RFC - Azure Key Vault Integration

Date

Instructions

1. This is instruction section. Don't put this section in your design
2. Only need to fill in the sections you need
3. If the design is big, write a intent-to-RFC first. Intent-to-RFC is a short document to describe the motivation and problem statement.

Authors

- Yong Lik, Chang (theeahlag@gmail.com)

Review Status

Put the names of people who are familiar with the area your design is about.

Reviewers (Tech leads, clients & dependencies that this RFC is going to impact)	Role in the project	Review Status - Intent To RFC (Write LGTM or blocking concerns)	Review Status - RFC (Write LGTM or blocking concerns)
Blair Chen (email)	Tech Lead of the project		
Xiaoyong, Zhu (email)	Tech Lead of the project		

Motivation

Why does the problem need to be solved? What is the business need?

Currently, deploying Feathr on a Kubernetes cluster requires the management of secrets. We would like to add support for deploying Feathr on Kubernetes from Azure Key Vault. The secret can be managed with Azure's third party plugin - CSI Secret Store.

Problem Statement

- **Managing secrets of Azure**

  Provides an alternative method to manage secrets in Azure Key Vault by utilizing Azure's third party plugin - CSI Secret Store. Feathr users can store secrets and keys in Azure Key Vault which can be used by Feathr on Kubernetes.

----------<This is where your Intent To RFC ends>----------

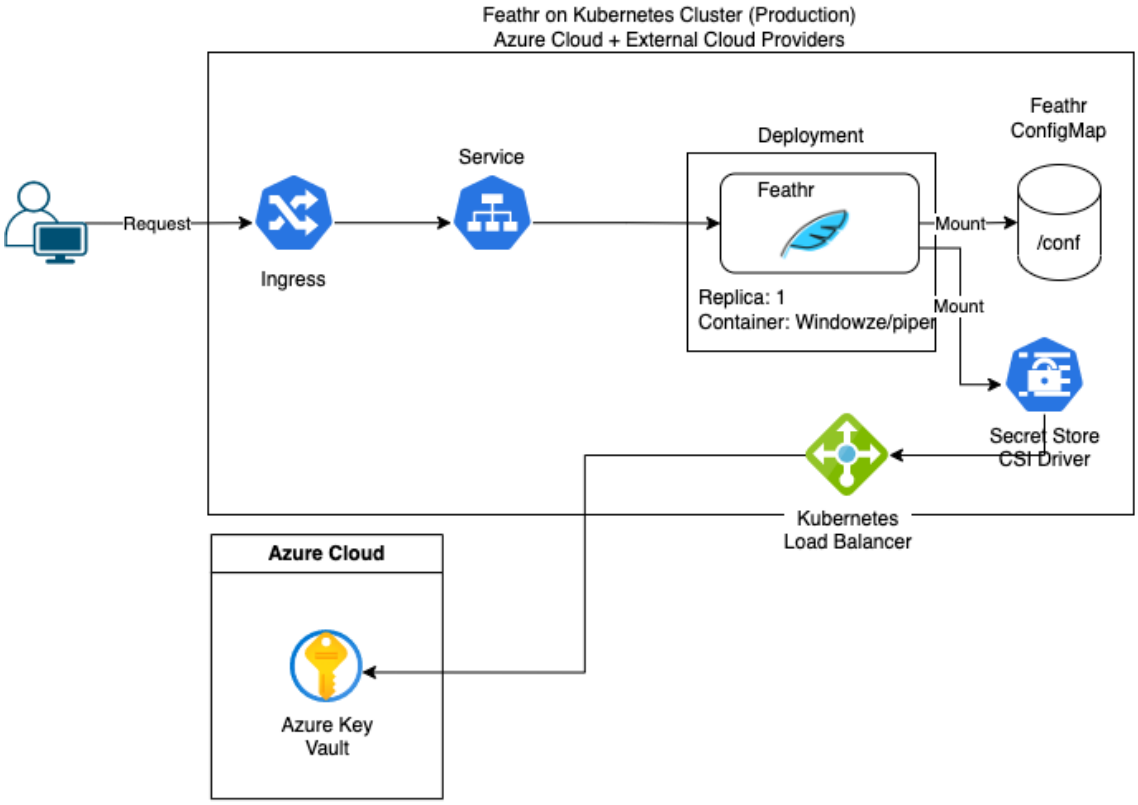
Pre-requisites

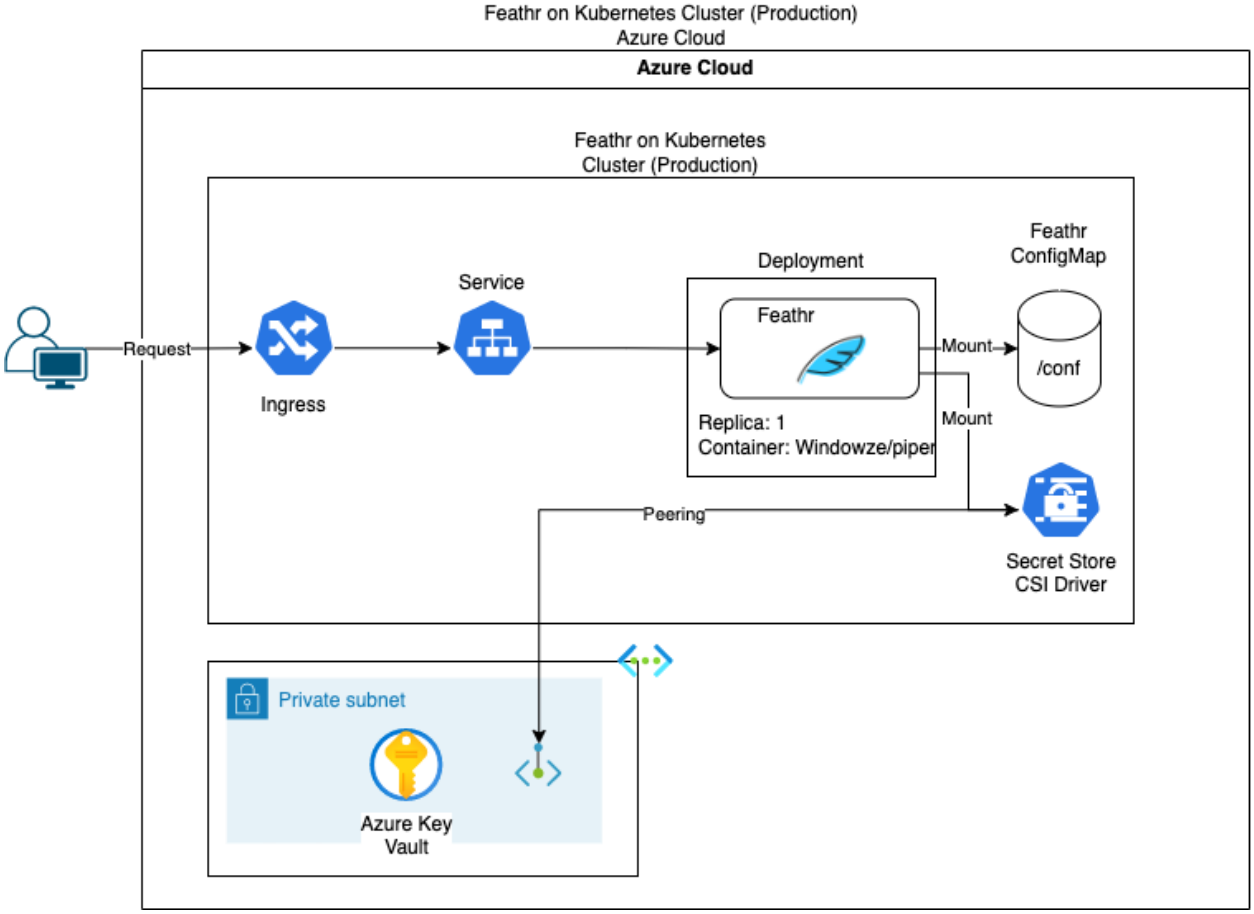
- Azure Key Vault
- CSI Secret Store (Third party driver provided by Azure)

Solution

Providing a helm chart to connect the secret's of Feathr on Kubernetes from Azure Key Vault.

There are two architectures we should consider, which are the deployment of Feathr on Azure Cloud or other cloud providers.


Feathr on Kubernetes Cluster (Production)
Azure Cloud + External Cloud Providers
Request
Ingress
Service
Deployment
Feathr
Replica: 1
Container: Windowze/piper
Mount
Mount
Feathr
ConfigMap
/conf
Secret Store
CSI Driver
Kubernetes
Load Balancer
Azure Cloud
Azure Key
Vault


Feathr on Kubernetes Cluster (Production)
Azure Cloud
Azure Cloud
Feathr on Kubernetes
Cluster (Production)
Request
Ingress
Service
Deployment
Feathr
Replica: 1
Container: Windowze/piper
Mount
Mount
Feathr
ConfigMap
/conf
Peering
Secret Store
CSI Driver
Private subnet
Azure Key
Vault

[Feathr deployment on Public Cloud]

If Feathr is deployed in other cloud providers, we should expect the Key Vault to be made public that only allows a whitelisted IP to access it from Feathr Pod. The Secret Store CSI Driver will be mounted to Feathr Pod. When a pod is deployed, it will make a request to the CSI Driver, the IP address will be resolved within the Kubernetes Cluster's DNS which will then connect to an internal Kubernetes Load Balancer. That request will lastly be forwarded to Azure Key Vault.

[Feathr deployment in Azure Cloud]

If Feathr is deployed in other cloud providers, we can deploy the Key Vault in a private subnet for better security. A private link connection can be made from the Feathr pod to Azure Key Vault after the private subnet of Azure Kubernetes Service and Azure Key Vault are peered if both are in different VNets. If it is in the same VNet, no peering is required. In this case, the connection to Azure Key Vault from Azure Kubernetes Service will be made within the Azure Network Backbone.

Performance, Scalability, Resource Provisioning

Need Discussion

Monitoring, Alerting, SLAs

Need Discussion

Failure Handling / Graceful Degradation

- How to handle failures

Risks

- Azure Key Vault requires secret names to conform to ^[0-9a-zA-Z-]+$. The secrets used in Feathr have underscores in it, e.g. CONN_STR. A secret name mapping should be used, i.e. CONN-STR (Keyvault) = CONN_STR (Feathr Pod) .

Security

Need discussion

Milestones

- Milestones of the project if it's phased.

Milestones	Description
phase1	
phase2	

Appendix

- Supporting information
- Alternative designs and pros/cons analysis
- Experimental data
- Anything that completes the story.