VAULT

Documentation Style Guide | v2.4 | July 2025

1. Voice and Tone

1.1 Core Principles

- Clear — Use plain English. Avoid jargon where simpler terms work.
- Direct — Address the reader as 'you'. Use imperative mood for instructions.
- Accurate — Never document behavior you haven't verified. When in doubt, test it.
- Helpful — Anticipate what the reader needs to know next. Add notes and warnings proactively.

1.2 Voice Examples

Avoid	Prefer
The user should click the button.	Click the button.
It is possible to enable audit logging.	You can enable audit logging.
Vault allows for the configuration of...	Configure Vault to...
The aforementioned policy...	This policy...

2. Formatting Standards

2.1 Headings

Level	Usage	Example
H1	Chapter or major section	3. Core Concepts
H2	Subsection within a chapter	3.1 Secrets Engines
H3	Sub-subsection	3.1.1 KV Engine
H4	Rarely used — avoid nesting deeper	n/a

2.2 Code Blocks

Use code blocks for:

- All CLI commands
- All API requests and responses
- Configuration file examples
- Code samples in any language

Do not use code blocks for:

- File paths mentioned inline — use monospace inline formatting instead
- Environment variable names mentioned inline — use monospace inline formatting

2.3 Callouts

Type	When to Use
NOTE	Additional context that is helpful but not critical.
TIP	A best practice or shortcut the reader may find useful.
WARNING	Actions that could cause data loss or service disruption.
DANGER	Actions that are irreversible or pose a security risk.

⚠ Do not overuse callouts. If every paragraph has a callout, none of them stand out. Reserve WARNING and DANGER for genuinely important alerts.

3. Terminology

3.1 Product Terms

Correct	Incorrect	Notes
Vault	vault, VAULT	Always capitalize as a proper noun.
Vault server	the server	Be specific — 'server' alone is ambiguous.
secrets engine	secret engine, plugin	Always plural 'secrets' as adjective.
KV v2	KV2, kv2, KV version 2	Use this exact form.
token	Token (do not capitalize)	Lowercase except at start of sentence.
policy	Policy (do not capitalize)	Lowercase except at start of sentence.
AppRole	app role, Approle, APPROLE	One word, camel case.

3.2 Commonly Confused Terms

Term	Definition
seal vs. initialize	Sealing encrypts Vault's keys. Initialization creates the keys. A newly installed Vault is uninitialized, not sealed.
path vs. mount	A mount is the root path where a secrets engine is enabled. A path is the full location of a secret including the mount.
delete vs. destroy	Delete (KV v2) creates a soft delete — data is retained but inaccessible. Destroy permanently removes data.
token vs. secret ID	A token is a Vault auth credential. A Secret ID is used only with AppRole — it's not a general Vault token.

4. Procedures

4.1 Numbered Steps

Use numbered steps for sequential tasks where order matters. Each step should contain exactly one action:

Good:

1. Enable the KV secrets engine.
2. Create a policy for your application.
3. Create a token with that policy.

Avoid:

4. Enable the KV secrets engine and create a policy, then create a token.

4.2 Prerequisites

Always list prerequisites before a procedure using a 'Before you begin' section:

- Vault 2.4.0 or later is installed
- You have a Vault token with the required permissions
- The VAULT_ADDR environment variable is set

5. Code Examples

5.1 CLI Examples

Always use realistic, consistent example values:

- Use secret/myapp/ as the example path
- Use myapp-policy as the example policy name
- Use $VAULT_TOKEN and $VAULT_ADDR as environment variable placeholders
- Use vault.example.com as the example Vault address
- Replace sensitive values with <PLACEHOLDER> in all-caps angle brackets

5.2 Code Comments

Add inline comments to complex commands:

```
# Generate a new Secret ID (treat like a password — store securely) vault write -f
auth/approle/role/myapp/secret-id
```

5.3 Expected Output

Always show expected output for commands where the output is not obvious. Label output sections clearly:

Command:

```
vault kv get secret/myapp/config
```

Expected output:

```
====== Secret Path ====== secret/data/myapp/config  === Metadata === Key           Value
created_time  2025-07-15T10:00:00Z version       3  === Data === Key       Value api_key
my-api-key
```

6. Review Checklist

Before submitting documentation for review, verify:

Content

5. All commands have been tested against the version being documented
6. All links are valid and point to the correct destinations
7. Prerequisites are listed before any procedure
8. Expected output is shown for all non-trivial commands

9. Callouts are used appropriately and sparingly

Style

10. Product terms match the terminology table
11. Headings follow the hierarchy (H1 → H2 → H3)
12. Numbered steps contain exactly one action each
13. Sensitive values use <PLACEHOLDER> format
14. Code blocks are used consistently for all commands and output

Accessibility

15. All tables have header rows
16. Images have descriptive alt text
17. Color is not the only way information is conveyed
18. Document can be navigated by heading structure alone