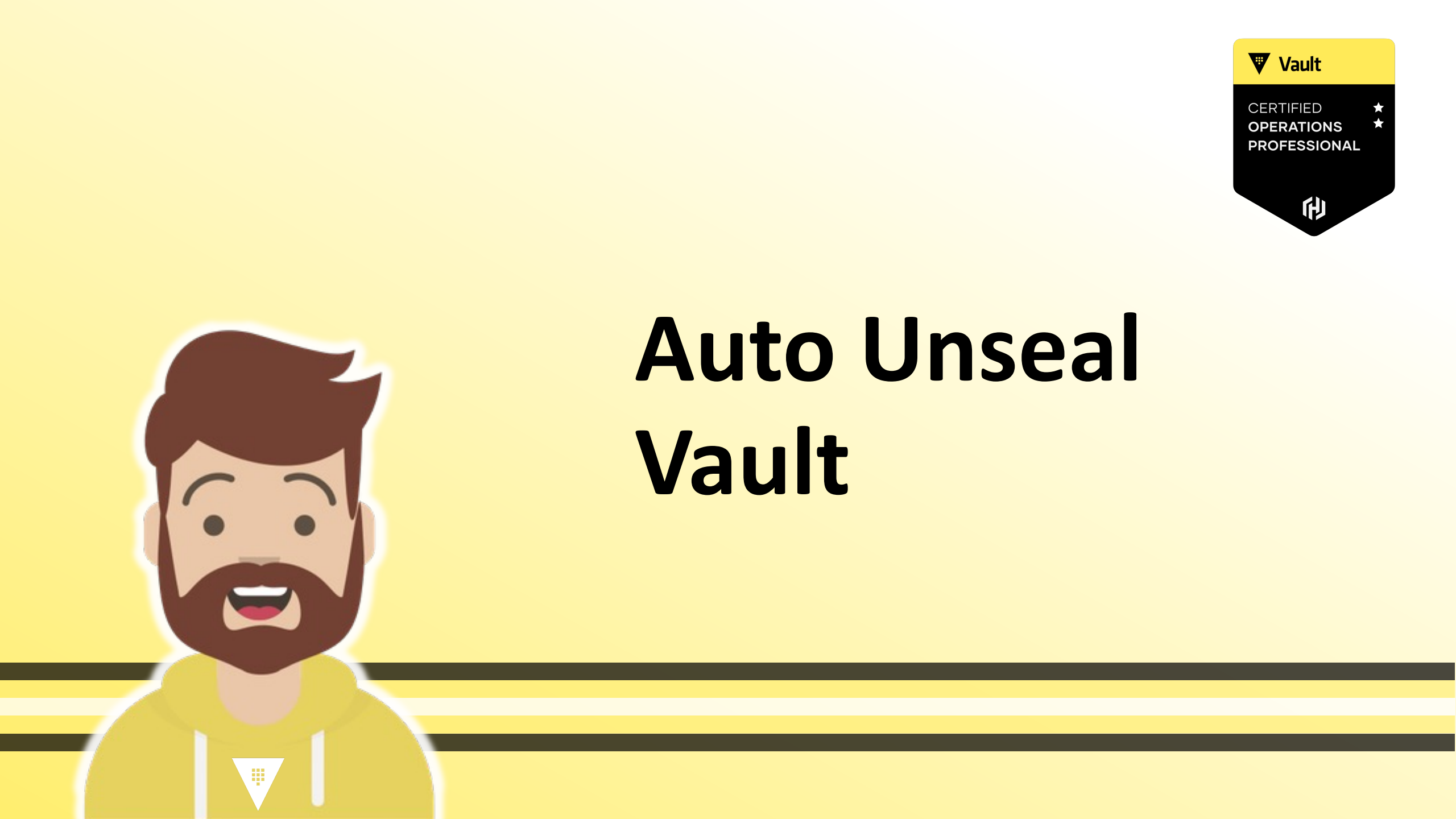
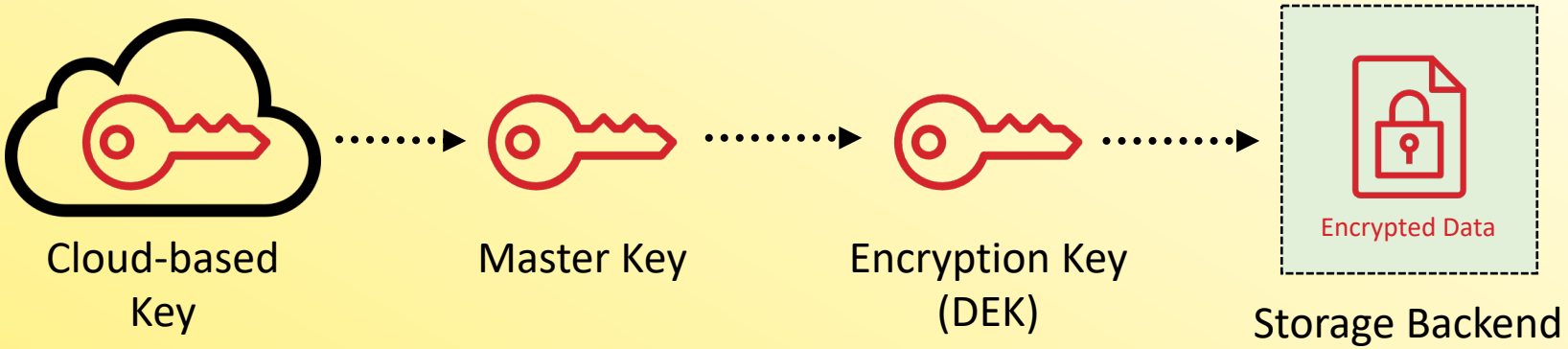
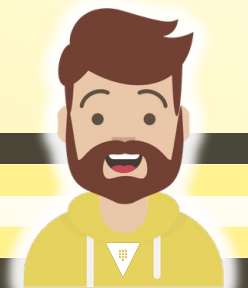


Auto Unseal Vault

What is Auto Unseal?

- Rather than use shared keys (unseal keys), auto unseal can use a trusted cloud-based key to protect the master key





Supported Auto Unseal Mechanisms

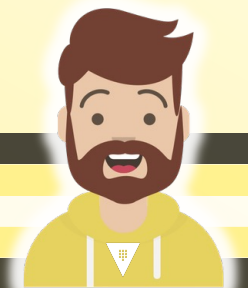
- Supported Services include:
  - AWS KMS
  - Azure Key Vault
  - GCP Cloud KMS
  - AliCloud KMS
  - OCI KMS
  - HSM *(Enterprise Only)*

  - Transit Secrets Engine from *another* Vault cluster can also be used to unseal

How Auto Unseal Works

- Vault does *not* write anything to the cloud-based service
- The master key is encrypted with the cloud-based key and stored on the storage backend – stored at path/core/master
- It uses the cloud-based key to decrypt the master key during the unseal process

- This is different from a Vault cluster not using auto unseal and is using Shamir (unseal keys) - where the master key is *never* written to persistent storage



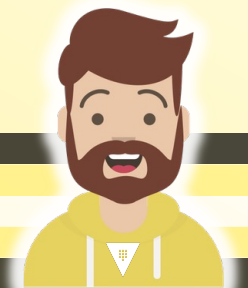
Design Considerations

- Vault does support rotation for cloud-based keys
- Remember that some cloud-based KMS services are regional, so if an entire region goes down, the KMS key will not be available to unseal
- During normal operations, the Vault cluster does not communicate with the auto unseal key service for unseal operations
  - However, recent versions of Vault have introduced a "health check" where Vault validates access to the key periodically
- Health Check will test the health of the auto-unseal backend once every 10 minutes.
  - If unhealthy, logs a warning on the condition and begin testing every one minute until healthy again.

Design Considerations

Service downtime is only a problem if your Vault cluster is restarted or sealed and needs to be unsealed during that time

- To avoid this problem, many folks will create a key outside of the cloud-based service and import it to multiple regions for a fail-safe
- Clusters used for Transit auto unseal need to be highly available to prevent this from happening to the cluster it supports
- Remember that *some* cloud-based services now offer multi-region keys to provide built-in regional high-availability



How to Enable Auto Unseal

Configured in the Vault configuration file

*Configuration truncated

```
...
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",
  endpoint = "example.kms.us-east-1.vpce.amazonaws.com"
}
api_addr = "https://vault-us-east-1.hcvop.com:8200"
cluster_addr = " https://node-a-us-east-1.hcvop.com:8201"
cluster_name = "vault-prod-us-east-1"
ui = true
log_level = "INFO"
```

How to Enable Auto Unseal

1. Add a seal stanza to your configuration.
2. Choose what service to use
3. Configure the parameters based on the service

Options include:

- alicloudkms
- awskms
- azurekeyvault
- gcpckms
- ocikms
- pkcs11
- transit

```
seal  "type" {
  parameter = "value"
  parameter = "value"
  parameter = "value"
}
```

Auto Unseal with AWS KMS

- Set the seal type to `awskms`
- Identify the KMS Key ID (ARN) for the key Vault will use
- Declare the region for the key
- Set the VPC Endpoint (if being used)

- Set the AWS credentials

```
seal "awskms" {
  region = "us-east-1"
  kms_key_id = "arn:aws:kms:us-east-1:12345678:key/abcd"
  endpoint = "example.kms.us-east-1.vpce.amazonaws.com"
  access_key = "AKIAIOSFODNN7EXAMPLE"
  secret_key = "wJalrXUtnFEMI/K7Mexample5"
}
```

Be careful with adding credentials to a clear text file:

- If hosting in AWS, use an IAM role
- If hosting on-prem, use environment variables

- AWS role needs kms:Encrypt, kms:Decrypt, and kms:Describe for the KMS key

Auto Unseal with Azure Key Vault

- Set the seal type to `azurekeyvault`
- Identify the Key Vault Name
- Set the Key Name for HashiCorp Vault to use

- Set the Azure credentials

```
seal "azurekeyvault" {
  vault_name =  "vault-hashicorp"
  key_name =  "hashicorp-vault-key"
  tenant_id = "8427464-8963-6422-example"
  client_id = "03dc33fc-16d9-example"
  client_secret = "DKEMCI8….."
}
```

Be careful with adding credentials to a clear text file:

- If hosting in Azure, use Managed Service Identities
- If hosting on-prem, use environment variables

Auto Unseal with GCP KMS

- Set the seal type to “gcpckms”
- Identify the GCP Project ID to use
- Set the region where the key ring lives
- Set the key ring to use
- Identify the crypto key that Vault will use

- Set the GCP credentials

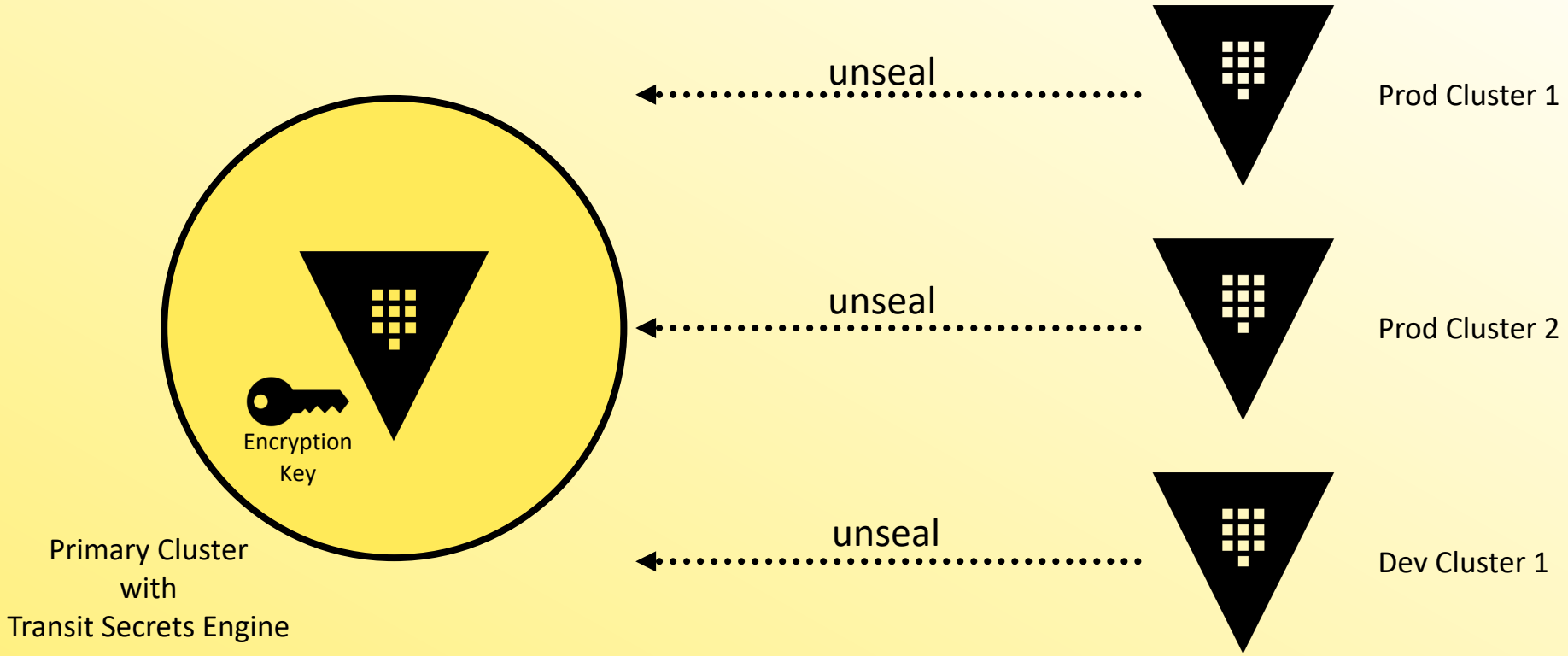
```
seal “gcpckms" {
  project =  "vault-project"
  region =  "global"
  key_ring = "hashicorp-vault-keyring"
  crypto_key = "hashicorp-vault-key"
  credentials = "/usr/gcp.json"
}
```

Be careful with adding credentials to a clear text file:

- If hosting in GCP, set the instance's service account with Cloud KMS role
- If hosting on-prem, use environment variables

Auto Unseal with Transit

- Using *another* Vault cluster, configure the transit secrets engine and create a key to be used by Vault



Auto Unseal with Transit

- Set the seal type to `transit`
- Identify the address of the primary cluster
- Set the key name
- Set the mount path and namespace (if applicable)

- Set the credentials (token)

```
seal "transit" {
  address =  "https://vault.hcvop.com:8200"
  token = "s.Qf1s5zigZ4OX6akYexample"
  key_name = "auto-unseal-key"
  mount_path = "/transit"
  tls_ca_cert = "/etc/vault/ca.pem"
  tls_client_cert = "/etc/vault/client.pem"
  tls_client_key = "/etc/vault/key.pem"
}
```

Be careful with adding credentials to a clear text file:

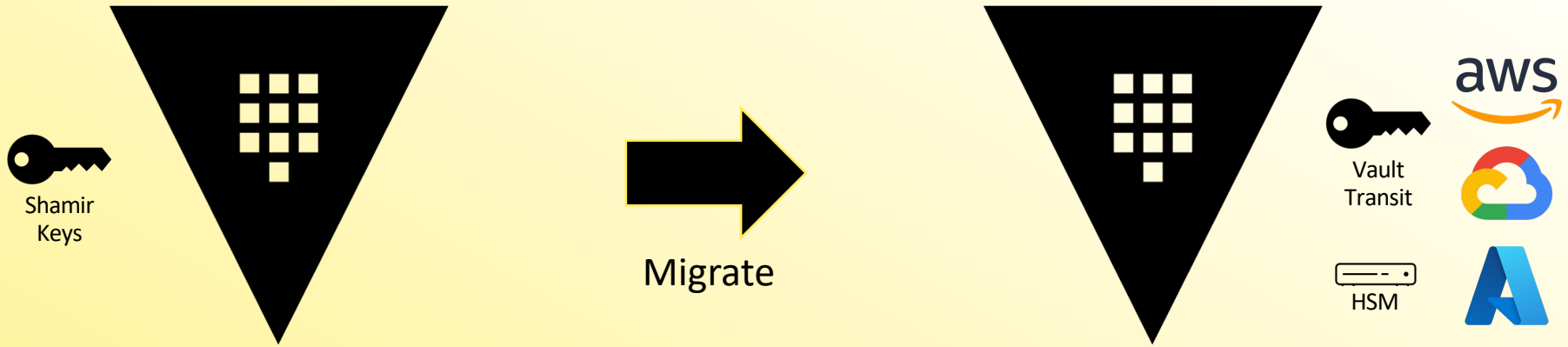
- Use environment variable VAULT_TOKEN
- The token needs "update" capability for *transit/encrypt/<key>* and *transit/decrypt/<key>*

Other Options Not Covered Here…

- **HSM**
  - Follows a similar process of using a key stored in a trusted HSM.
  - Vault uses the key to decrypt the master key
  - HSM integration also enables Seal Wrapping as well to provide FIPS 140-2 compliance
- **AliCloud**
- **Oracle Cloud**

Seal Migration

Seal Migration



Vault Cluster

Shamir Keys

- Manual Unseal
- Risk of Losing Keys
- Not Automated

Vault Cluster

Auto Unseal

- Automatic Unseal
- Protected by trusted KMS
- No Reliance on Humans

Seal Migration

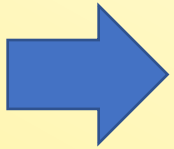
- You can migrate from one seal type to another if needed
- Unfortunately, it requires downtime for Vault since you must restart the service
- This is not a process that you will perform very often, if at all…

- Examples of migrations:
  - Shamir → AWS KMS
  - GCP Cloud KMS → Azure Key Vault
  - AWS KMS → AWS KMS
  - Azure Key Vault → HSM

Shamir to Auto Unseal Migration

Shamir clusters don't have a seal configuration because it's the default configuration

```
...
listener "tcp" {
 address = "0.0.0.0:8200"
 cluster_address = "0.0.0.0:8201"
 tls_disable = 0
...
}
api_addr = "https://vault-us-east-1.hcvop.com:8200"
cluster_addr = " https://node-a-us-east-1.hcvop.com:8201"
cluster_name = "vault-prod-us-east-1"
ui = true
log_level = "INFO"
```



…but auto unseal does, so the first step is to add the new configuration to a standby node (or the only node)

```
...
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-us-east-1.hcvop.com:8200"
cluster_addr = " https://node-a-us-east-1.hcvop.com:8201"
cluster_name = "vault-prod-us-east-1"
ui = true
log_level = "INFO"
```

Shamir to Auto Unseal Migration

Restart the Vault service

```
# Restart the Vault service
$ systemctl restart vault.service



# View the log file to see a new log entry

2022-12-25T10:47:09.295-0400 [WARN]  core: entering seal migration mode; Vault will not
automatically unseal even if using an autoseal: from_barrier_type=shamir to_barrier_type=awskms
```

Shamir to Auto Unseal Migration

Unseal Vault using the `-migrate` flag

```
# Unseal Vault using the migrate flag
$ vault operator unseal -migrate
Unseal Key (will be hidden): <enter key here>

Key                            Value
---                            -----
Recovery Seal Type             shamir
Initialized                    true
Sealed                         false
Total Recovery Shares          5
Threshold                      3
Seal Migration in Progress     true
Version                        1.10.0+ent
Storage Type                   raft
Cluster Name                   vault-prod-us-east-1
Cluster ID                     fa1b-4ccb17a1ecac
HA Enabled                     true
HA Cluster                     n/a
HA Mode                        standby
Active Node Address            <none>
Raft Committed Index           65
Raft Applied Index             65
```

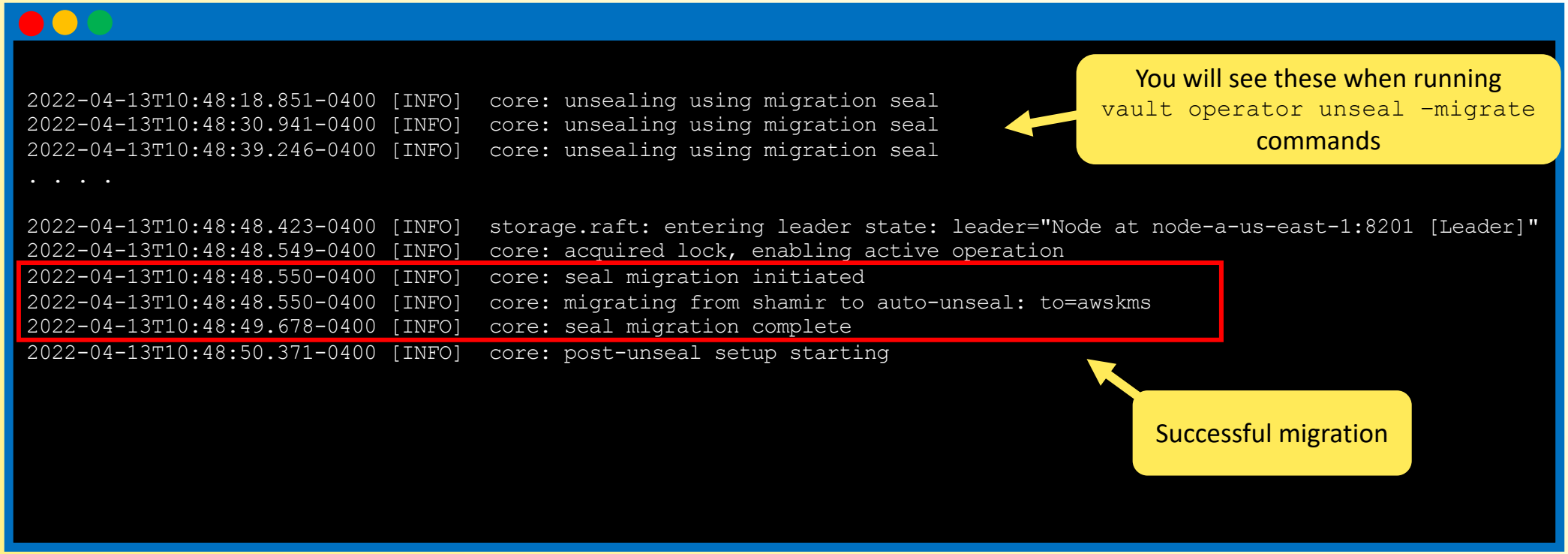
Do this multiple times to meet key threshold

Note: If the key threshold is 3, then you will need to run this command 3 times.

If it is set to 1, *like you'll find in the exam*, then you'll just need to run it 1 time

Shamir to Auto Unseal Migration

Viewing the Logs During Migration Steps



Shamir to Auto Unseal Migration

Order of Operations

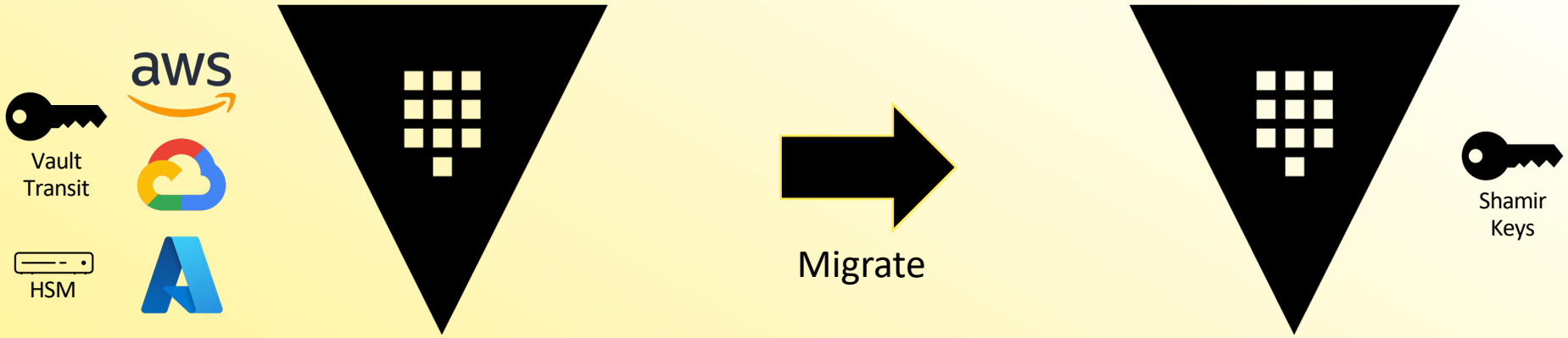
1. Perform unseal migration on standby node 1
2. Perform unseal migration on standby node 2
3. Run `vault operator step-down` on the Leader node
4. Perform unseal migration on the last node (previous leader)

Standby Node

Leader Node

Standby Node

Auto Unseal to Shamir Migration



Vault Cluster

Auto Unseal

- Automatic Unseal
- Protected by trusted KMS
- No Reliance on Humans

Vault Cluster

Shamir Keys

- Manual Unseal
- Risk of Losing Keys
- Not Automated

Auto Unseal to Shamir Migration

- Update the seal stanza to include `disabled = true`
  - This allows Vault to decrypt the key, but it will NOT use it for unseal operations

- Follow the same process as previously discussed:
  - Restart the Vault service
  - Run `vault operator unseal -migrate`
    - Provide RECOVERY keys to perform the migration
    - Recovery keys will be migrated to be used as unseal keys moving forward

Auto Unseal to Shamir Migration

Auto Unseal configuration on an auto unsealed cluster needs to be modified…

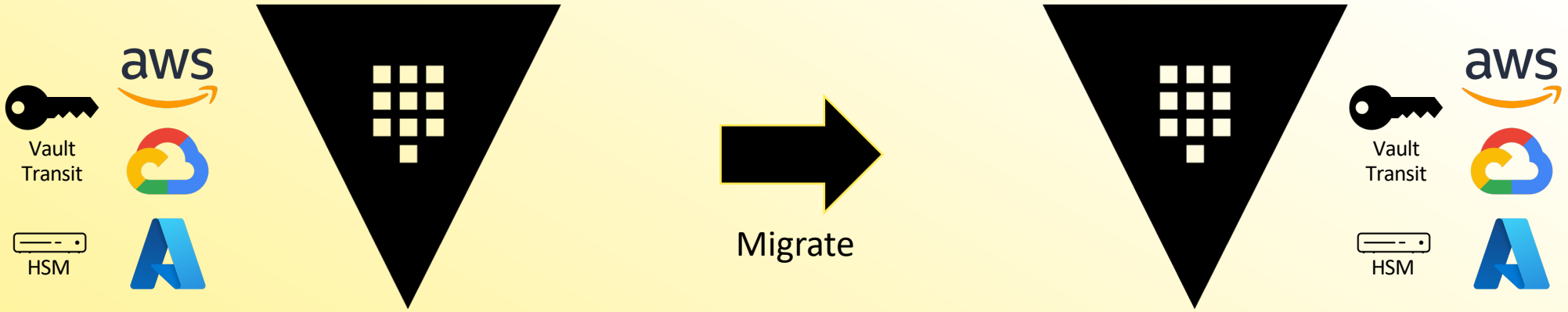
```
...
listener "tcp" {
 address = "0.0.0.0:8200"
 cluster_address = "0.0.0.0:8201"
 tls_disable = 0
...
}
seal "awskms" {
  region = "us-east-1"
  kms_key_id = "12345678-abcd-1234-abcd-123456789101"
}
api_addr = "https://vault-us-east-1.hcvop.com:8200"
cluster_addr = " https://node-a-us-east-1.hcvop.com:8201"
cluster_name = "vault-prod-us-east-1"
ui = true
log_level = "INFO"
```

…add the `disabled =true` configuration in the seal stanza as the first step for migration

```
...
listener "tcp" {
 address = "0.0.0.0:8200"
 cluster_address = "0.0.0.0:8201"
 tls_disable = 0
...
}
seal "awskms" {
  disabled = true
  region = "us-east-1"
  kms_key_id = "12345678-abcd-1234-abcd-123456789101"
}
api_addr = "https://vault-us-east-1.hcvop.com:8200"
cluster_addr = " https://node-a-us-east-1.hcvop.com:8201"
cluster_name = "vault-prod-us-east-1"
ui = true
log_level = "INFO"
```

Auto Unseal to Auto Unseal Migration

Including Changing Keys when using the Same Auto Unseal



Vault Cluster

Auto Unseal

- Automatic Unseal
- Protected by trusted KMS
- No Reliance on Humans

Vault Cluster

Auto Unseal

Automatic Unseal
Protected by trusted KMS
No Reliance on Humans

Auto Unseal to Auto Unseal

1. Update the **original** seal stanza to include `disabled = true`
    - This allows Vault to decrypt the key, but it will NOT use it for unseal operations
2. Add the NEW stanza to the configuration
3. Follow the same process as previously discussed:
    - Restart the Vault service
    - Run `vault operator unseal -migrate`
        - Provide RECOVERY keys to perform the migration
        - Recovery keys will be migrated to be used as unseal keys moving forward

Auto Unseal to Shamir Migration

Auto Unseal configuration on an auto unsealed cluster needs to be modified……add the `disabled =true` configuration in the seal stanza as the first step for migration

```
...
listener "tcp" {
 address = "0.0.0.0:8200"
 cluster_address = "0.0.0.0:8201"
 tls_disable = 0
...
}
seal "awskms" {
  disabled = true
  region = "us-east-1"
  kms_key_id = "12345678-abcd-1234-abcd-123456789101"
}
api_addr = "https://vault-us-east-1.hcvop.com:8200"
cluster_addr = "https://node-a-us-east-1.hcvop.com:8201"
cluster_name = "vault-prod-us-east-1"
ui = true
log_level = "INFO"
```

…add the new seal configuration as a second stanza to the configuration file

```
...
listener "tcp" {
 address = "0.0.0.0:8200"
 cluster_address = "0.0.0.0:8201"
 tls_disable = 0
...
}
seal "awskms" {
  disabled = true
  region = "us-east-1"
  kms_key_id = "12345678-abcd-1234-abcd-123456789101"
}
seal "awskms" {
  region = "us-east-1"
  kms_key_id = "987654321-dcba-4321-dcba-10987654321"
}
api_addr = "https://vault-us-east-1.hcvop.com:8200"
cluster_addr = "https://node-a-us-east-1.hcvop.com:82
cluster_name = "vault-prod-us-east-1"
ui = true
log_level = "INFO"
```