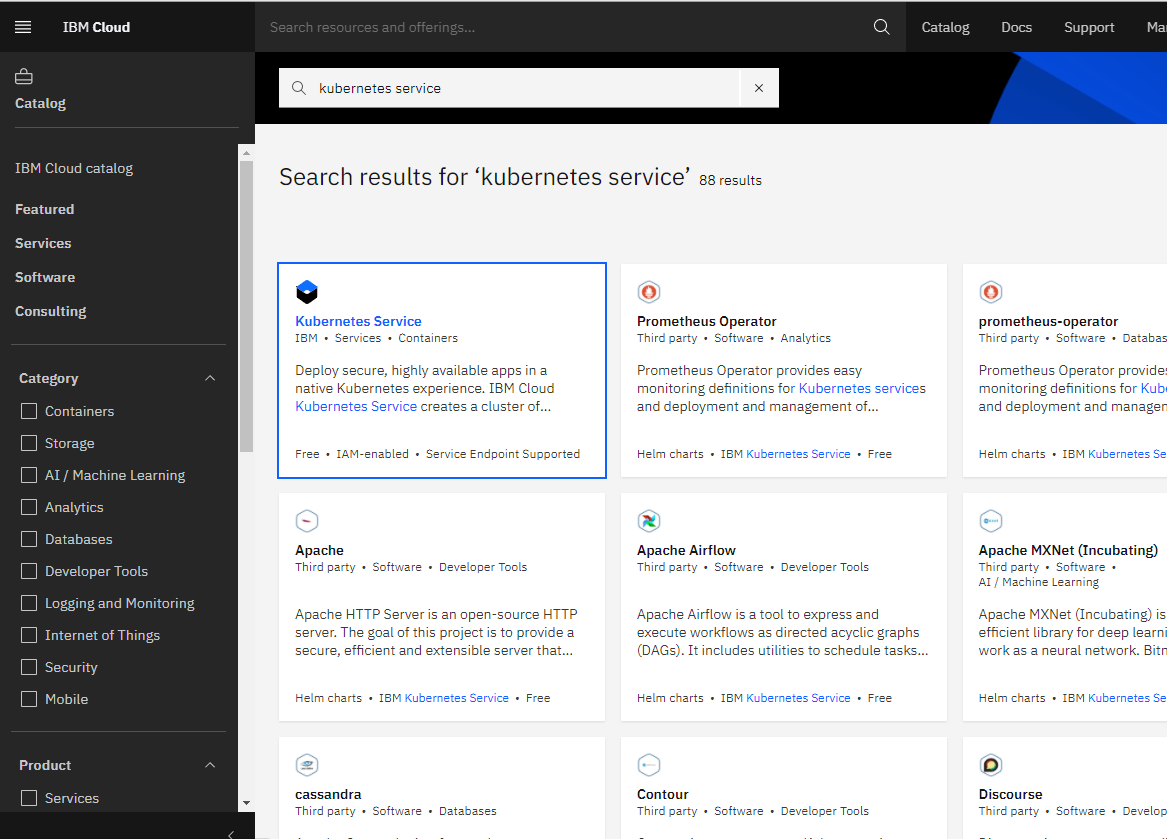
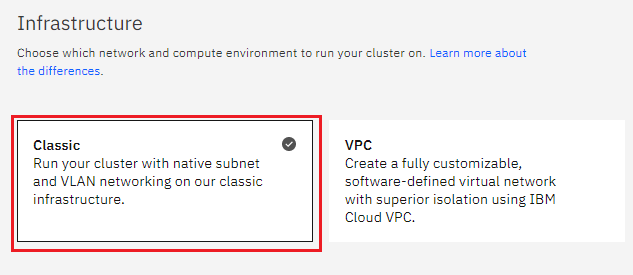
**Installing NAT on IBM Cloud**

**Step 1 provision Kubernetes Cluster**

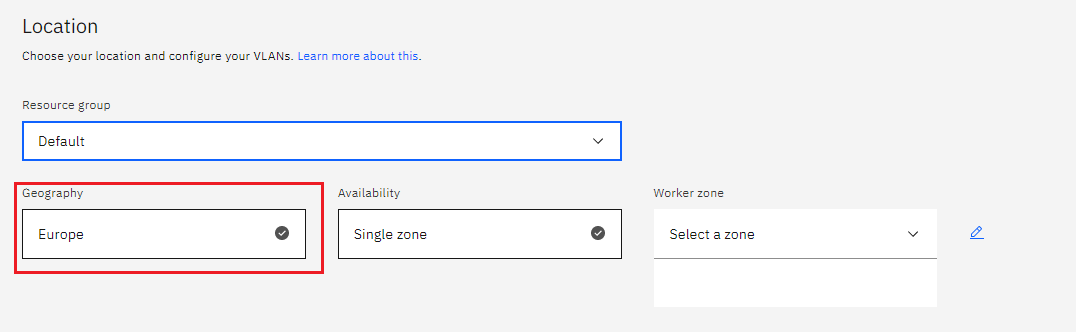
* Click the **Catalog** button on the top
* Select **Service** from the **Catalog**
* Search for **Kubernetes Service** and click on it



* 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 NATS**

1. Install [Docker](https://docs.docker.com/install)
2. Install [Helm Client](https://helm.sh/docs/using_helm/#installing-the-helm-client)
3. Install [kubectl](https://kubernetes.io/docs/tasks/tools/install-kubectl/" \l "install-kubectl" \t "_blank)
4. Install [IBM Cloud CLI](https://cloud.ibm.com/docs/cli/reference/ibmcloud?topic=cloud-cli-install-ibmcloud-cli#shell_install)
5. Install IBM Cloud Kubernetes service (IKS) plugin
6. Login IBM Cloud account
7. Initialize IKS plugin
8. Set kubectl to manage IKS cluster
9. Verify kubectl settings for cluster

**Step 4 Deploy NATS**

1. Download NATS Streaming server helm package

$ wget -O /tmp/nats-ss-0.0.1.tgz <https://github.com/ssibm/iks-nats-streaming/raw/master/deploy/nats-ss-0.0.1.tgz>

1. Create deployment yaml files for NATS Streaming StatefulSet, Service, Persistent Volume, and Persistent Volume Claim. In the command below, persistence will be set to local

$ mkdir -p /tmp/helm-output  
$ helm template --name test-drive --set persistence.local.enabled=true --output-dir /tmp/helm-output /tmp/nats-ss-0.0.1.tgzOutput:  
wrote /tmp/helm-output/nats-ss/templates/pv.yaml  
wrote /tmp/helm-output/nats-ss/templates/pvc.yaml  
wrote /tmp/helm-output/nats-ss/templates/service.yaml  
wrote /tmp/helm-output/nats-ss/templates/statefulset.yaml

1. Deploy NATS Streaming on IKS Cluster using the deployment files

$ kubectl apply --recursive --filename /tmp/helm-output/nats-ssOutput:  
persistentvolume/pv-nats-ss created  
persistentvolumeclaim/pvc-nats-ss created  
service/test-drive-nats-ss created  
statefulset.apps/test-drive-nats-ss create

1. It may take few minutes for the Persistence Volume Claim to finish binding and for the Pods to be in Running state. Verify using the following command

$ kubectl get all -l release=test-driveNAME READY STATUS RESTARTS AGE  
pod/test-drive-nats-ss-0 1/1 Running 0 6m55sNAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE  
service/test-drive-nats-ss LoadBalancer 172.21.108.200 169.46.126.110 4222:30882/TCP,8222:30685/TCP 6m55sNAME DESIRED CURRENT AGE  
statefulset.apps/test-drive-nats-ss 1 1 6m55s

5. After the Pods are in the  ***Running*** state, go to **http://*<EXTERNAL-IP>*:8222** to confirm NATS Streaming monitoring page loads successfully.

Subscribe to NATS Streaming channel

A docker container will be started to run on local machine to create a subscription on the NATS Streaming running in IKS. This subscription will be tied to a single channel named ‘*test-channel*’.

1. Open a new terminal window/tab. Refer to this terminal session as ‘*subscription-terminal*’

2. Run following command to start a docker container that starts a subscription client to the NATS Streaming service running in IKS at **nats://*<EXTERNAL-IP>*:4222**

$ docker run --name test-subscriber-1 --sig-proxy=false sside/nats-ss-client examples/stan-sub.js -c test-cluster -s “nats://169.46.126.110:4222” -id sub1 test-channel --allOutput:  
STAN connected!  
subscribed to test-channel qgroup:

3. Leave ‘*subscription-terminal*’ running

**Publish to NATS Streaming channel**

Using a docker container, a new message will be published to the NATS Streaming test-channel. Subscription client created above will receive the message on the same channel.

* 1. Open a new terminal window/tab. Refer to this terminal session as ‘*publisher-terminal*’

2. Run following command in publisher-terminal to publish a new message to ‘test-channel’ on NATS Streaming running in IKS at **nats://*<EXTERNAL-IP>*:4222**

$ docker run --rm --name test-publisher sside/nats-ss-client examples/stan-pub.js -c test-cluster -s “nats://169.46.126.110:4222” -id pub1 test-channel ‘message-one to subject test-channel’Output:  
published test-channel (VCQQJ3HBJ7GQS0YBESCSA4)

Verify Subscription client

1. In terminal session ‘subscription-terminal’, following output should be displayed confirming receipt of the published message.

STAN connected!  
subscribed to test-channel qgroup:  
**test-channel[1]: *message-one to subject test-channel***

This confirms NATS Streaming is successfully deployed and running in an IKS cluster.