Vault Policy Syntax

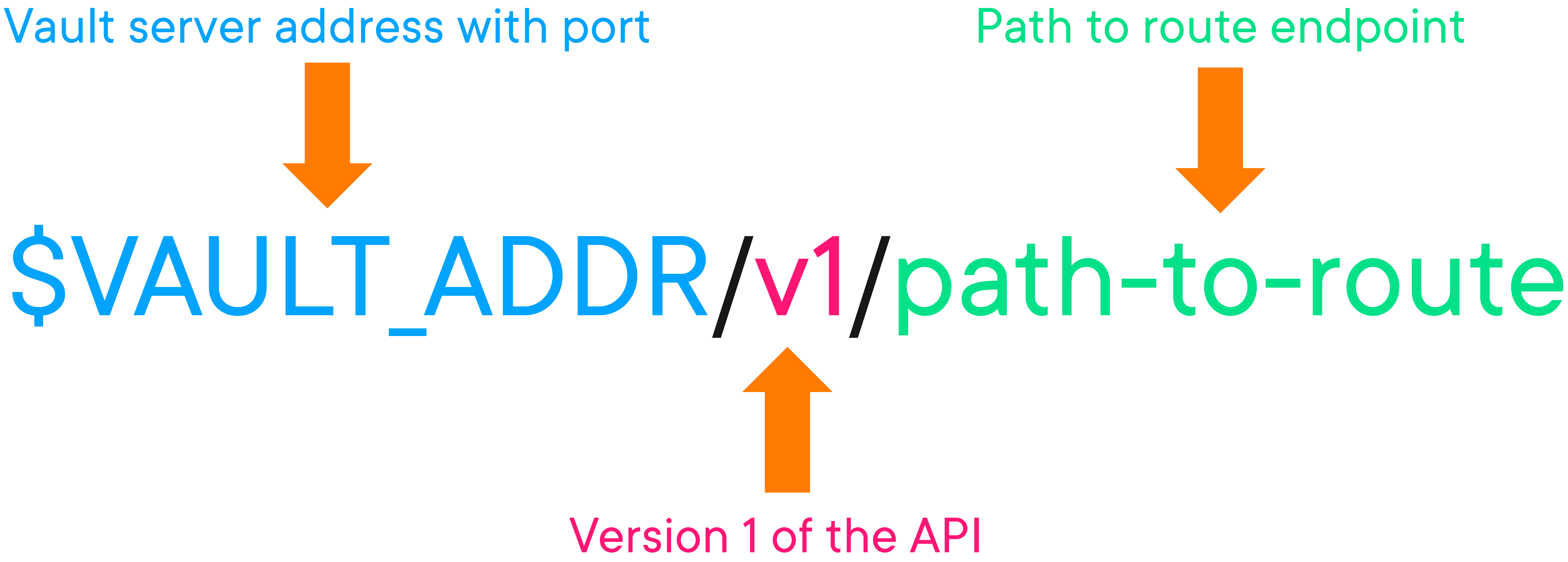
Ned Bellavance

HashiCorp Certified Instructor

@nedinthecloud | nedinthecloud.com

Everything in Vault is treated as a path

Vault API Path Construction



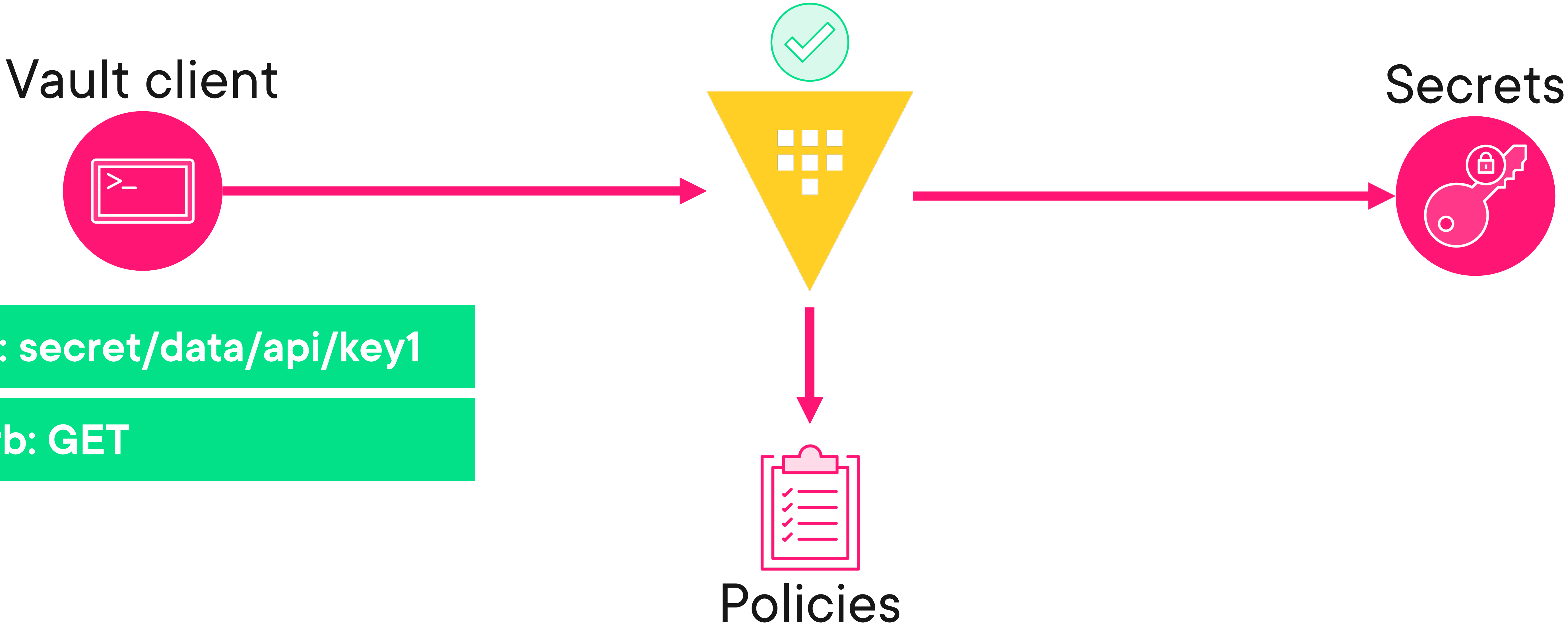
Vault server address with port

Path to route endpoint

$VAULT_ADDR/v1/path-to-route

Version 1 of the API

Vault Policy Evaluation



Vault client

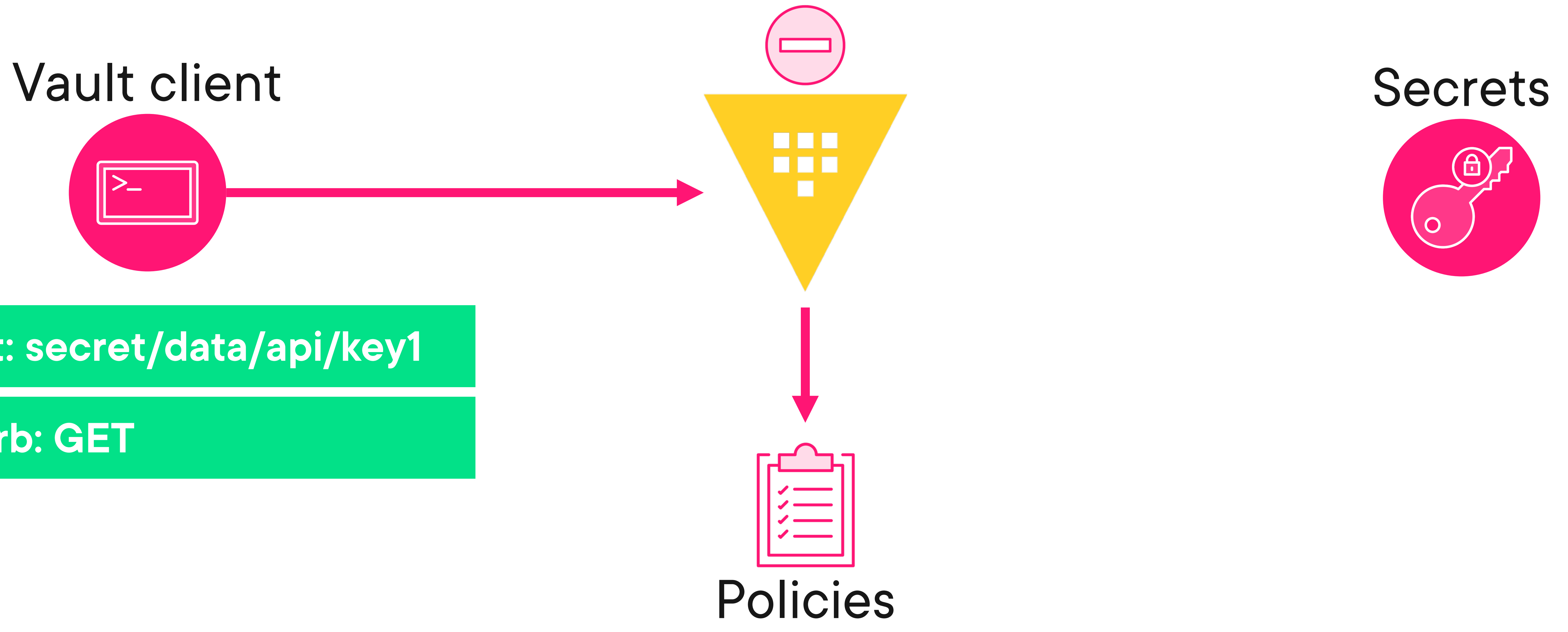
Secrets

Policies

Endpoint: secret/data/api/key1

HTTP Verb: GET

Vault Policy Evaluation



Endpoint: secret/data/api/key1

HTTP Verb: GET

Policy File Basics

example.hcl

```
block_type "block_label" {
  # Arguments
  identifier = value
}
```

Policy File Basics

my-policy.hcl

```
path "some_path_expression" {
  capabilities = []
}

path "some_path_expression" {
  capabilities = []
}

path "some_path_expression" {
  capabilities = []
}
```

Path Block Label

Type	Example	Matches
Literal	`secret/taco/sauce`	`secret/taco/sauce`

Path Block Label

Type	Example	Matches
Literal	`secret/taco/sauce`	`secret/taco/sauce`
Glob (*)	`secret/taco*`	`secret/taco-tuesdays` `secret/taco/toppings`

Path Block Label

Type	Example	Matches
Literal	`secret/taco/sauce`	`secret/taco/sauce`
Glob (*)	`secret/taco*`	`secret/taco-tuesdays` `secret/taco/toppings`
Path segment (+)	`secret/+/sauce`	`secret/taco/sauce` `secret/enchilada/sauce`

Path Block Label

Type	Example	Matches
Literal	`secret/taco/sauce`	`secret/taco/sauce`
Glob (*)	`secret/taco*`	`secret/taco-tuesdays` `secret/taco/toppings`
Path segment (+)	`secret/+/sauce`	`secret/taco/sauce` `secret/enchilada/sauce`
Path segment (+)	`secret/+/+/recipe`	`secret/taco/red/recipe` `secret/burrito/beef/recipe`

Path Block Label

Type	Example	Matches
Literal	`secret/taco/sauce`	`secret/taco/sauce`
Glob (*)	`secret/taco*`	`secret/taco-tuesdays` `secret/taco/toppings`
Path segment (+)	`secret/+/sauce`	`secret/taco/sauce` `secret/enchilada/sauce`
Path segment (+)	`secret/+/+/recipe`	`secret/taco/sauce/recipe` `secret/burrito/beef/recipe`
Combination	`secret/+/sauce/*`	`secret/burrito/sauce/ingredients` `secret/taco/sauce/recipe`

Path Block Label

Type	Example	Matches
Literal	`secret/taco/sauce`	`secret/taco/sauce`
Glob (*)	`secret/taco*`	`secret/taco-tuesdays` `secret/taco/toppings`
Path segment (+)	`secret/+/sauce`	`secret/taco/sauce` `secret/enchilada/sauce`
Path segment (+)	`secret/+/+/recipe`	`secret/taco/sauce/recipe` `secret/burrito/beef/recipe`
Combination	`secret/+/sauce/*`	`secret/burrito/sauce/ingredients` `secret/taco/sauce/recipe`
Template parameter	`secret/{{identity.entity.name}}/*`	`secret/ned/*`

Priority Matching

More specific path wins

Union of capabilities on exact match

Given policies P1 and P2:

1. If the first wildcard (+) or glob (*) occurs earlier in P1, P1 is lower priority
2. If P1 ends in * and P2 doesn't, P1 is lower priority
3. If P1 has more + (wildcard) segments, P1 is lower priority
4. If P1 is shorter, it is lower priority
5. If P1 is smaller lexicographically, it is lower priority

Capabilities

Capability	HTTP Verb
`create*`	**POST/PUT**

Capabilities

Capability	HTTP Verb
create*	**POST/PUT**
read	**GET**

Capabilities

Capability	HTTP Verb
`create*`	**POST/PUT**
`read`	**GET**
`update*`	**POST/PUT**

Capabilities

Capability	HTTP Verb
`create*`	**POST/PUT**
`read`	**GET**
`update*`	**POST/PUT**
`patch`	**PATCH**

Capabilities

Capability	HTTP Verb
create*	**POST/PUT**
read	**GET**
update*	**POST/PUT**
patch	**PATCH**
delete	**DELETE**

Capabilities

Capability	HTTP Verb
create*	**POST/PUT**
read	**GET**
update*	**POST/PUT**
patch	**PATCH**
delete	**DELETE**
list	**LIST**

Capabilities

Capability	HTTP Verb
create*	**POST/PUT**
read	**GET**
update*	**POST/PUT**
patch	**PATCH**
delete	**DELETE**
list	**LIST**

*create and update are usually granted together. Refer to API documentation.

Special Capabilities

`sudo`: **required for root-protected endpoints**

`deny`: **overrides all other capabilities**

`subscribe`: **events at the endpoint (Enterprise only)**

Capabilities

my-policy.hcl

```
path "auth/ldap" {
  capabilities = ["create","read","update","sudo"]
}

path "secret/taco*" {
  capabilities = ["read","list"]
}

path "secret/+/sauce" {
  capabilities = ["deny"]
}
```

Fine-grained Controls

required_parameters

allowed_parameters

denied_parameters

Fine-grained Controls

my-policy.hcl

```
path "secret/taco/recipes/*" {
 capabilities = ["create", "read", "update", "list"]

  required_parameters = ["spice_level"]

  allowed_parameters = {
    "spice_level" = ["mild","medium","hot"]
    "*" = []
  }

  denied_parameters = {
    "price" = []
  }
}
```

Special Policies

`root`

- Full permissions on Vault
- Cannot be renamed, altered, or deleted
- Only assignable by root token

`default`

- Assigned to all tokens unless exempted
- Set of useful permissions for clients
- Cannot be renamed or deleted
- Can be altered
- Latest version from dev server

Helpful Tools

```
> vault list –output-policy auth/ldap/users

path "auth/ldap/users" {
  capabilities = ["list"]
}

> vault token capabilities auth/ldap/users

deny
```