Selecting a Deployment Model

Ned Bellavance

Founder, Ned in the Cloud LLC

@ned1313 | nedinthecloud.com

Overview

Vault deployment models

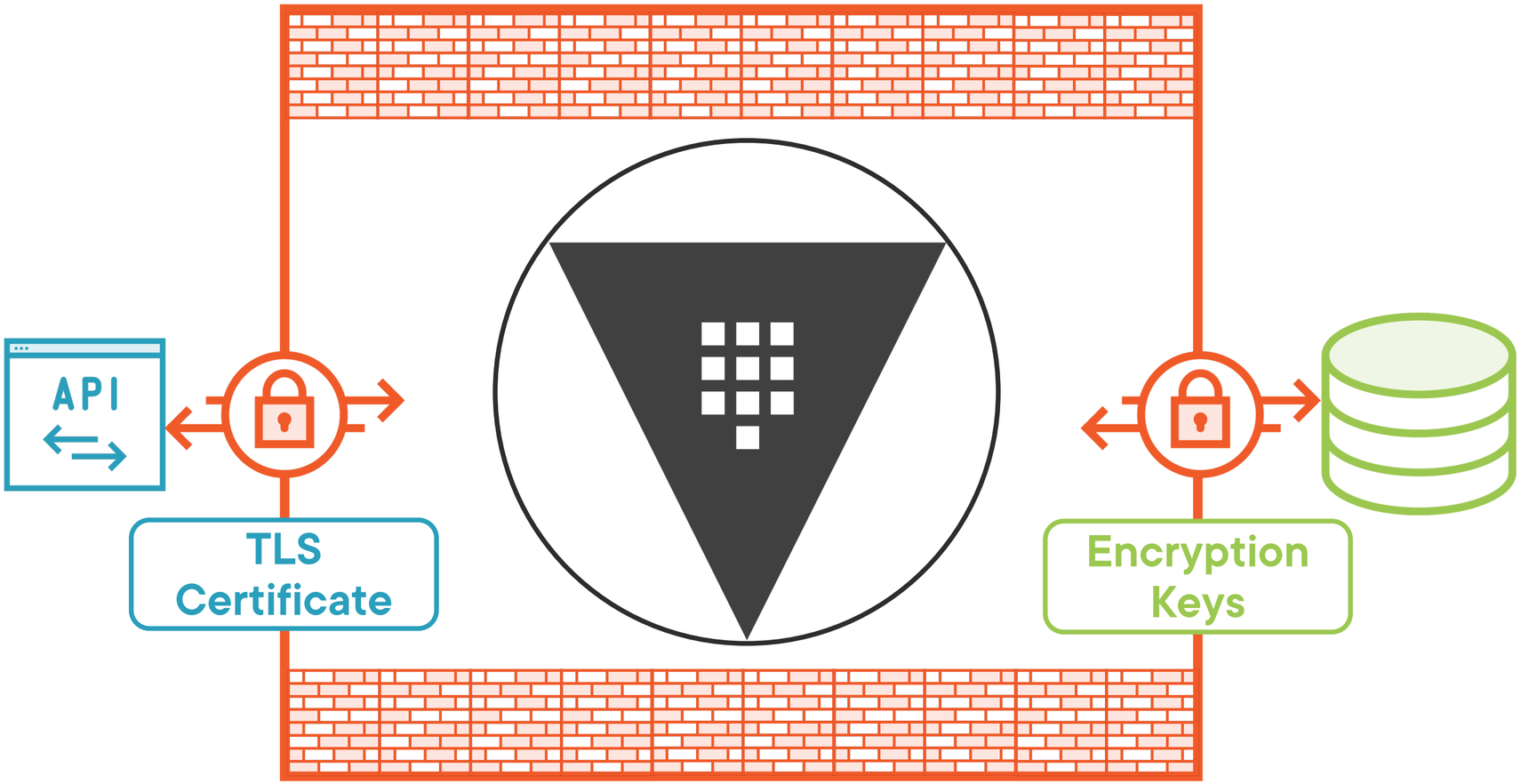
Configuration options

Scenario requirements

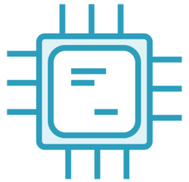
Deployment design

Vault Architecture and Deployment Models

Vault Logical Architecture

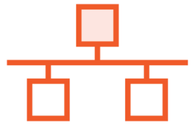


Deployment Components



Bare Metal / VM / Container

Multiple Operating Systems



Client and storage communication

Load balancer or DNS

HashiCorp or community support

High availability support

API TLS certificate

Storage backend traffic

Deployment Considerations

Service level agreement and uptime

Component failure

Health monitoring

Capacity monitoring

Key shares

Server configuration

Storage backend

Distributed key shares

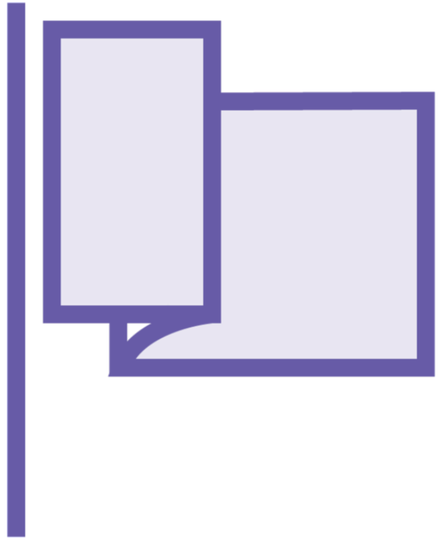
Auto unseal

Configuration Options

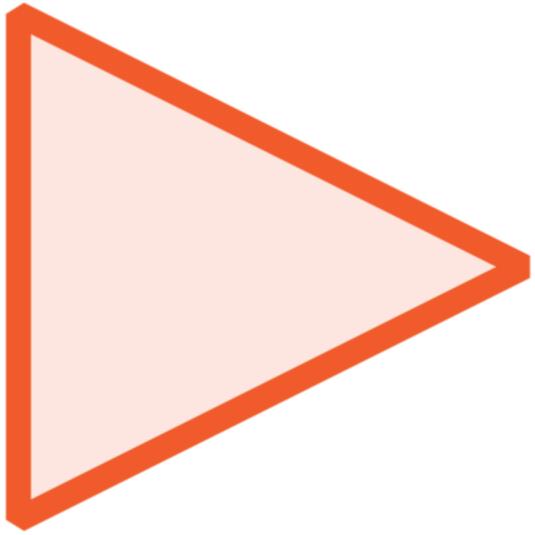
Vault Server Configuration

Defined in HCL or JSON

Supports multiple files



Passed using config flag



Loaded when service starts

Parameter Categories

Single value

Listener

Storage

Seal

Telemetry

Service registration

Vault-Config.hcl

```
# General Settings

ui = [true | false]

disable_mlock = [true | false]

log_level = "level"

log_format = ["standard" | "json" ]

max_lease_ttl = "768h"

default_lease_ttl = "768h"

cluster_addr = "https://address:port"

api_addr = "https://address:port"
```

Listener Parameters

Address information

HTTP timeouts

Request control

Proxy behavior

TLS settings

X-Forwarded-For

Vault-Config.hcl

```
# Listener Settings

listener "tcp" {
 # Listener address
 address             = "0.0.0.0:8200"
 cluster_address     = "0.0.0.0:8201"

 # TLS settings
 tls_disable         = 0
 tls_cert_file       = "/opt/vault/tls/vault-full.pem"
 tls_key_file        = "/opt/vault/tls/vault-key.pem"
 tls_min_version     = "tls12"
}
```

Storage Backend

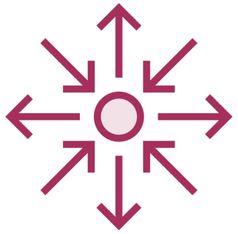
Storage types

- Object
- Database
- Key/Value
- File
- Memory

Integrated Storage (Raft)

- Local storage
- Highly available
- Replicated

Support



High availability

Storage configuration

Vault-Config.hcl

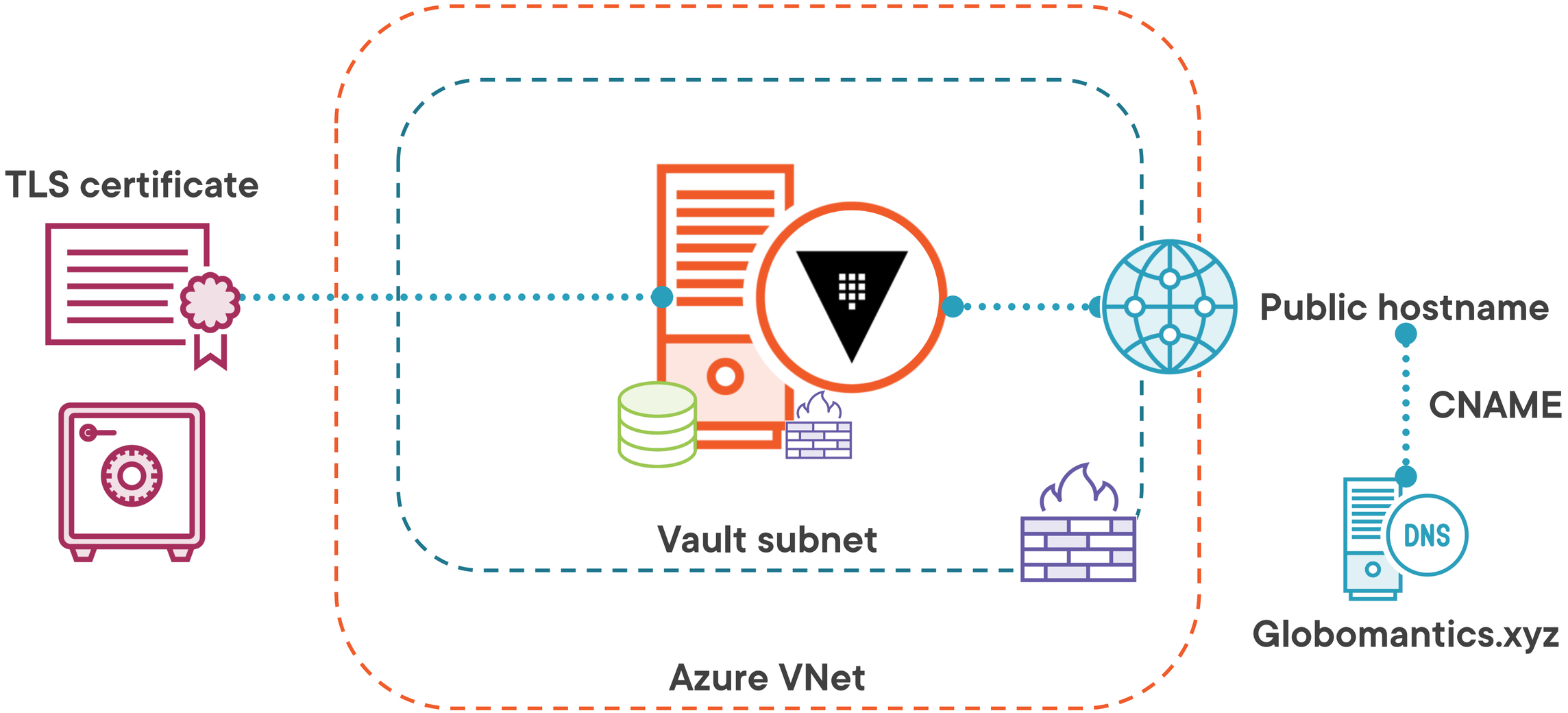
```
# Storage Settings
storage "consul" {
  address = "127.0.0.1:8500"
  path = "vault"
}

storage "raft" {
  path    = "/opt/vault/data"
  node_id = "vault-0"

  retry_join {
    leader_tls_servername = "vault-0.local"
    leader_api_addr = "https://vault-0.local:8200"
    leader_ca_cert_file = "/opt/vault/tls/vault-ca.pem"
    leader_client_cert_file = "/opt/vault/tls/vault-cert.pem"
    leader_client_key_file = "/opt/vault/tls/vault-key.pem"
  }
}
```

Deployment Design

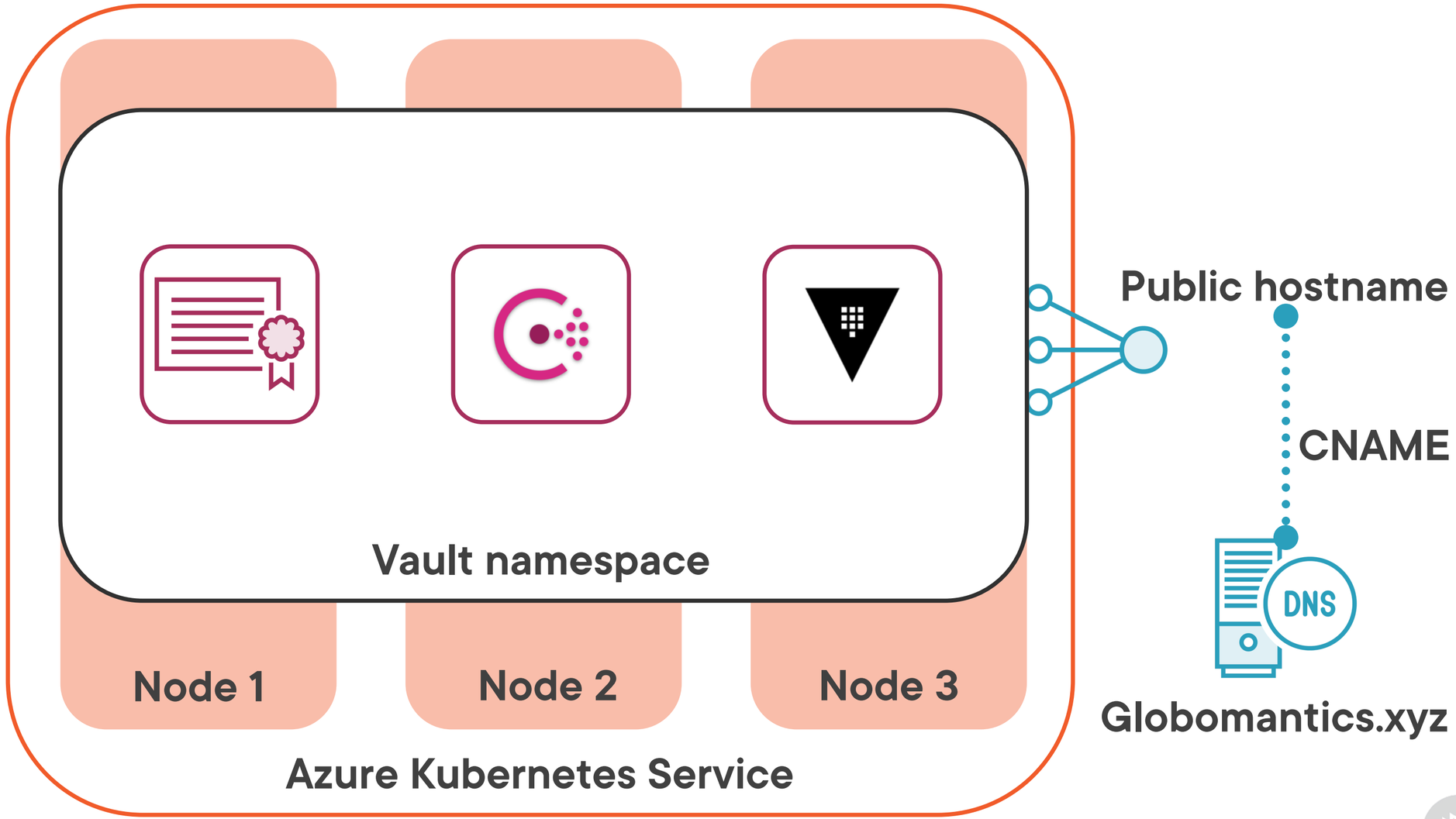
Globomantics Requirements

- **Deploy in Azure**
- **Publicly available endpoint**
- **Use third-party certificates**
- **SLA of 99.99% for Vault**
- **Auto unseal of Vault**

Azure VMs Deployment



Azure Kubernetes Service Deployment



Module Summary

Vault deployment depends on requirements

Vault configuration is defined by HCL or JSON files

Listener controls how Vault receives requests

Storage determines where data is stored

Up Next: Deploying Vault Server