Token Creation and Lifecycle

Ned Bellavance

HashiCorp Certified Instructor

@nedinthecloud | nedinthecloud.com

Vault CLI Basics

```
$ vault -h

Usage: vault <command> [args]

Common commands:
    read        Read data and retrieves secrets
    write       Write data, configuration, and secrets
    delete      Delete secrets and configuration
    list        List data or secrets
    login       Authenticate locally
    agent       Start a Vault agent
    server      Start a Vault server
    status      Print seal and HA status
    unwrap      Unwrap a wrapped secret
```

Environment Variables

```
# Vault Server Address

VAULT_ADDR="https://vault.yolo.local:8200"

# Vault Token

VAULT_TOKEN=hvs.NJI87YHJHY78UHJU87678UJHGY678
```

Token Commands

```
# Create a token directly
vault token create -policy=POLICY_NAME
vault token create -policy=vault-admins

# Lookup token properties by token id or accessor
vault token lookup
vault token lookup -accessor 8675309JEN

# Renew a token by a given increment
vault token renew
vault token renew -increment=10m
```

Token Commands

```
# Revoke a token by token id or accessor
vault token revoke
vault token revoke -accessor 8675309JEN

# Lookup token capabilities on a given path
vault token capabilities PATH
vault token capabilities -accessor 8675309JEN secret/tacos
```

Service and Batch Token Comparison

Feature	Service	Batch
Can be root token	Yes	No
Can create child token	Yes	No
Can be renewable	Yes	No
Manually revocable	Yes	No
Can be periodic	Yes	No
Can have explicit max TTL	Yes	No
Has accessor	Yes	No
Has cubbyhole	Yes	No
Revoked with parent (not orphan)	Yes	Stops working
Dynamic secrets lease assignment	Self	Parent
Creation scales with Performance Standby	No	Yes
Can be used across Performance clusters	No	Yes
Cost	Heavyweight	Lightweight

Working with TTLs

Time-to-live Basics

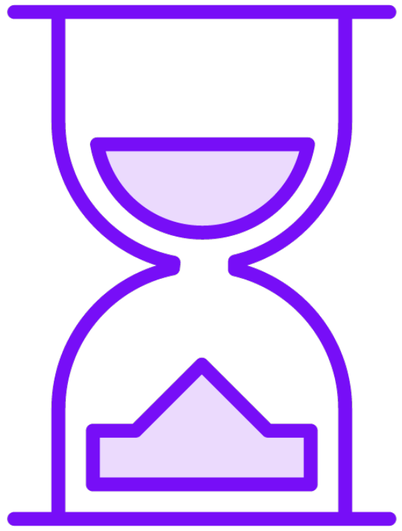
All tokens have a TTL

- Except root tokens

TTL is an absolute value

Token expires when TTL runs out

Max TTL



Maximum lifetime based on creation time

Default value if no TTL specified

Max TTL sources

- System max TTL
- Mount max TTL
- Auth method max TTL
- Explicit token max TTL

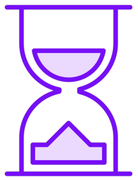
More specific wins

Service and Batch Token Comparison

Feature	Service	Batch
Can be root token	Yes	No
Can create child token	Yes	No
Can be renewable	Yes	No
Manually revocable	Yes	No
Can be periodic	Yes	No
Can have explicit max TTL	Yes	No
Has accessor	Yes	No
Has cubbyhole	Yes	No
Revoked with parent (not orphan)	Yes	Stops working
Dynamic secrets lease assignment	Self	Parent
Creation scales with Performance Standby	No	Yes
Can be used across Performance clusters	No	Yes
Cost	Heavyweight	Lightweight

Token Renewal

Renewal extends TTL



Cannot exceed max TTL

Evaluated during renewal

Service and Batch Token Comparison

Feature	Service	Batch
Can be root token	Yes	No
Can create child token	Yes	No
Can be renewable	Yes	No
Manually revocable	Yes	No
Can be periodic	Yes	No
Can have explicit max TTL	Yes	No
Has accessor	Yes	No
Has cubbyhole	Yes	No
Revoked with parent (not orphan)	Yes	Stops working
Dynamic secrets lease assignment	Self	Parent
Creation scales with Performance Standby	No	Yes
Can be used across Performance clusters	No	Yes
Cost	Heavyweight	Lightweight

Periodic Token

Unlimited renewals

Explicit max TTL allowed

Period defines initial TTL

Only renewed for period increment

Service and Batch Token Comparison

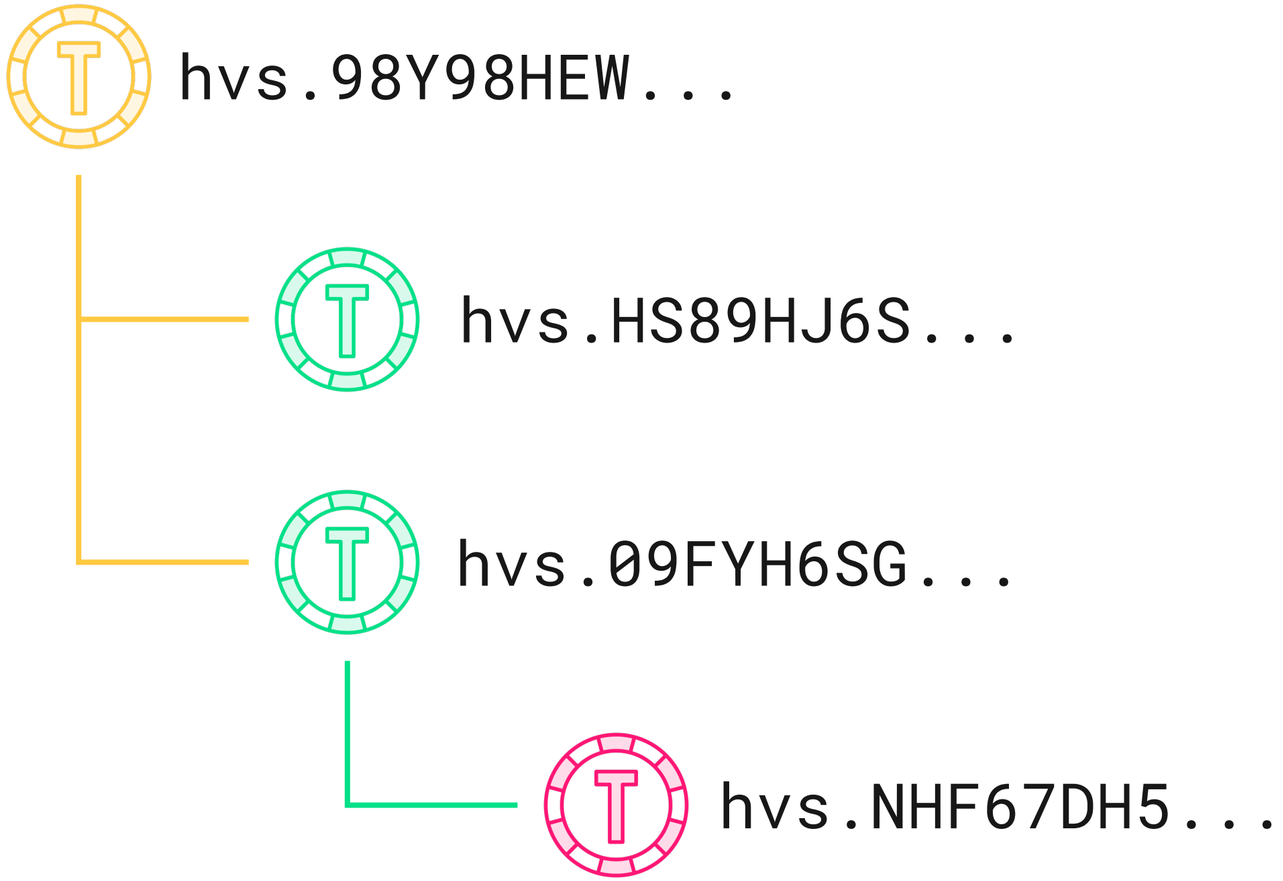
Feature	Service	Batch
Can be root token	Yes	No
Can create child token	Yes	No
Can be renewable	Yes	No
Manually revocable	Yes	No
Can be periodic	Yes	No
Can have explicit max TTL	Yes	No
Has accessor	Yes	No
Has cubbyhole	Yes	No
Revoked with parent (not orphan)	Yes	Stops working
Dynamic secrets lease assignment	Self	Parent
Creation scales with Performance Standby	No	Yes
Can be used across Performance clusters	No	Yes
Cost	Heavyweight	Lightweight

Use Limits

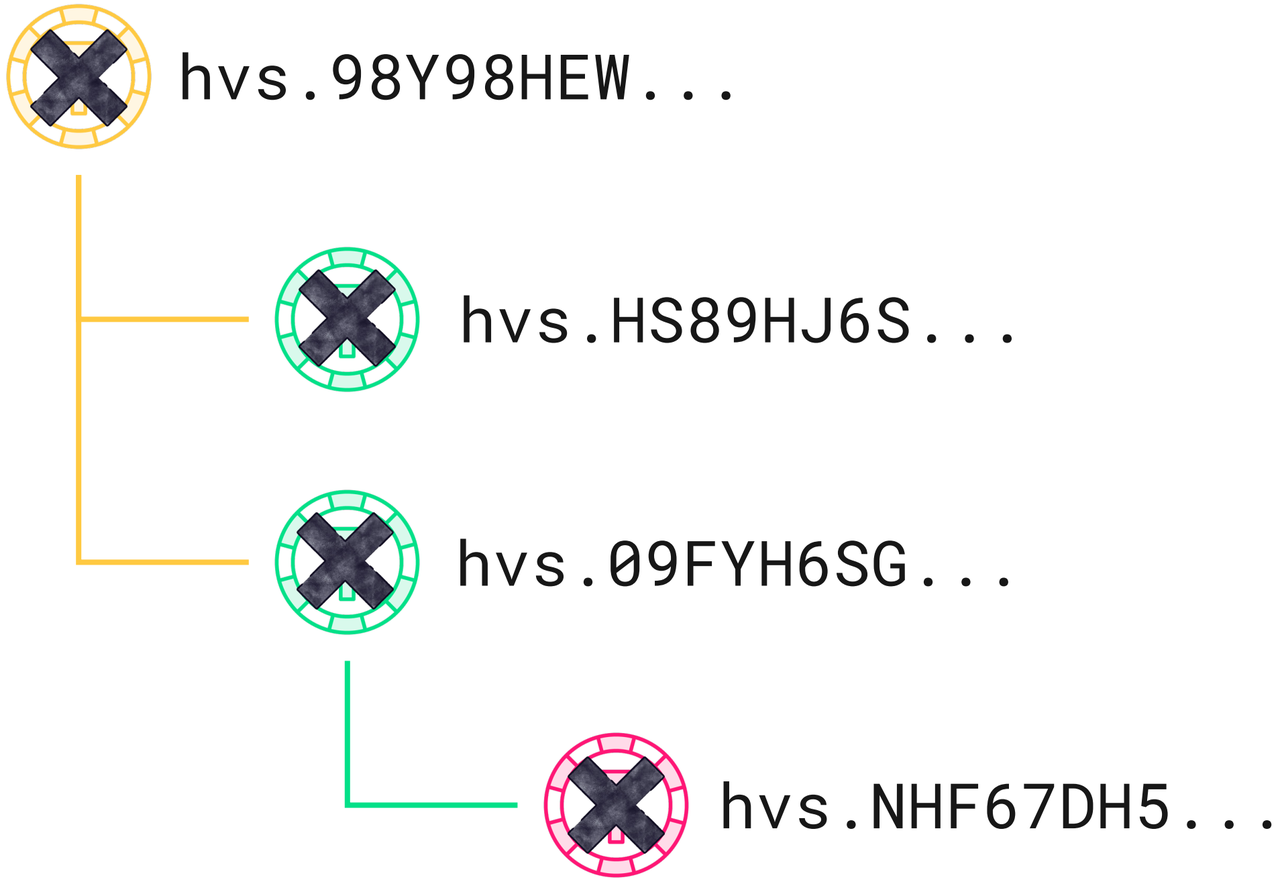
- Unlimited uses by default
- Set limit of uses
- Prevent replay attacks
- Revoked when limit reached
- Service tokens only

Token Lineage

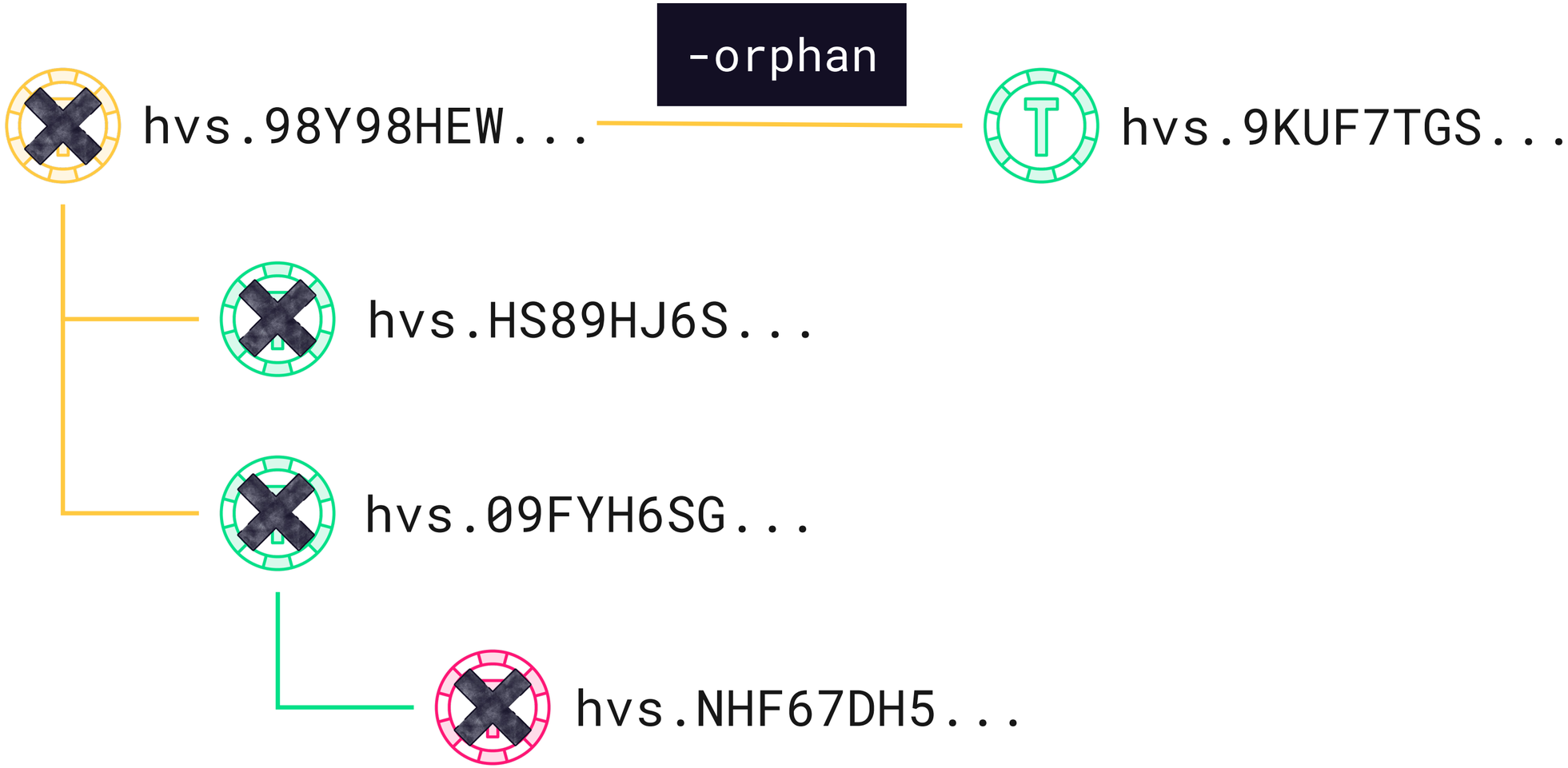
Token Hierarchy



Token Hierarchy



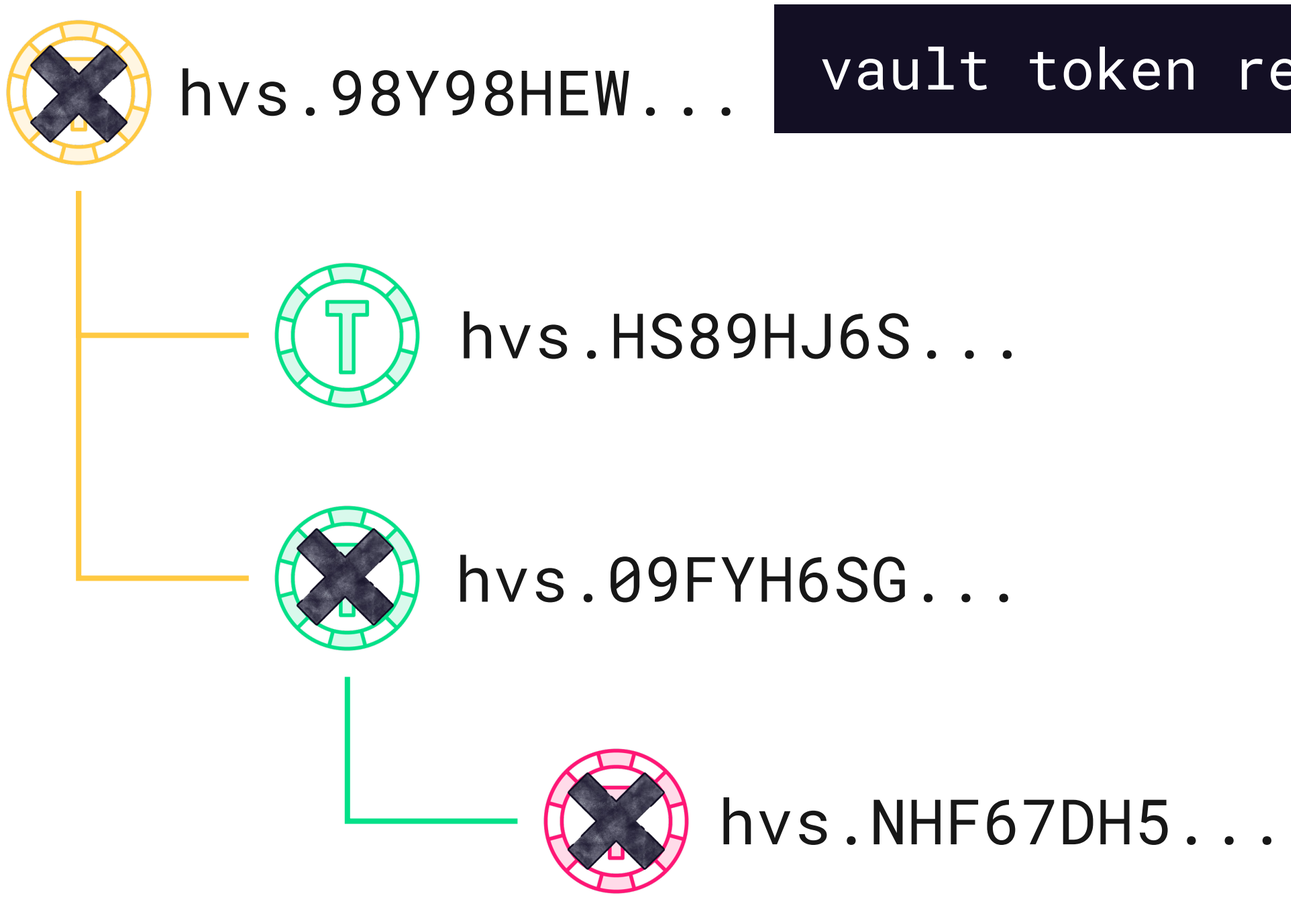
Orphan Tokens



Orphan Permissions

1. Write access on auth/token/create-orphan
2. Sudo access on auth/token/create
3. Token role with permissions
4. Authentication with non-token method

Orphan on Revoke



Service and Batch Token Comparison

Feature	Service	Batch
Can be root token	Yes	No
Can create child token	Yes	No
Can be renewable	Yes	No
Manually revocable	Yes	No
Can be periodic	Yes	No
Can have explicit max TTL	Yes	No
Has accessor	Yes	No
Has cubbyhole	Yes	No
Revoked with parent (not orphan)	Yes	Stops working
Dynamic secrets lease assignment	Self	Parent
Creation scales with Performance Standby	No	Yes
Can be used across Performance clusters	No	Yes
Cost	Heavyweight	Lightweight

Service and Batch Token Comparison

Feature	Service	Batch
Can be root token	Yes	No
Can create child token	Yes	No
Can be renewable	Yes	No
Manually revocable	Yes	No
Can be periodic	Yes	No
Can have explicit max TTL	Yes	No
Has accessor	Yes	No
Has cubbyhole	Yes	No
Revoked with parent (not orphan)	Yes	Stops working
Dynamic secrets lease assignment	Self	Parent
Creation scales with Performance Standby	No	Yes
Can be used across Performance clusters	No	Yes
Cost	Heavyweight	Lightweight

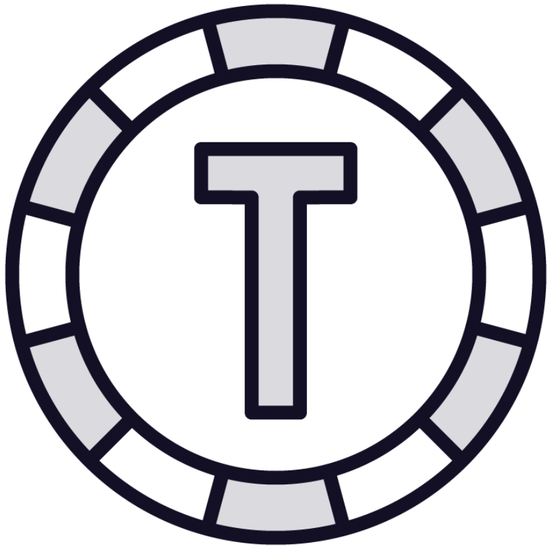
Service and Batch Token Comparison

Feature	Service	Batch
Can be root token	Yes	No
Can create child token	Yes	No
Can be renewable	Yes	No
Manually revocable	Yes	No
Can be periodic	Yes	No
Can have explicit max TTL	Yes	No
Has accessor	Yes	No
Has cubbyhole	Yes	No
Revoked with parent (not orphan)	Yes	Stops working
Dynamic secrets lease assignment	Self	Parent
Creation scales with Performance Standby	No	Yes
Can be used across Performance clusters	No	Yes
Cost	Heavyweight	Lightweight

Service and Batch Token Comparison

Feature	Service	Batch
Can be root token	Yes	No
Can create child token	Yes	No
Can be renewable	Yes	No
Manually revocable	Yes	No
Can be periodic	Yes	No
Can have explicit max TTL	Yes	No
Has accessor	Yes	No
Has cubbyhole	Yes	No
Revoked with parent (not orphan)	Yes	Stops working
Dynamic secrets lease assignment	Self	Parent
Creation scales with Performance Standby	No	Yes
Can be used across Performance clusters	No	Yes
Cost	Heavyweight	Lightweight

Root Token Creation



Token with root policy attached

TTL is optional

Revoke as soon as possible

Creation process

- Vault initialization
- Another root token
- Using operator generate-root

Used for break-glass scenarios

Root Token Creation

Vault Operator Process

```
> vault operator generate-root -init

Nonce            635d8a0f-fe5e-77a6-ef2d-0554aed62765
Started          true
Progress         0/1
Complete         false
OTP              OTP_VALUE
OTP Length       28
```

Root Token Creation

Vault Operator Process

```
> vault operator generate-root

Operation nonce: 635d8a0f-fe5e-77a6-ef2d-0554aed62765
Unseal Key (will be hidden):

Nonce            635d8a0f-fe5e-77a6-ef2d-0554aed62765
Started          true
Progress         1/1
Complete         true
Encoded Token    ENCODED_TOKEN
```

Root Token Creation

Vault Operator Process

```
> vault operator generate-root -decode=ENCODED_TOKEN -otp=OTP_VALUE

hvs.o1vTpJ2wmfm2QCPfJRCDgawf
```