Deploying RabbitMQ in Kubernetes with Helm

Prerequisites

Kubernetes Cluster: Ensure you have access to a Kubernetes cluster.

Helm: Install Helm on your local machine. You can follow the instructions here.

Steps

Add Bitnami Helm Repository:

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

Update Helm Repositories:

```
helm repo update
```

Fetch RabbitMQ Values:

```
helm show values bitnami/rabbitmq > values.yaml
```

Review and Modify Configuration:

Open values.yaml in a text editor to review and modify the RabbitMQ configuration as needed. You can adjust parameters such as replica count, storage configuration, and RabbitMQ credentials.

Install RabbitMQ:

```
helm install my-release bitnami/rabbitmq -f values.yaml
```

Replace my-release with your preferred release name.

Verify Deployment:

After installation, verify that RabbitMQ has been deployed successfully by checking the status of the pods:

```
kubectl get pods
```

Ensure that all RabbitMQ pods are running and ready.

Access RabbitMQ Management Console:

Once done use below command that you will get once by applying helm install command. These are the commands to get RabbitMQ “username” and “Password”

Credentials:

```
   echo "Username      : user"
   echo "Password      : $(kubectl get secret --namespace default my-release-rabbitmq -o jsonpath="{.data.rabbitmq-password}" | base64 -d)"
   echo "ErLang Cookie : $(kubectl get secret --namespace default my-release-rabbitmq -o jsonpath="{.data.rabbitmq-erlang-cookie}" | base64 -d)"
```

And below information will give you the idea to access RabbitMQ. you will also get once by applying helm install command. These are the commands to get RabbitMQ.

RabbitMQ can be accessed within the cluster on port 5672 at my-release-rabbitmq.default.svc.cluster.local

To access for outside the cluster, perform the following steps:

To Access the RabbitMQ AMQP port:

```
   echo "URL : amqp://127.0.0.1:5672/"
   kubectl port-forward --namespace default svc/my-release-rabbitmq 5672:5672
```

To Access the RabbitMQ Management interface:

```
   echo "URL : http://127.0.0.1:15672/"
   kubectl port-forward --namespace default svc/my-release-rabbitmq 15672:15672
```

Access RabbitMQ Instances publically:

If it need to access the service publically. then you need to shift clusterip service to “loadbalancer”..

Conclusion

You have successfully deployed RabbitMQ in your Kubernetes cluster using Helm. You can now utilize RabbitMQ for message queuing and other distributed messaging tasks.