Use Batch Tokens

Introduction to Batch Tokens

Batch tokens are encrypted binary large objects (blobs)

- Designed to be lightweight & scalable
- They are NOT persisted to storage, but they are not fully-featured
- Ideal for high-volume operations
- Can be used for DR Replication cluster promotion as well
- Includes information such as policy, TTL, and other attributes

- Batch tokens are encrypted using the barrier key, which is why they can be used across all clusters within the replica set

Compare Batch Tokens vs. Service Tokens

	Service Tokens	Batch Tokens
Can be root tokens	Yes	No
Can create child tokens	Yes	No
Renewable	Yes	No
Listable	Yes	No
Manually Revocable	Yes	No
Can be periodic	Yes	No
Can have explicit Max TTL	Yes	No (always uses a fixed TTL)
Has accessors	Yes	No
Has Cubbyhole	Yes	No
Revoked with parent (if not orphan)	Yes	Stops Working
Dynamic secrets lease assignment	Self	Parent (if not orphan)

Know these differences very well

Compare Batch Tokens vs. Service Tokens

	Service Tokens	Batch Tokens
Can be used across Performance Replication clusters	No	Yes (if orphan)
Creation scales with performance standby node count	No	Yes
Cost	Heavyweight; multiple storage writes per token creation	Lightweight; no storage cost for token creation

Know these differences very well

Identifying Token Types in Vault via Prefix

Token Type	Vault 1.9.x or earlier	Vault 1.10 and later
Service tokens	s.	hvs.
Batch tokens	b.	hvb.
Recovery tokens	r.	hvr.

Identifying Token Types in Vault via Prefix

hvs.

Service Token

hvb.

Batch Token

hvr.

Recovery Token

Token Size

Service Token ⟶

Size: ~98 bytes

hvs.CAESIA4CZQisJNn9eq3g5TS5xP0-
DPkFDsshIi_jb5UH28AuGiAKHGh2cy5wZjl
PU1NsVlpWaTQxSFUyczFuQk9DOFgQHQ

Batch Token ⟶

Size: ~128 bytes

hvb.AAAAAQKskxnAqTz0Ah3qu5Hc4Q3lY
dqCocdDZjLXhyLAjuhhBJktOCrBaIJVbKwE
6AVSxD6WAFvI2ZUHs2MUb1gcpqYvro-
kfVv10x7tKZ9GqUObUwKnn5341sU-

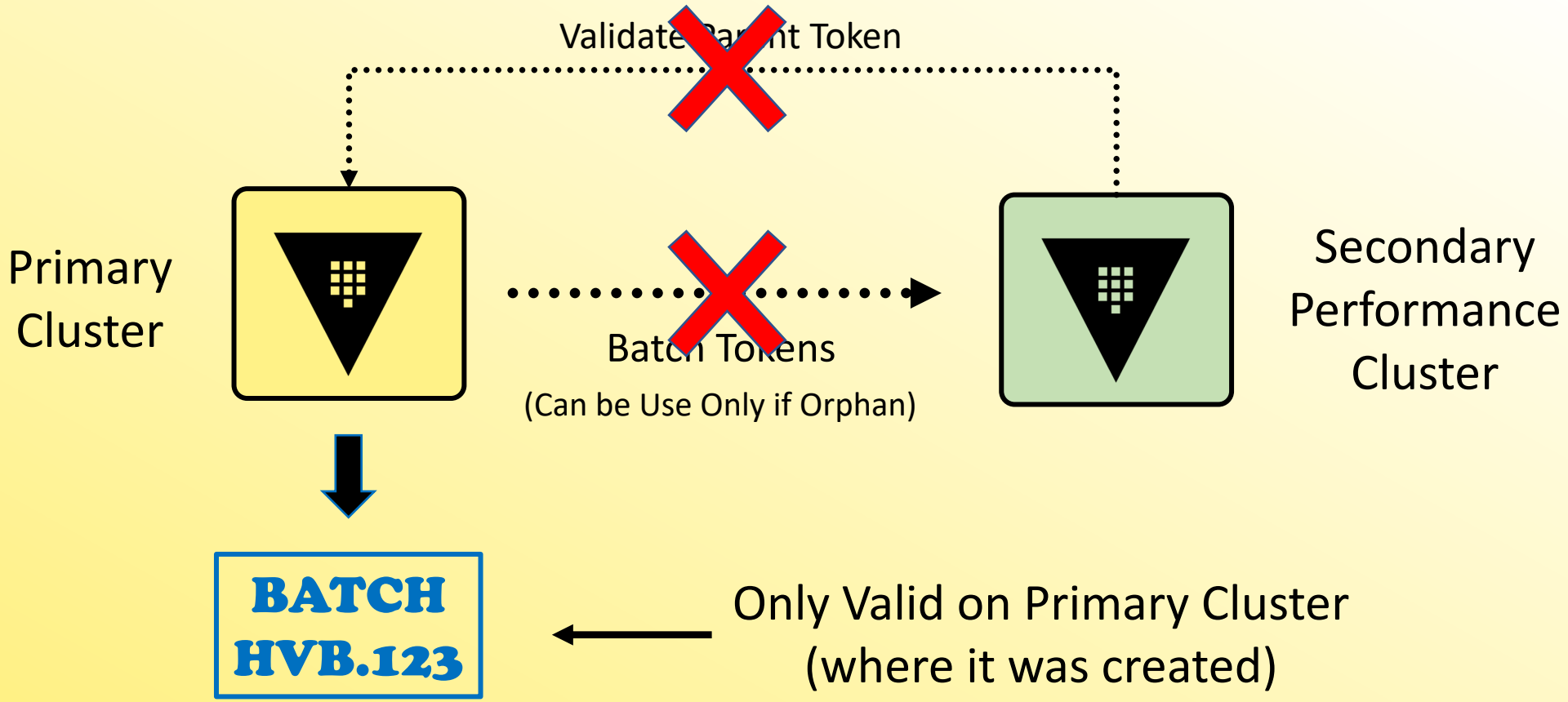
Token Size

Initial Root Token ⟶ hvs.JTjQKbLZOja5LO2anRbGjG6h

Size: ~28 bytes

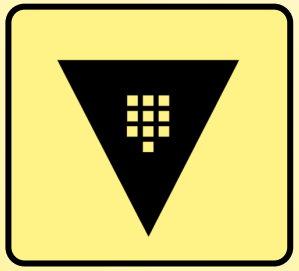
- Token sizes can change. In Vault 1.10, they changed it to 95+ bytes.
- HashiCorp recommends that you plan for a maximum length of 255 bytes to future proof yourself if you have workflows that rely on the token size

Using Batch Tokens – Non-Orphan Token

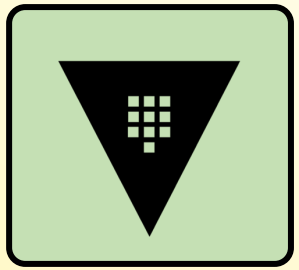


Using Batch Tokens – Orphaned Token

Primary Cluster



Batch Tokens
(Can Be Used Only if Orphan)



Secondary Performance Cluster

BATCH
HVB.987
(orphan)

Has no parent, so we don't need to validate parent
(Valid on any cluster in replica set)

Creating a Batch Token

Available to Use Across Perf Clusters

Terminal

```
$ vault token create -type=batch -orphan=true -policy=hcvop
Key                  Value
---                  -----
Token                hvb.AAAAAQKskxnAqTz0Ah3qu5Hc4Q3lYdqCocdDZ
jLXhyLAjuhhBJktOCrBaIJVbKwE6AVSxD6WAFvI2ZUHs2MUb1gcpqYvro-kfVv
10x7tKZ9GqUObUwKnn5341sU-
token_accessor       n/a
token_duration       768h
token_renewable      false
token_policies       ["default" "hcvop"]
identity_policies    []
policies             ["default" "hcvop"]
```

Creating a Batch Token

Terminal

```
$ vault write auth/approle/role/hcvop policies=devops \
      token_type="batch" \
      token_ttl="60s"
```

DR Operation Batch Token

- *As presented in previous objective, y*ou can use a batch token to promote a DR secondary cluster
    - Eliminates the requirement to generate a DR operation token using the unseal/recovery keys

- This can be a strategic operation that the Vault Operator can to do prepare for an unexpected loss of the primary cluster
- However, the batch token must have the proper permissions to promote a secondary and perform related actions

DR Operation Batch Token

```
path "sys/replication/dr/secondary/promote" {
  capabilities = ["update"]
}

# To update the primary to connect
path "sys/replication/dr/secondary/update-primary" {
  capabilities = ["update"]
}

# Only if using integrated storage (raft) as the storage backend
# To read the current autopilot status
path "sys/storage/raft/autopilot/state" {
  capabilities = ["update" , "read"]
}
```