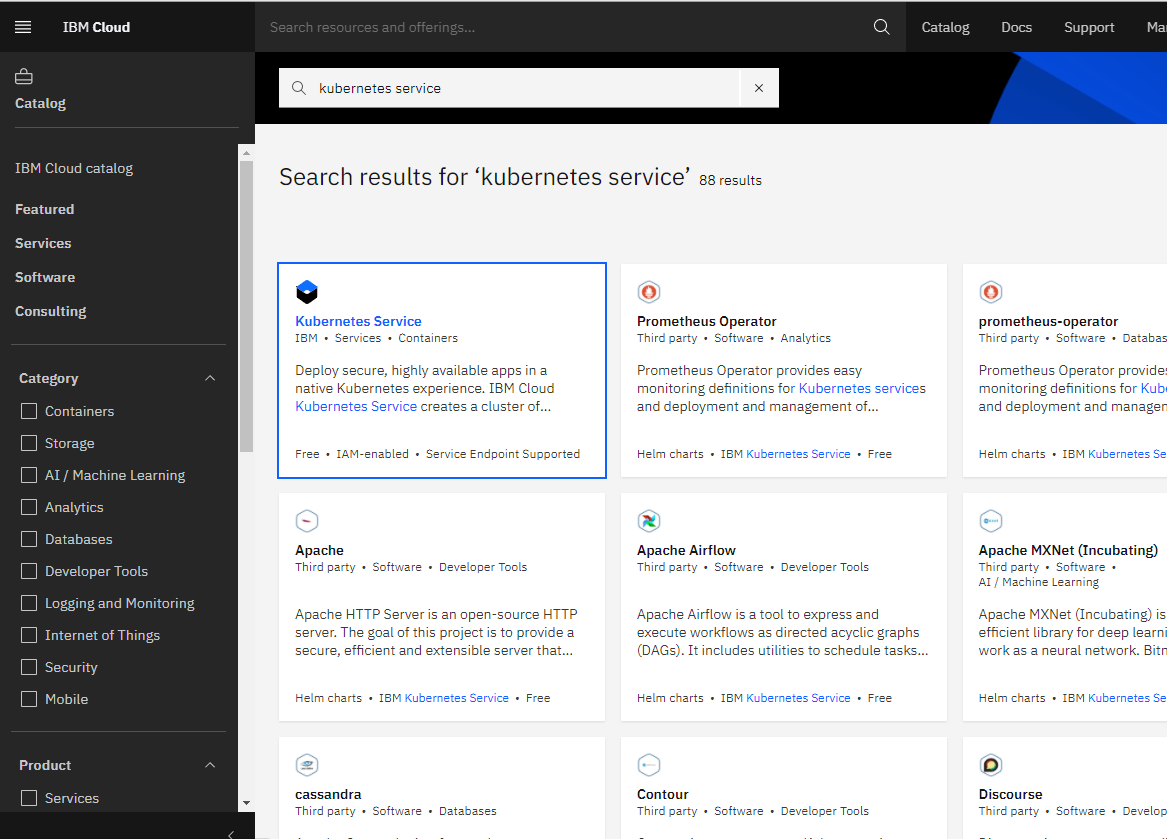
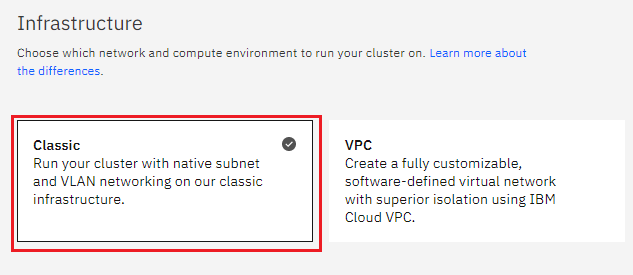
# Installing NGINX Ingress controller on IBM Cloud

**Step 1 provision Kubernetes Cluster**

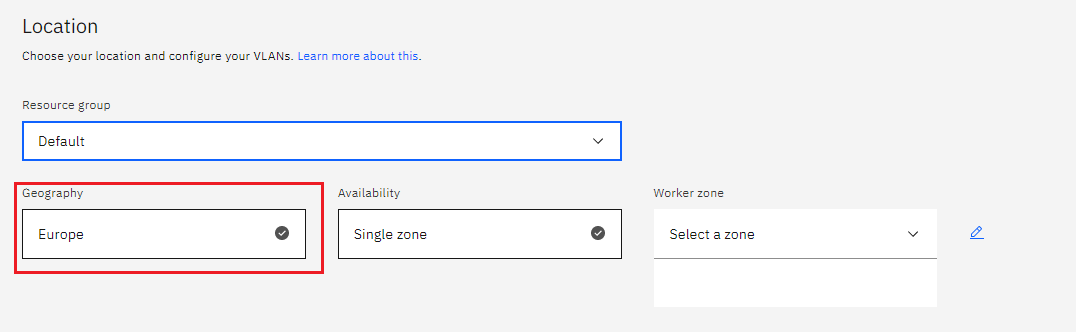
* Click the **Catalog** button on the top
* Select **Service** from the **Catalog**
* Search for **Kubernetes Service** and click on it



* You are now at the Kubernetes deployment page. You need to specify some details about the cluster
* Choose a plan **standard** or **free** , the free plan only has one worker node and no subnet, to provision a standard cluster, you will need to upgrade your account to Pay-As-You-Go
* To upgrade to a Pay-As-You-Go account, complete the following steps:
* In the console, go to Manage > Account.
* Select Account settings; and click Add credit card.
* Enter your payment information, click Next, and submit your information
* Choose **classic** or **VPC** , read the docs and choose the most suitable type for yourself



* Now choose your location settings,
* Choose **Geography** (continent)



* + Choose Single or Multizone, in single zone your data is only kept in on datacenter, on the

other hand with Multizone it is distributed to multiple zones, thus safer in an unforeseen

zone failure

* If you wish to use Multizone please set up your account with[VRF
* If at your current location selection, there is no available Virtual LAN, a new Vlan will be created for you
* Choose a Worker node setup or use the preselected one, set Worker node amount per zone
* Choose **Master Service Endpoint** , In VRF-enabled accounts, you can choose private-only to make your master accessible on the private network or via VPN tunnel. Choose public-only to make your master publicly accessible. When you have a VRF-enabled account, your cluster is set up by default to use both private and public endpoints.  
  Give desired **tags** to your cluster, for more information visit tags
* Click **create**  
  • Wait for your cluster to be provisioned  
  • Your cluster is ready for usage

**Step 2 Deploy IBM Cloud Block Storage plug-in**

The Block Storage plug-in is a persistent, high-performance iSCSI storage that you can add to your apps by using Kubernetes Persistent Volumes (PVs).

* Click the **Catalog** button on the top
* Select **Software** from the catalog
* Search for **IBM Cloud Block Storage plug-in** and click on it  
  • On the application page Click in the dot next to the cluster, you wish to use  
  • Click on Enter or Select Namespace and choose the default Namespace or use a custom one (if you get error please wait 30 minutes for the cluster to finalize)
* Give a **name** to this workspace
* Click **install** and wait for the deployment

**Step 3 Installing NGINX Ingress controller**

By default, the Ingress Controller requires a number of custom resource definitions (CRDs) installed on the cluster. Helm 3.x client will install those CRDs. If you’re using a Helm 2.x client, you need to install the CRDs via kubectl:

$ kubectl create -f crds/

Change your working directory to /deployments/helm-chart:

*$ cd kubernetes-ingress/deployments/helm-chart*

*$ git checkout v1.9.1*

## **Adding the Helm Repository**

This step is required if you’re installing the chart via the helm repository.

*$ helm repo add nginx-stable https://helm.nginx.com/stable*

*$ helm repo update*

### **Installing via Helm Repository**

To install the chart with the release name my-release (my-release is the name that you choose):

* Using Helm 3.x client:

For NGINX:

*$ helm install my-release nginx-stable/nginx-ingress*

For NGINX Plus: (assuming you have pushed the Ingress controller image nginx-plus-ingress to your private registry myregistry.example.com)

*$ helm install my-release nginx-stable/nginx-ingress --set controller.image.repository=myregistry.example.com/nginx-plus-ingress --set controller.nginxplus=true*

* Using Helm 2.x client:

For NGINX:

$ helm install --name my-release nginx-stable/nginx-ingress

For NGINX Plus: (assuming you have pushed the Ingress controller image nginx-plus-ingress to your private registry myregistry.example.com)

$ helm install --name my-release nginx-stable/nginx-ingress --set controller.image.repository=myregistry.example.com/ng

## **Upgrading the Chart**

### **Upgrading the CRDs**

**Note**: If you’re using Kubernetes 1.14, make sure to add --validate=false to the kubectl apply command below.

Helm does not upgrade the CRDs during a release upgrade. Before you upgrade a release, run the following command to upgrade the CRDs:

$ kubectl apply -f crds/

**Note**: The following warning is expected and can be ignored: Warning: kubectl apply should be used on resource created by either kubectl create --save-config or kubectl apply.

### **Upgrading the Release**

To upgrade the release my-release:

#### **Upgrade Using Chart Sources:**

$ helm upgrade my-release .

#### **Upgrade via Helm Repository:**

$ helm upgrade my-release nginx-stable/nginx-ingress

## **Uninstalling the Chart**

### **Uninstalling the Release**

To uninstall/delete the release my-release:

* Using Helm 3.x client:
* $ helm uninstall my-release
* Using Helm 2.x client:
* $ helm delete --purge my-release

The command removes all the Kubernetes components associated with the chart and deletes the release.

### **Uninstalling the CRDs**

Uninstalling the release does not remove the CRDs. To remove the CRDs, run:

$ kubectl delete -f crds/