SecureVault CLI Reference

Document ID: PRD-SV-015 **Last Updated:** 2024-02-15 **Owner:** SecureVault Engineering **Classification:** Public

---

Installation

macOS

```
brew tap novatech/securevault
brew install securevault
```

Linux

```
curl -fsSL https://get.securevault.novatech.com | bash
```

Windows

```
winget install NovaTech.SecureVault
```

Docker

```
docker run -it novatech/securevault:latest
```

Configuration

Initial Setup

```
# Interactive configuration
securevault configure

# Or specify options
securevault configure \
  --addr https://vault.novatech.com \
  --token sv_xxx
```

Configuration File

Location: `~/.securevault/config.yaml`

```yaml
default_vault: production

vaults:
  production:
    address: https://vault.novatech.com
    token: sv_prod_xxx

  development:
    address: https://vault-dev.novatech.com
    token: sv_dev_xxx
```

Environment Variables

Variable	Description
`VAULT_ADDR`	SecureVault server address
`VAULT_TOKEN`	Authentication token
`VAULT_NAMESPACE`	Default namespace
`VAULT_FORMAT`	Output format (json, table, yaml)

Authentication

Login Methods

```bash
# Token authentication
securevault login token
# Enter token when prompted

# OIDC authentication
securevault login oidc
# Opens browser for SSO

# AppRole authentication
securevault login approle \
  --role-id abc123 \
  --secret-id xyz789
```

```
# Kubernetes authentication
securevault login kubernetes \
  --role my-role
```

Token Management

```
# Check current token
securevault token lookup

# Show token capabilities
securevault token capabilities secrets/data/myapp/*

# Renew token
securevault token renew

# Revoke token
securevault token revoke
```

Secrets Operations

Read Secrets

```
# Read a secret
securevault secrets get secrets/myapp/api-key

# Read specific field
securevault secrets get secrets/myapp/config --field database_url

# Output as JSON
securevault secrets get secrets/myapp/config --format json

# Read specific version
securevault secrets get secrets/myapp/config --version 3
```

Write Secrets

```
# Write a secret
securevault secrets put secrets/myapp/api-key value=sk_live_xxx

# Write multiple fields
securevault secrets put secrets/myapp/config \
  database_url=postgres://... \
  redis_url=redis://... \
  api_key=xxx

# Write from file
securevault secrets put secrets/myapp/config @config.json

# Write from stdin
echo '{"key": "value"}' | securevault secrets put secrets/myapp/config -
```

Delete Secrets

```
# Delete latest version (soft delete)
securevault secrets delete secrets/myapp/api-key

# Delete specific versions
securevault secrets delete secrets/myapp/api-key --versions 2,3

# Permanently destroy
securevault secrets destroy secrets/myapp/api-key --versions 1,2,3

# Delete all versions (metadata remains)
securevault secrets delete secrets/myapp/api-key --all
```

List Secrets

```
# List secrets in path
securevault secrets list secrets/myapp/

# Recursive listing
securevault secrets list secrets/ --recursive

# List with metadata
securevault secrets list secrets/myapp/ --detailed
```

Secret Versions

```
# List versions
securevault secrets versions secrets/myapp/config

# Rollback to version
securevault secrets rollback secrets/myapp/config --version 2
```

```
# Undelete version
securevault secrets undelete secrets/myapp/config --versions 3
```

Dynamic Secrets

Database Credentials

```
# Generate database credentials
securevault dynamic get database/creds/myapp-readonly

# Output:
# username: v-token-myapp-xxx
# password: xxx
# ttl: 1h

# Renew lease
securevault lease renew <lease_id>

# Revoke lease
securevault lease revoke <lease_id>
```

AWS Credentials

```
# Generate AWS credentials
securevault dynamic get aws/creds/s3-access

# Output:
# access_key: AKIA...
# secret_key: xxx
# ttl: 1h
```

Policies

View Policies

```
# List all policies
securevault policy list

# Read policy
```

```
securevault policy read my-policy

# Check your policies
securevault token lookup --format json | jq '.policies'
```

Create/Update Policies

```
# Write policy from file
securevault policy write my-policy policy.hcl

# Write policy inline
securevault policy write my-policy - << EOF
path "secrets/data/myapp/*" {
  capabilities = ["read", "list"]
}
EOF
```

Delete Policies

```
securevault policy delete my-policy
```

Access Control

Roles

```
# List AppRoles
securevault approle list

# Create AppRole
securevault approle create my-app \
  --policies my-policy \
  --token-ttl 1h

# Get Role ID
securevault approle role-id my-app

# Generate Secret ID
securevault approle secret-id my-app
```

Groups

```
# List groups
securevault group list

# Create group
securevault group create platform-team \
  --policies platform-secrets

# Add member
securevault group add-member platform-team \
  --entity my-entity
```

Audit

View Audit Log

```
# Recent audit entries
securevault audit list --limit 100

# Filter by path
securevault audit list --path secrets/myapp/*

# Filter by time range
securevault audit list \
  --start 2024-01-01 \
  --end 2024-01-31
```

Operators

Server Status

```
# Check server status
securevault status

# Check health
securevault health

# Seal status
securevault seal-status
```

Key Rotation

```
# Rotate encryption key
securevault operator rotate

# Rekey (requires multiple key holders)
securevault operator rekey
```

Import/Export

Export Secrets

```
# Export to JSON
securevault export secrets/myapp/ > backup.json

# Export with encryption
securevault export secrets/myapp/ \
  --encrypt \
  --public-key key.pem > backup.enc
```

Import Secrets

```
# Import from JSON
securevault import secrets/myapp/ < backup.json

# Import with prefix
securevault import secrets/myapp/restored/ < backup.json
```

Environment Injection

Run Commands with Secrets

```
# Inject secrets as env vars
securevault exec --secrets secrets/myapp/config -- ./myapp

# Template file with secrets
securevault template config.tpl config.yaml
```

Template Example

```
# config.tpl
database:
  url: {{ secret "secrets/myapp/config" "database_url" }}

api:
  key: {{ secret "secrets/myapp/api-key" "value" }}
```

Output Formats

Available Formats

```
# Table (default)
securevault secrets get secrets/myapp/config

# JSON
securevault secrets get secrets/myapp/config --format json

# YAML
securevault secrets get secrets/myapp/config --format yaml

# Just the value
securevault secrets get secrets/myapp/config --field api_key --raw
```

Global Format Setting

```
export VAULT_FORMAT=json
```

Common Options

Option	Description
`--addr`	Override vault address
`--token`	Override token
`--namespace`	Set namespace
`--format`	Output format
`--help`	Show help

Option	Description
`--version`	Show version
`-v, --verbose`	Verbose output
`-q, --quiet`	Suppress output

Examples

Application Configuration

```
# Store app config
securevault secrets put secrets/myapp/config \
  DATABASE_URL='postgres://user:pass@host:5432/db' \
  REDIS_URL='redis://localhost:6379' \
  API_KEY='sk_live_xxx'

# Read in app
export $(securevault secrets get secrets/myapp/config --format env)
./myapp
```

CI/CD Integration

```
#!/bin/bash
# ci-script.sh

# Login with AppRole
securevault login approle \
  --role-id "$VAULT_ROLE_ID" \
  --secret-id "$VAULT_SECRET_ID"

# Get secrets
DB_PASSWORD=$(securevault secrets get secrets/myapp/db --field password --raw)
API_KEY=$(securevault secrets get secrets/myapp/api --field key --raw)

# Use in deployment
./deploy.sh --db-password "$DB_PASSWORD" --api-key "$API_KEY"
```

Secret Rotation Script

```
#!/bin/bash
# rotate-secrets.sh
# Generate new API key
NEW_KEY=$(openssl rand -hex 32)

# Store in vault
securevault secrets put secrets/myapp/api-key \
  value="$NEW_KEY" \
  rotated_at="$(date -u +%Y-%m-%dT%H:%M:%SZ)"

# Trigger app reload
curl -X POST https://myapp/admin/reload
```

Troubleshooting

Common Errors

"permission denied"

```
# Check capabilities
securevault token capabilities secrets/data/myapp/*
```

"token expired"

```
# Check token
securevault token lookup

# Re-authenticate
securevault login <method>
```

"connection refused"

```
# Check address
echo $VAULT_ADDR

# Test connectivity
curl -s $VAULT_ADDR/v1/sys/health
```

Related Documents: Setup & Installation (PRD-SV-001), Access Control Guide (PRD-SV-010), API Reference (PRD-SV-020)