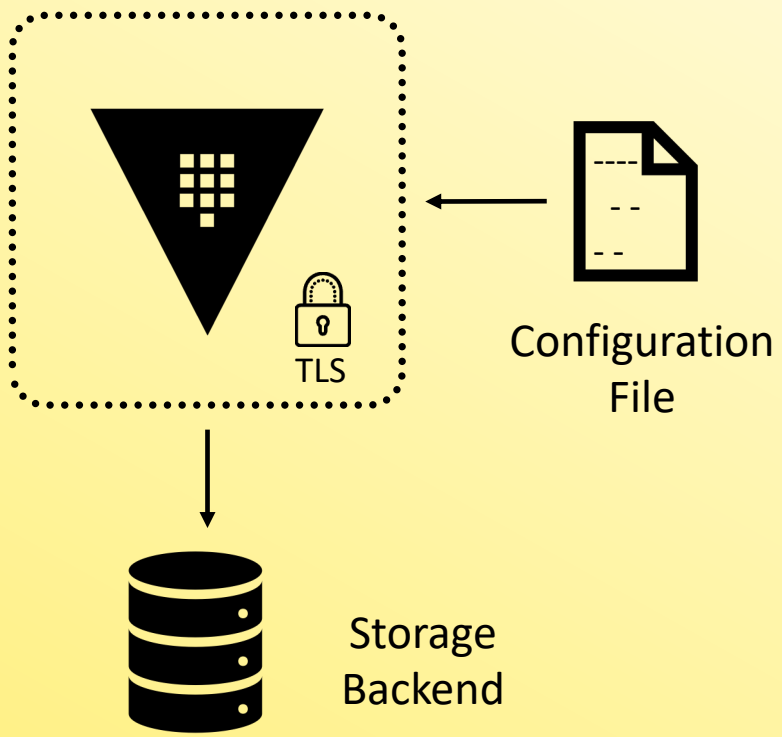
Configure a Highly Available [HA] Cluster

Single-Node Vault Server



Not a Recommended Architecture

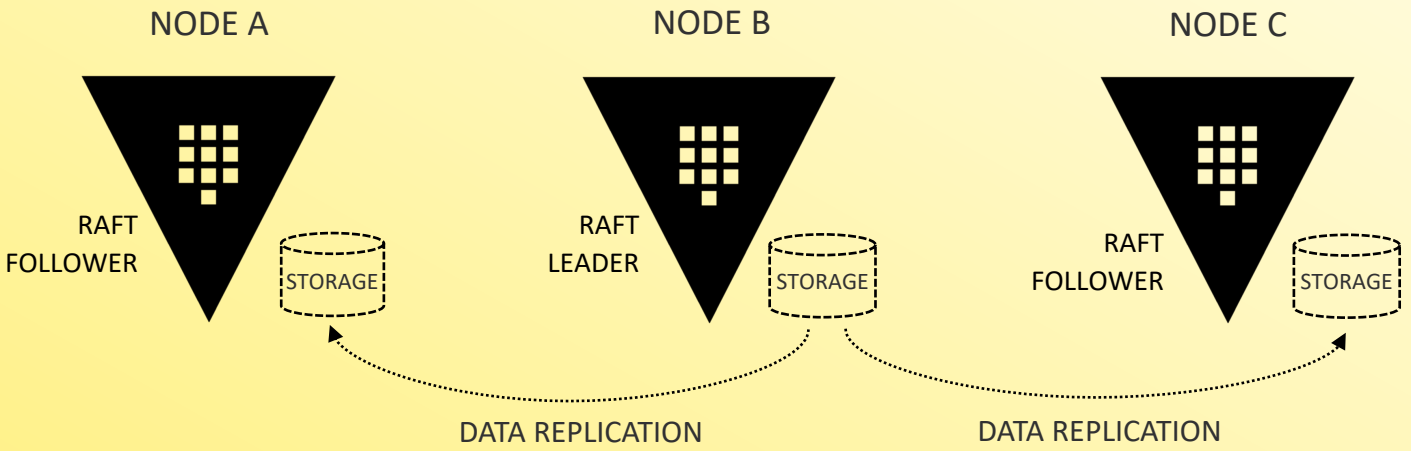
- No redundancy
- No scalability
- No failure tolerance

What Should a Cluster Look Like?

- Ideally, we want something that provides redundancy, failure tolerance, scalability, and a fully replicated architecture
- For Vault Enterprise, you are limited to either Integrated Storage or Consul storage backends
- HashiCorp (and consultants like me) are moving away from Consul as the primary storage backend and using Integrated Storage for everything
- The Vault Operations Professional exam will NOT feature Consul as a configuration or deployment option

Multi-Node Cluster using Integrated Storage

- Integrated Storage (aka Raft) allows Vault nodes to provide its own replicated storage across the Vault nodes within a cluster
- Define a local path to store replicated data
- All data is replicated among all nodes in the cluster



How Do I Configure Integrated Storage?

- Initial configuration of Integrated Storage is done in the Vault configuration file
- Multiple ways to join nodes to create a Vault cluster in the configuration file….or you do it manually
- Use `retry_join` stanza to automate the creation of the cluster from participating Vault nodes

Terminal

```
storage "raft" {
  path    = "/opt/vault/data"
  node_id = "node-a.hcvop.com"
  retry_join {
    auto_join = "provider=aws region=us-east-1 tag_key=vault tag_value=east-1"
  }
}
listener "tcp" {
 address = "0.0.0.0:8200"
 cluster_address = "0.0.0.0:8201"
 tls_disable = 0
}
seal "awskms" {
  region = "us-east-1"
  kms_key_id = "12345678-abcd-1234-abcd-123456789101",
}
api_addr = "https://vault.hcvop.com:8200"
cluster_addr = " https://node-a.hcvop.com:8201"
cluster_name = "vault-prod-us-east-1"
ui = true
log_level = "INFO"
```

How Do I Configure Integrated Storage?

- **`path`** = the filesystem path where all the Vault data will be stored
- **`node_id`** = the identifier for the node in the cluster – cannot be duplicated within a cluster
- **`retry_join`** [optional] – automatically join the listed nodes to create a cluster

Terminal

```
storage "raft" {
  path    = "/opt/vault/data"
  node_id = "node-a.hcvop.com"
  retry_join {
    auto_join = "provider=aws region=us-east-1 tag_key=vault tag_value=east-1"
  }
}
listener "tcp" {
 address = "0.0.0.0:8200"
 cluster_address = "0.0.0.0:8201"
 tls_disable = 0
}
seal "awskms" {
  region = "us-east-1"
  kms_key_id = "12345678-abcd-1234-abcd-123456789101",
}
api_addr = "https://vault.hcvop.com:8200"
cluster_addr = " https://node-a.hcvop.com:8201"
cluster_name = "vault-prod-us-east-1"
ui = true
log_level = "INFO"
```

Configure Integrated Storage in the Vault Configuration File

Each `retry_join` stanza can include DNS names or IP addresses and the port

Terminal

```
storage "raft" {
  path    = "/opt/vault/data"
  node_id = "node-a.hcvop.com"
  retry_join {
    leader_api_addr = "https://node-b.hcvop.com:8200"
  }
  retry_join {
    leader_api_addr = "https://node-c.hcvop.com:8200"
  }
  retry_join {
    leader_api_addr = "https://node-d.hcvop.com:8200"
  }
  retry_join {
    leader_api_addr = "https://node-e.hcvop.com:8200"
  }
}
```

Multiple retry_join stanzas

Configure Integrated Storage in the Vault Configuration File

Using `auto_join` to discover other Vault nodes using tags

Terminal

```
storage "raft" {
  path    = "/opt/vault/data"
  node_id = "node-a.hcvop.com"
  retry_join {
    auto_join = "provider=aws region=us-east-1 tag_key=vault tag_value=east-1"
}}
```

What cloud/provider are you using?

What region should Vault look at to find tags?

The tag key that Vault should search for

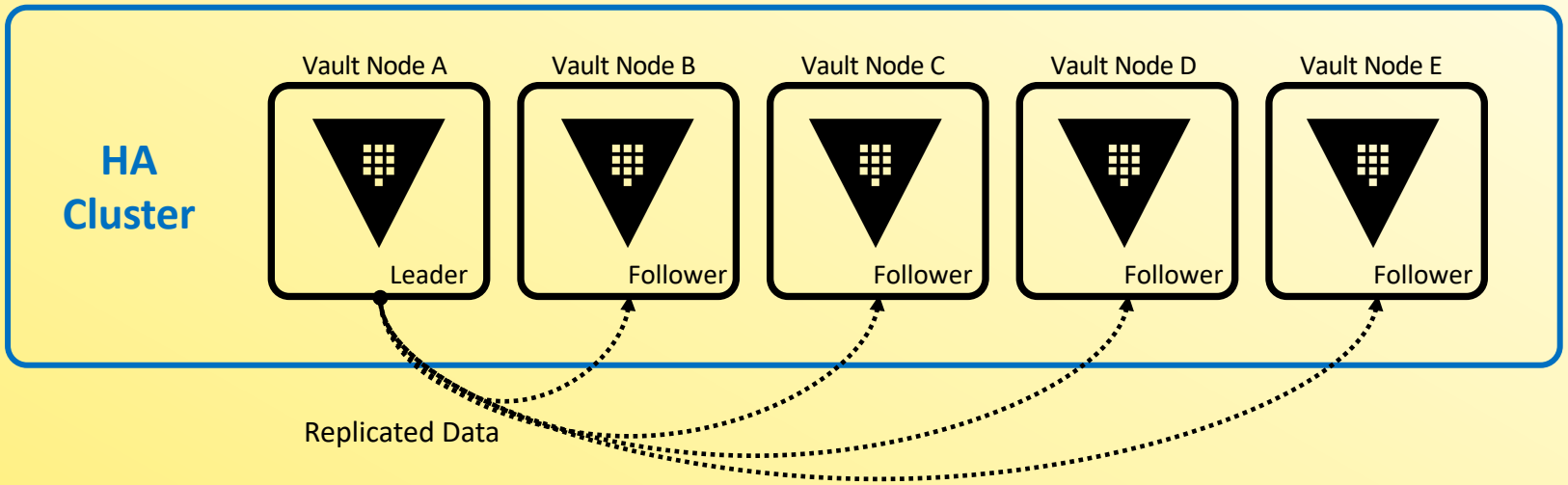
The tag value that Vault should search for

Join Nodes to Form a Cluster

Manually join standby nodes to the cluster using the CLI:

Terminal

```
$ vault operator raft join https://active_node.example.com:8200
```

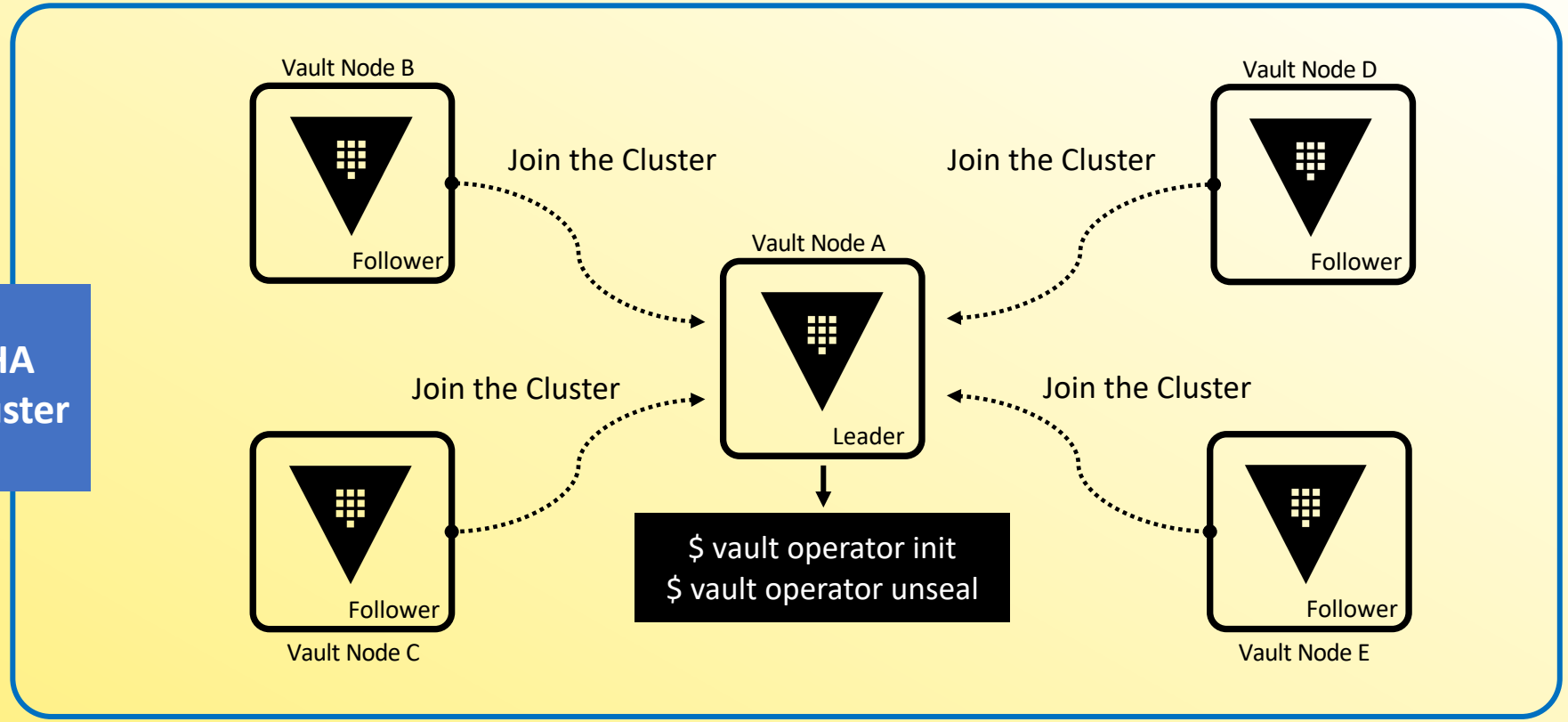


Managing Integrated Storage via CLI

Use the `vault operator raft` command

`list-peers`	Returns the raft cluster member information
`join`	Joins a node to the cluster
`remove-peer`	Removes a node from the cluster
`snapshot`	Restores and saves snapshots from the cluster

Manual Cluster Configuration Workflow



Viewing Cluster Information

List the cluster members - determine which node is the leader

- Note: You must be authenticated (client token) to run this command

Terminal

```
$ vault operator raft list-peers

Node       Address             State       Voter
----       -------             -----       -----
node-a     10.0.101.22:8201    leader      true
node-b     10.0.101.23:8201    follower    true
node-c     10.0.101.24:8201    follower    true
node-d     10.0.101.25:8201    follower    true
node-e     10.0.101.26:8201    follower    true
```

Remove a Node from the Cluster

Name of the node to be removed

Terminal

```
$ vault operator raft remove-peer node-e
Peer removed successfully!


$ vault operator raft list-peers

Node       Address             State       Voter
----       -------             -----       -----
node-a     10.0.101.22:8201    leader      true
node-b     10.0.101.23:8201    follower    true
node-c     10.0.101.24:8201    follower    true
node-d     10.0.101.25:8201    follower    true
```