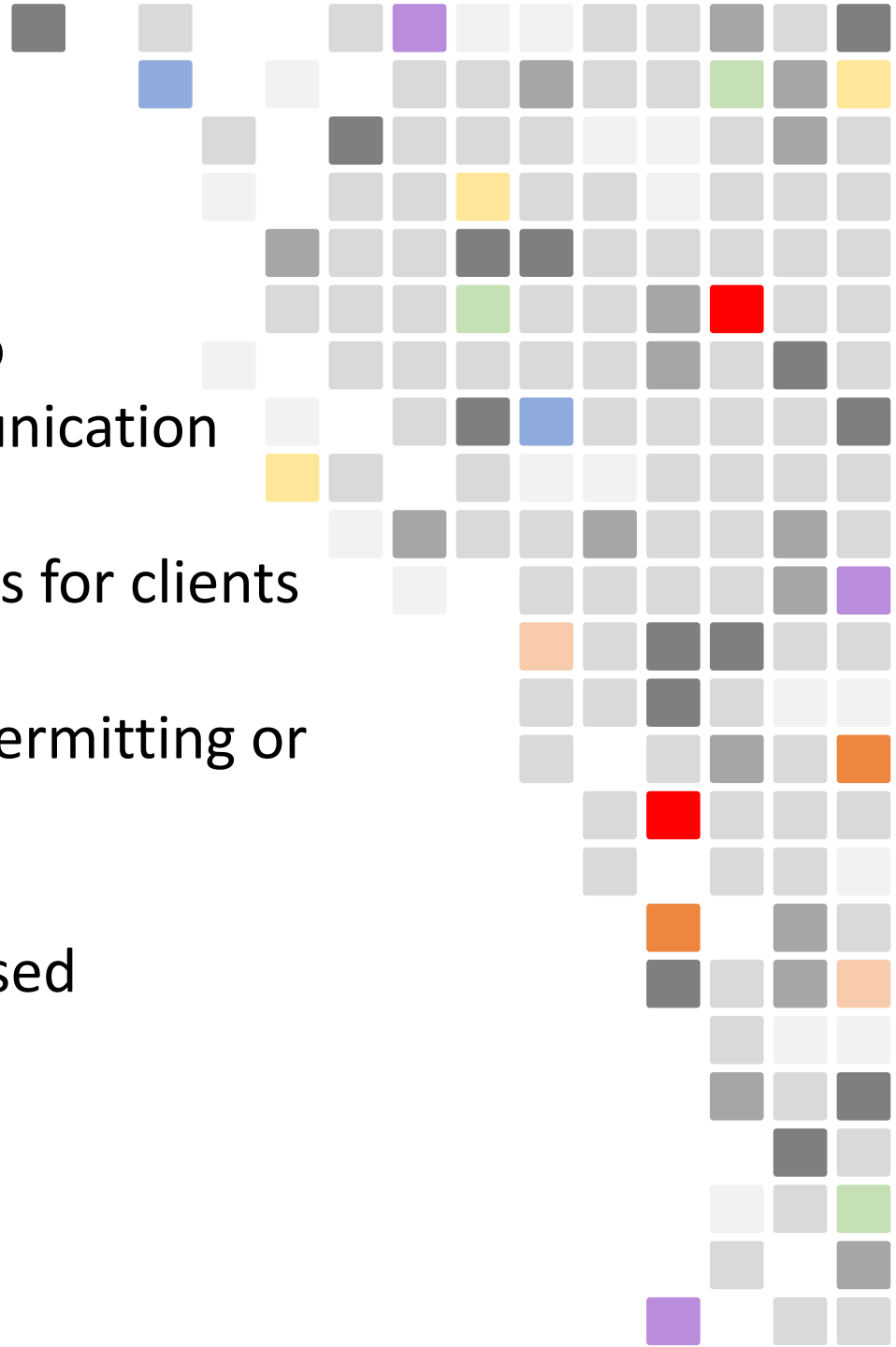
Section 4

Consul ACLs for Vault

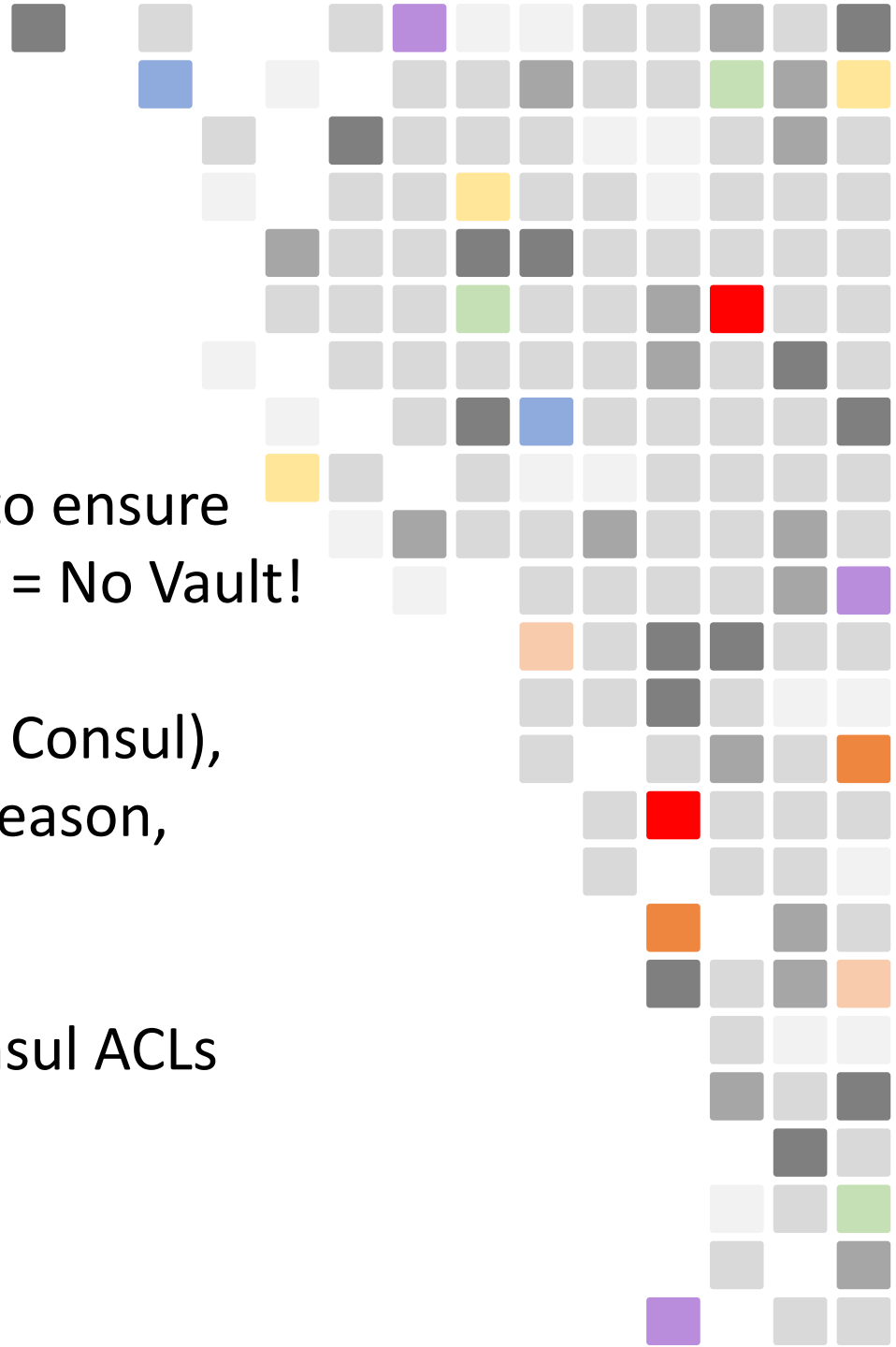
What are Consul ACLs?

- Consul ACLs are a system for controlling access to Consul data and UI, API, CLI, and all other communication
- Consul ACLs rely on policies and associated tokens for clients
- Policies are, well, policies and made up of rules permitting or denying access to Consul data and functions
- ACLs must be bootstrapped before they can be used



Why does Vault Need ACLs?

- Vault stores data on Consul's integrated KV store
- The data stored in the KV needs to be protected to ensure continuous functionality of Vault. No Consul data = No Vault!
- Although the data is encrypted (by Vault – not by Consul), nobody should be accessing the KV data for any reason, especially not deleting data.
- The only way to protect this KV data is to use Consul ACLs



Steps Required for Consul ACL Deploy

1 Bootstrap Consul ACLs

```
$ consul acl bootstrap
```

2 Create Policies for Required Services

```
$ consul acl policy create –name “Consul_Nodes” –rules @consul.hcl
```

3 Create Tokens for Associated Policies

```
$ consul acl token create –description “Consul Node Policy” –policy-id “<id>”
```

4 Update Services with Tokens

Consul server/client configuration: →

```
“acl”: {
  “tokens”: {
     “agent”: “<token-id>
  }
}
```

5 Set Default ACL Policy to ‘Deny’

```
“default_policy”: “deny”
```

Consul ACL Policies for Vault Deploy

1 Consul Nodes for Internal Communication

— the Consul nodes that make up the Consul cluster

2 Consul Agent for Vault

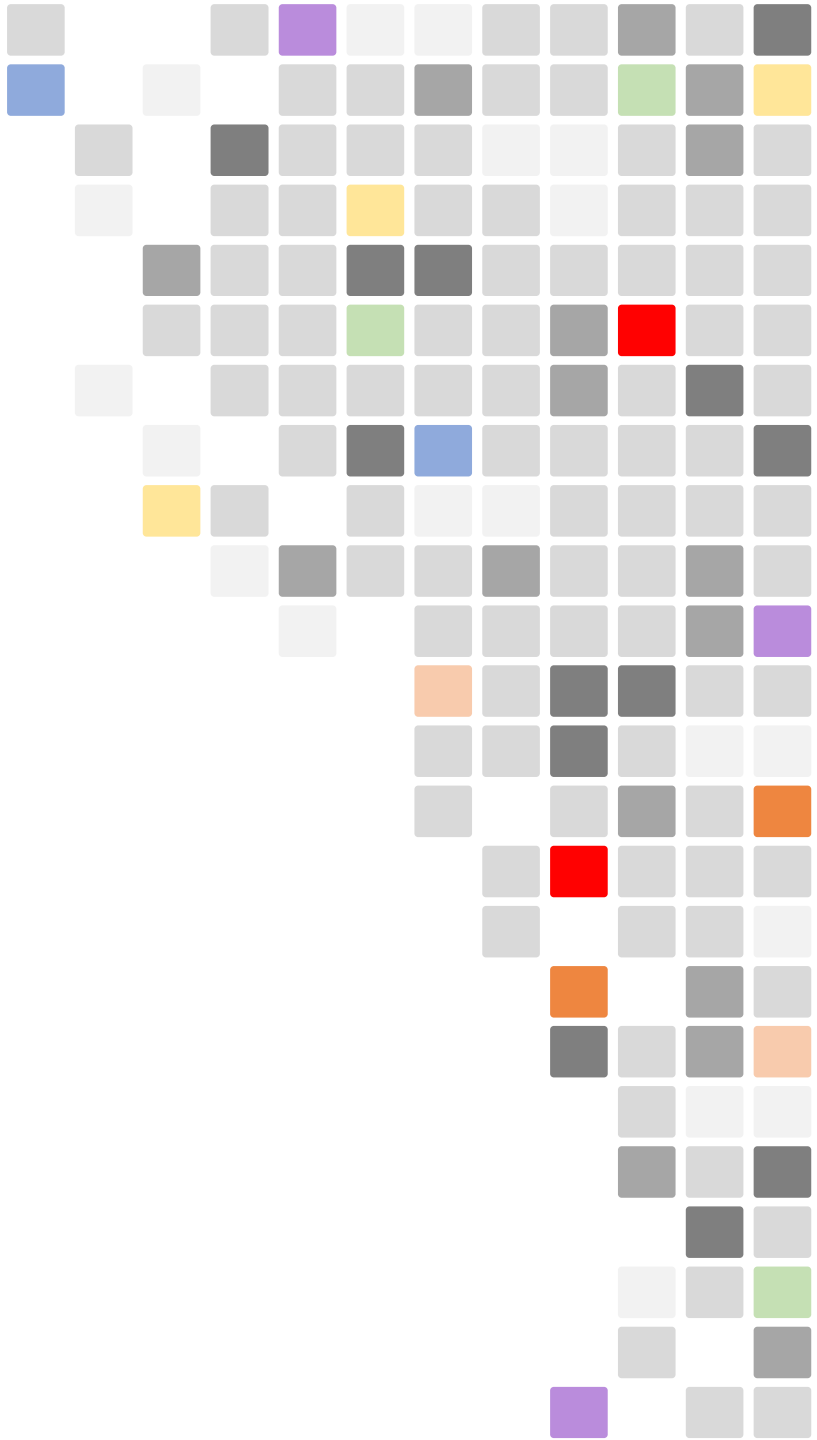
— the Consul agent running on Vault

3 Vault Service

— the Vault service running on the Vault nodes

4 Consul Snapshots

— the Consul snapshot agent for Consul backups (Ent Feature)



Defining Consul ACL Policies

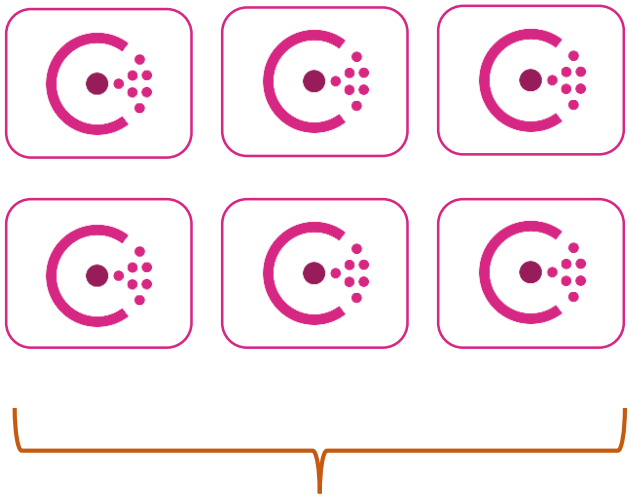
Defining Consul ACL Policies

Consul Nodes for Internal Communication

```
# Consul Node Policy

node_prefix "" {
  policy = "write"
}
agent_prefix "" {
  policy = "write"
}
service "consul" {
  policy = "write"
}
service_prefix "" {
  policy = "read"
}
```

used for:



the Consul nodes that make up the Consul cluster

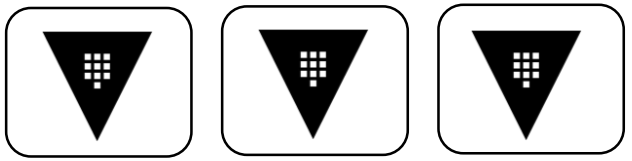
Defining Consul ACL Policies

Consul Agent for Vault

```
# Consul Agent for Vault

node_prefix "" {
  policy = "write"
}
service_prefix "" {
  policy = "read"
}
agent_prefix "" {
  policy = "write"
}
```

used for:



the Consul agent running on Vault

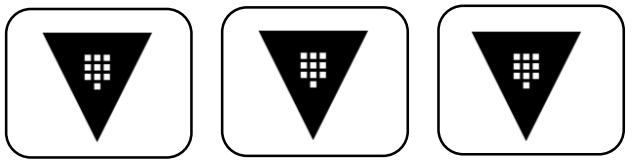
Defining Consul ACL Policies

Vault Service

```
# Vault Service

key_prefix "vault/" {
  policy = "write"
}
node_prefix "" {
  policy = "write"
}
service "vault" {
  policy = "write"
}
agent_prefix "" {
  policy = "write"
}
session_prefix "" {
  policy = "write"
}
```

used for:



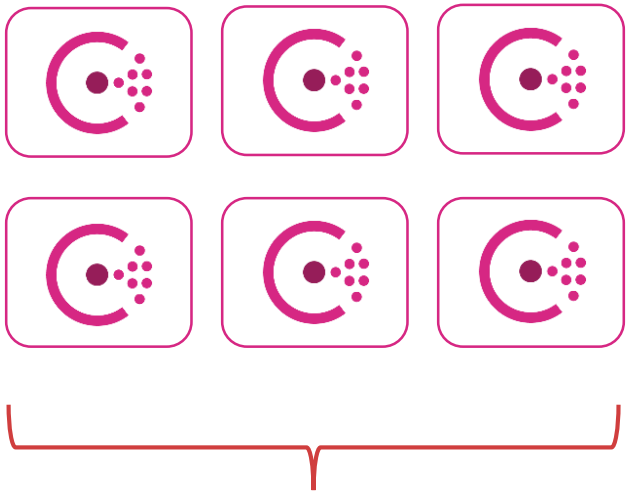
the Vault service running on the Vault nodes

Defining Consul ACL Policies

Consul Snapshots

```
# Consul Snapshot Agent
acl = "write"
key "consul-snapshot/lock" {
  policy = "write"
}
session_prefix "" {
  policy = "write"
}
service "consul-snapshot" {
  policy = "write"
}
```

used for:

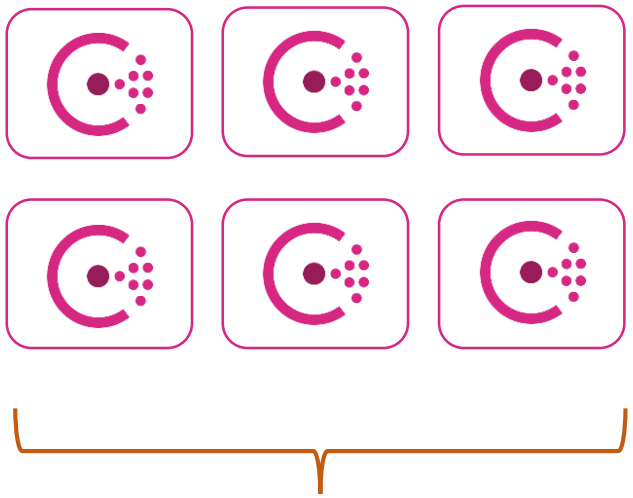


the Consul snapshot agent for Consul backups (Ent Feature)

Updating Configurations with ACL Token

Updating Configs with ACL Token

Consul Nodes for internal communication



the Consul nodes that make up the Consul cluster

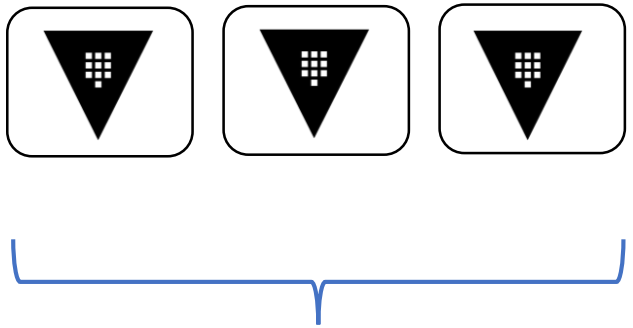
Update with:

```
"acl": {
  "enabled": true,
  "default_policy": "allow",
  "down_policy": "extend-cache“,
  “tokens”: {
     “agent”: “<token-id>”
    }
},
```

Consul Configuration File

Updating Configs with ACL Token

Consul Agent for Vault



the Consul agent running on Vault

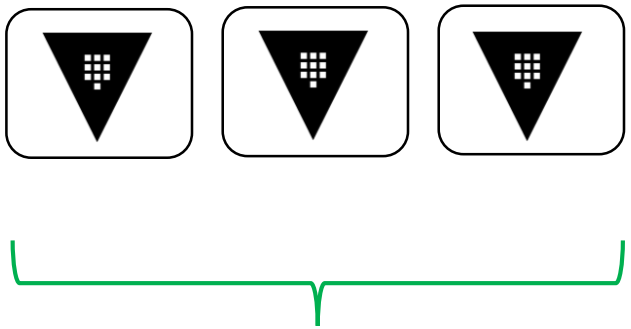
Update with:

```
"acl": {
 "tokens ": {
   "agent": "<token-id>"
    }
},
```

Consul Configuration File

Updating Configs with ACL Token

Vault Service



the Vault service running on the Vault nodes

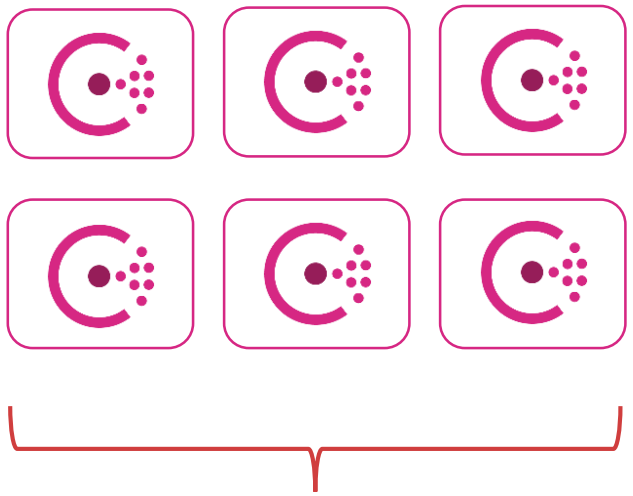
Update with:

```
storage "consul" {
  address = "127.0.0.1:8500"
  path    = "vault/“
  token = "<token_id>"
}
```

Vault Configuration File

Updating Configs with ACL Token

Consul Snapshots



the Consul snapshot agent for Consul backups (Ent Feature)

used for:

```
{
 "snapshot_agent": {
  "http_addr": "127.0.0.1:8500",
  "token": “<token_id>",
  "datacenter": "",
  "snapshot": {
    "interval": "30m",
    "retain": 336,
    "deregister_after": "8h"
  },
 "aws_storage": {
  "s3_region": "us-east-1",
  "s3_bucket": "consulsnapshots"
 }
}
}
```

Consul Snapshot Configuration File

Things to Keep in Mind

- You can’t bootstrap the Consul cluster with a default_policy of ‘deny’.
- Don’t lose the bootstrap token
- Once the default_policy parameter is set to “deny”, you’ll need to use the bootstrap token to make any changes, run any commands on Consul, etc.
- You can use –token <token_id> flag to execute a single command, or you can simply set the CONSUL_HTTP_TOKEN environment variable

Demo

Consul ACLs for Vault

Bootstrap Consul ACLs

Create ACL Policies

Create ACL Tokens

Update Configurations with Token