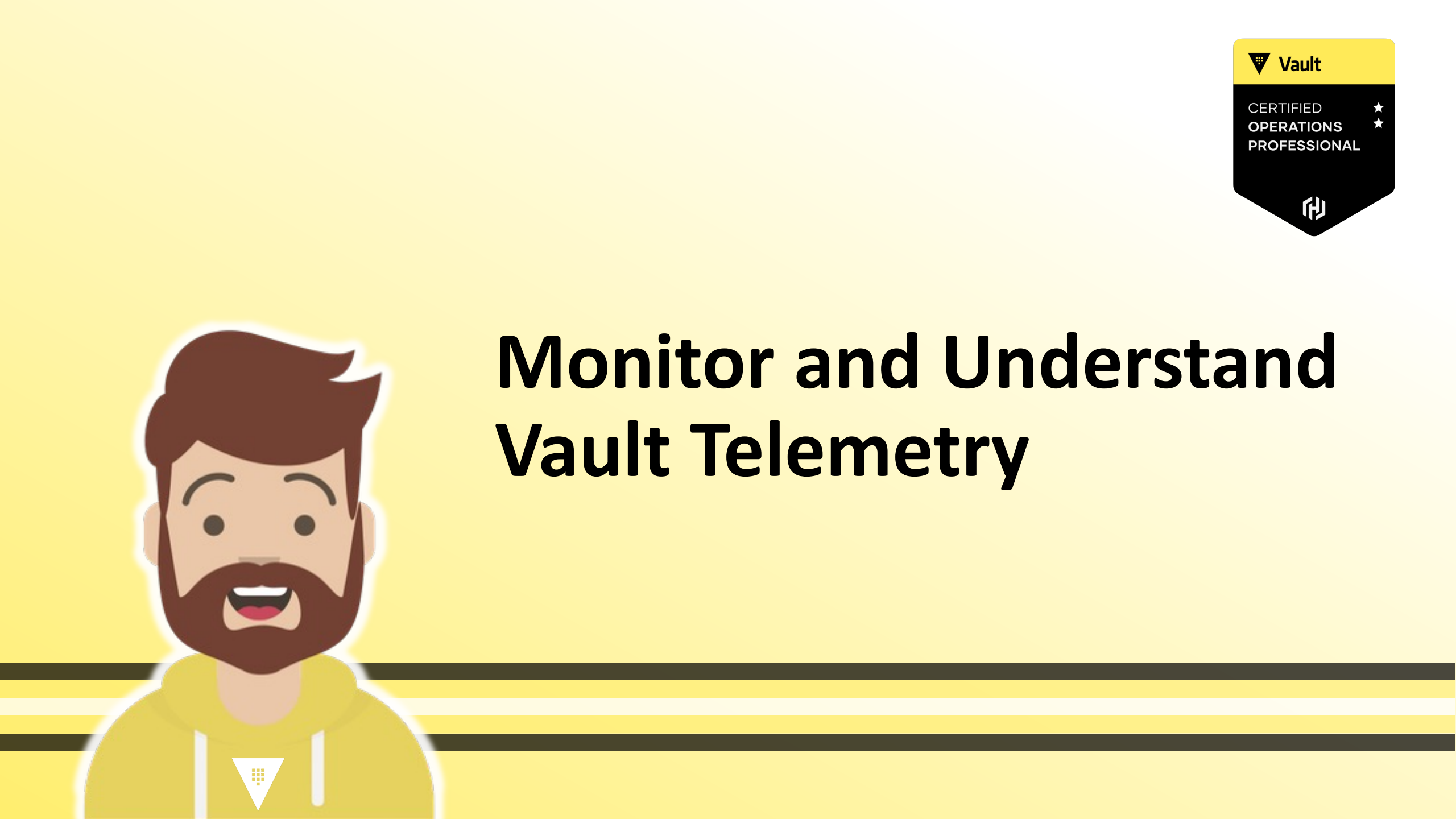


Monitor and Understand Vault Telemetry

What is Telemetry?

- The collection of various runtime metrics about the performance of different components of the Vault environment
- Can be used for debugging but it can also be used for performance monitoring and trending
- Metrics are aggregated every 10 seconds and retained for one minute
- The telemetry information is sent to a local or remote agent which generally aggregates this information to an aggregation solution, such as DataDog or Prometheus, for example

What Does Vault Support?

- Supports the following providers:
  - `statsite`
  - `statsd`
  - `circonus`
  - `dogstatsd`
  - `prometheus`
  - `stackdriver`

Example of Metrics Collected

Metric	Description
vault.core.handle_request	Duration of requests handled by Vault. This is the key measurement of Vault's response time
vault.runtime.total_gc_pause_ns	Garbage collection pause. You don't want this happening frequently or taking too long
mem.used_percent	Percentage of physical memory in use
mem.total_bytes	Total amount of physical memory available on the server
vault.audit.log_request	Duration of time taken by all audit log requests across all audit log devices
vault.policy.get_policy	Time taken to get a policy

https://www.vaultproject.io/docs/internals/telemetry

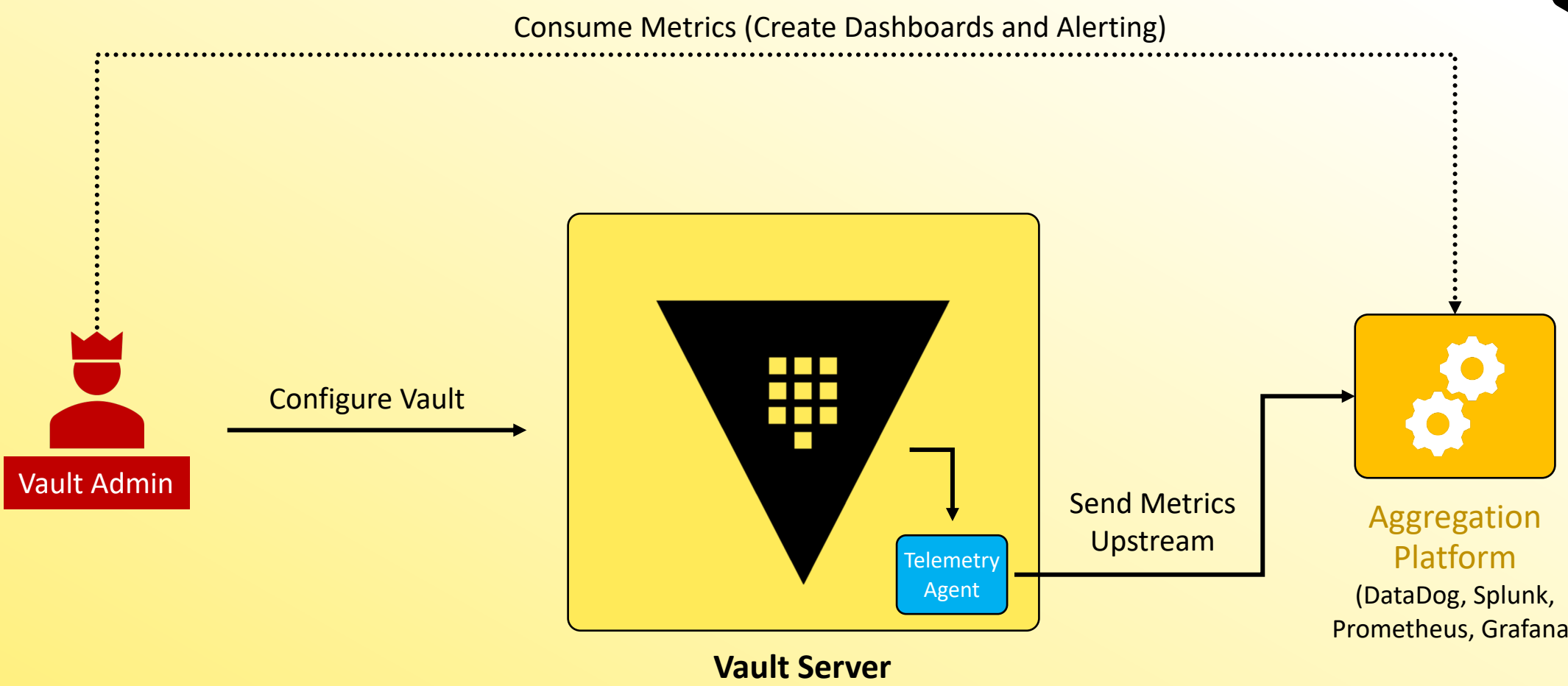
Telemetry Configuration

- Telemetry is configured in the Vault configuration file using the `telemetry` stanza
- The configuration specifies the upstream system to publish the metrics to…

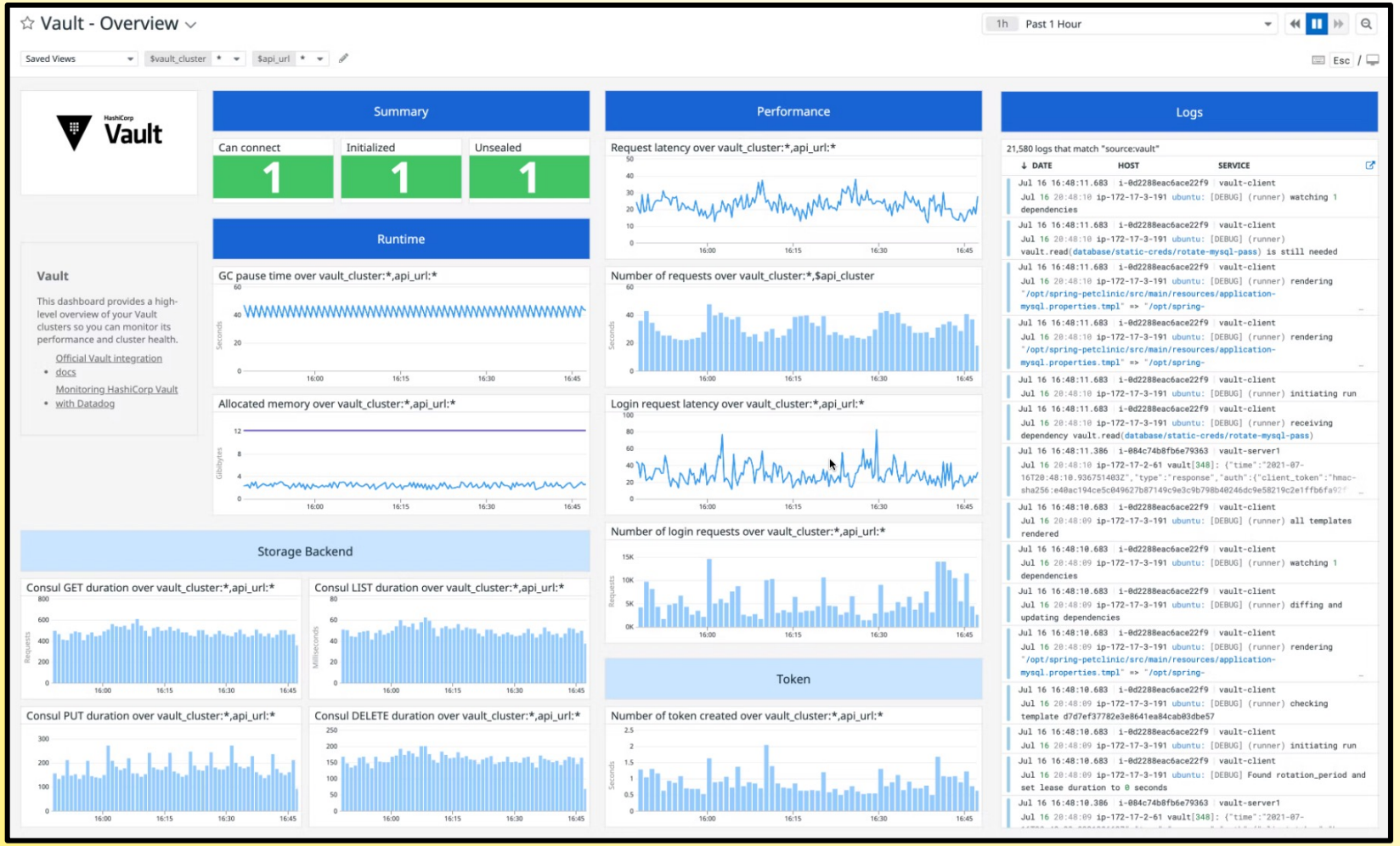
Terminal

```
…
telemetry {
  dogstatsd_addr = "metrics.hcvop.com:8125"
  dogstatsd_tags = ["vault_env:production"]
}
seal "transit" {
  address = "transit.hcvop.com:8200"
  key_name = "autounseal"
…
```

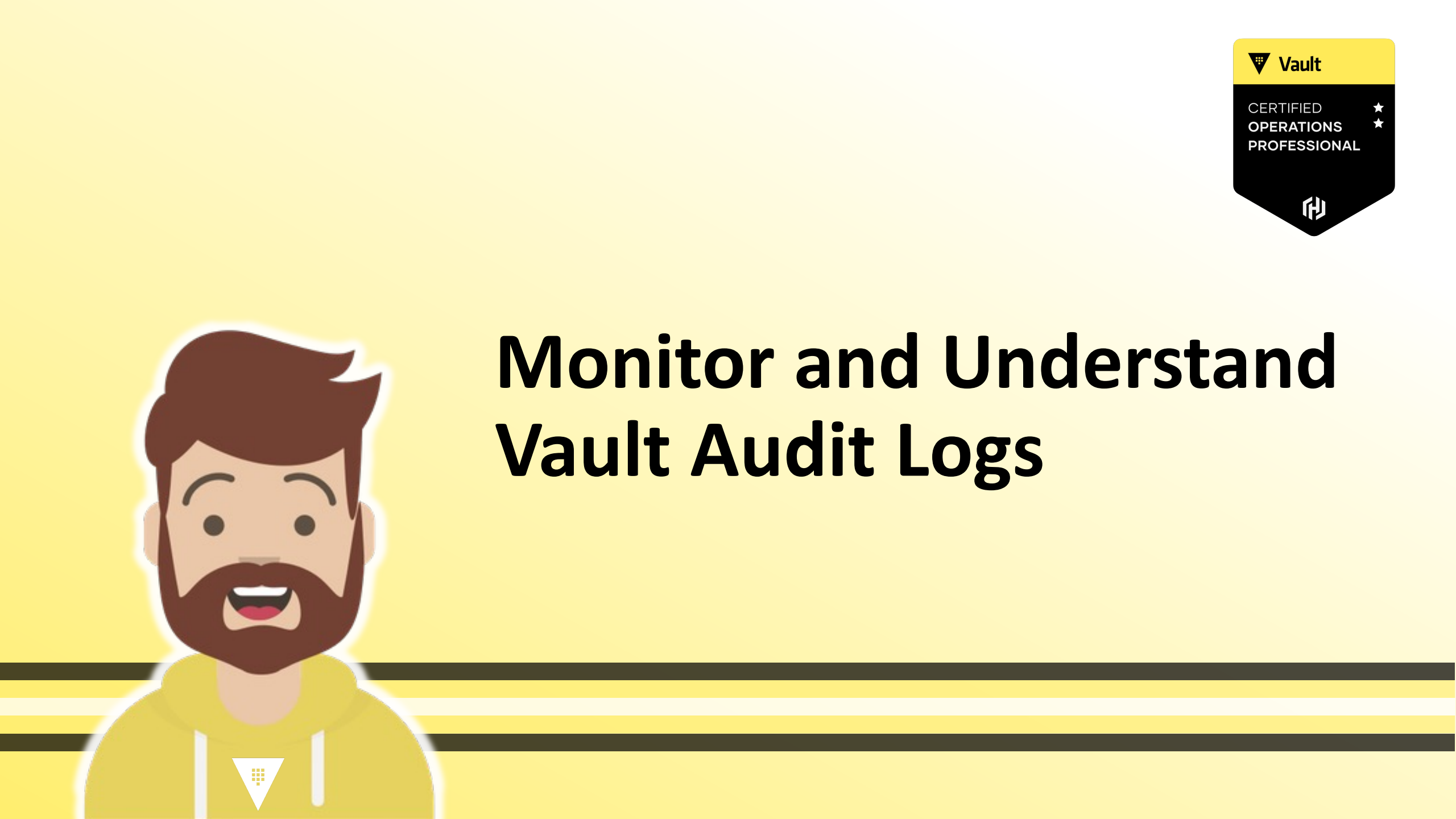
Telemetry Workflow



Dashboards & Monitoring



https://www.datadoghq.com/blog/vault-monitoring-with-datadog/



Monitor and Understand Vault Audit Logs

Introduction to Audit Devices

- Keep a detailed log of all authenticated requests and responses to Vault
- Audit log is formatted using JSON
- Sensitive information is hashed with a salt using HMAC-SHA256 to ensure secrets and tokens are never in plain text
- Log files should be protected as a user with permission can still check the value of those secrets via the /sts/audit-hash API and compare to the log file

What Audit Devices Does Vault Support?

File

- writes to a file – appends logs to the file
- does not assist with log rotation
- use fluentd or similar tool to send to collector

Syslog

- writes audit logs to a syslog
- sends to a local agent only

Socket

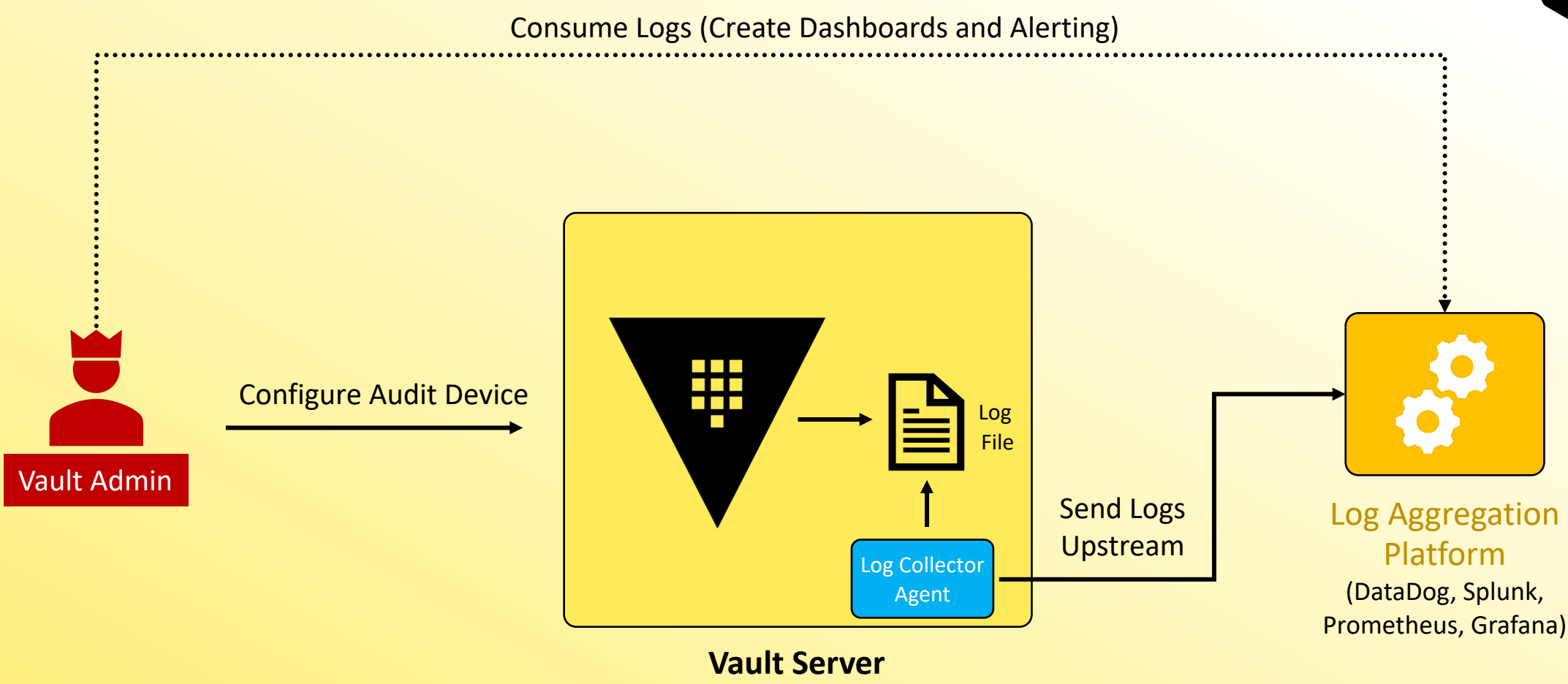
- writes to a tcp, udp, or unix socket
- TCP should be used where strong guarantees are required

Important Info about Audit Devices

- Can and should have more than one audit device enabled
- If there are any audit devices enabled, Vault requires that it can write to the log before completing the client request.
  - Prioritizes safety over availability
- If Vault cannot write to a persistent log, it will stop responding to client requests – which means Vault is down!

Vault requires at least one audit device to write the log before completing the Vault request – if enabled

Audit Log Workflow



Enabling an Audit Device

Use the `vault audit` command

Terminal

```
# Enable file audit device at default path
$ vault audit enable file file_path="/var/log/vault_audit.log
Success! Enabled the file audit device at: file/


#Enable file audit device at custom path of "logs"
$ vault audit enable -path=logs file \
   file_path="/var/log/audit.log"
Success! Enabled the file audit device at: logs/
```

Enabling an Audit Device

Use the `vault audit` command

Terminal

```
# View the audit devices currently enabled
$ vault audit list
Path     Type     Description
----     ----     -----------
file/    file     n/a
syslog/  syslog   n/a


# Disable an Audit Device
$ vault audit disable syslog/
Success! Disabled audit device (if it was enabled) at: syslog/
```

Reading an Audit Log

Terminal

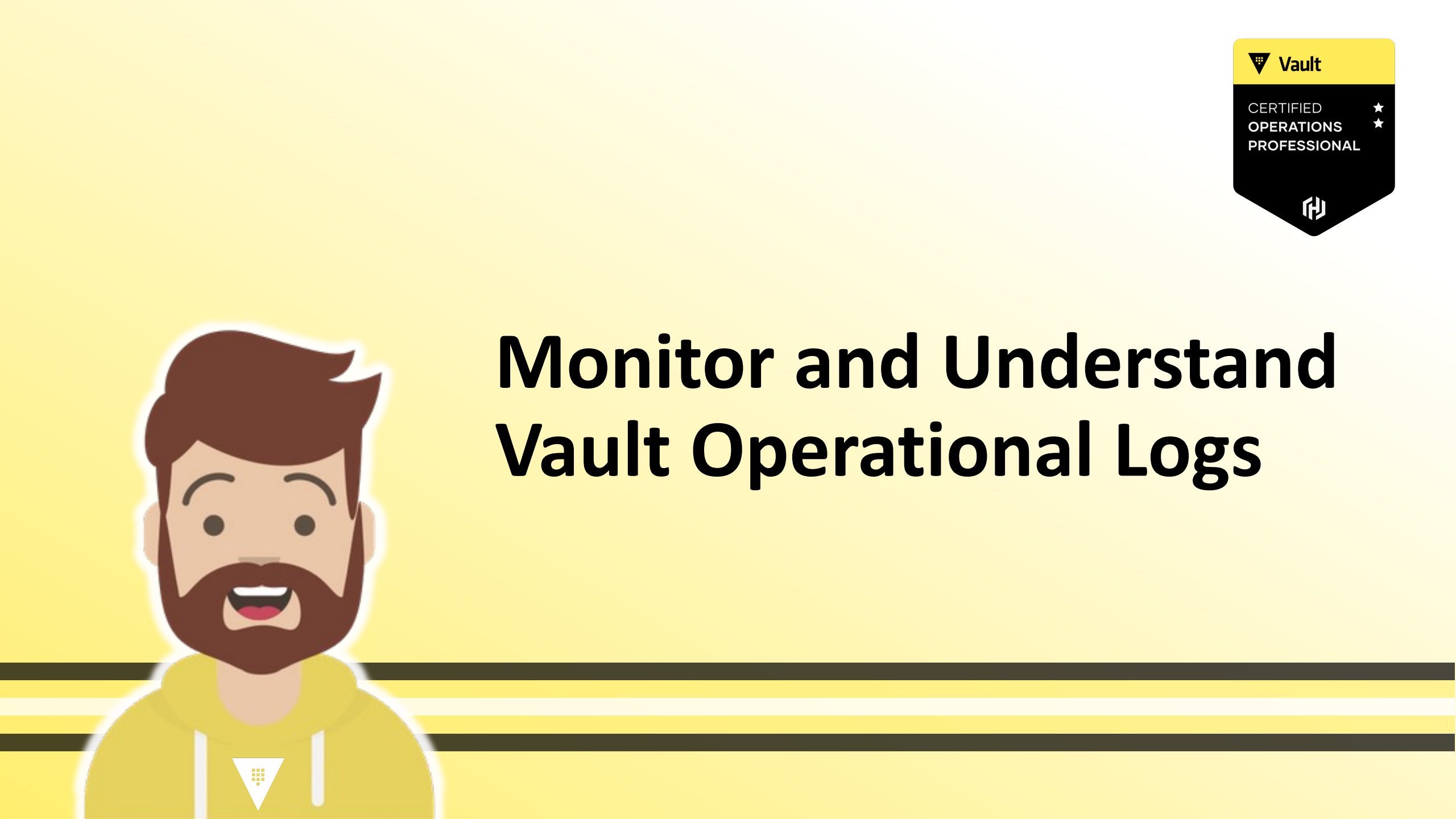
```
$ cat vault_audit.log | jq

{
  "time": "2022-12-25T21:20:12.40607Z",
  "type": "response",
  "auth": {
    "client_token": "hmac-sha256:c134d4c72a6cd891102c654b0b897f3b747a3366e88b6b2fc25247bd977ec949",
    "accessor": "hmac-sha256:e307f9f20d81fc513904534d74f5dab2348a612543271f0c2f3aa1eafe951576",
    "display_name": "root",
    "policies": [
      "root"
    ],
    "token_policies": [
      "root"
    ],
    "token_type": "service",
    "token_issue_time": "2022-12-25T11:07:35-04:00"
  },
  "request": {
    "id": "96801004-f2a5-a994-bc7a-0b15e3739db9",
    "operation": "update",
```

Permissions Needed for Audit Devices

If you need to work with an Audit Device, you need a root token or `sudo` privileges (plus the capabilities you need for the action) on the specific path

Terminal

```
# Required Permissions for interacting with the file audit device
at the default path of file/
path "sys/audit/file" {
  capabilities = ["read","create","list","update","delete","sudo"]
}
```

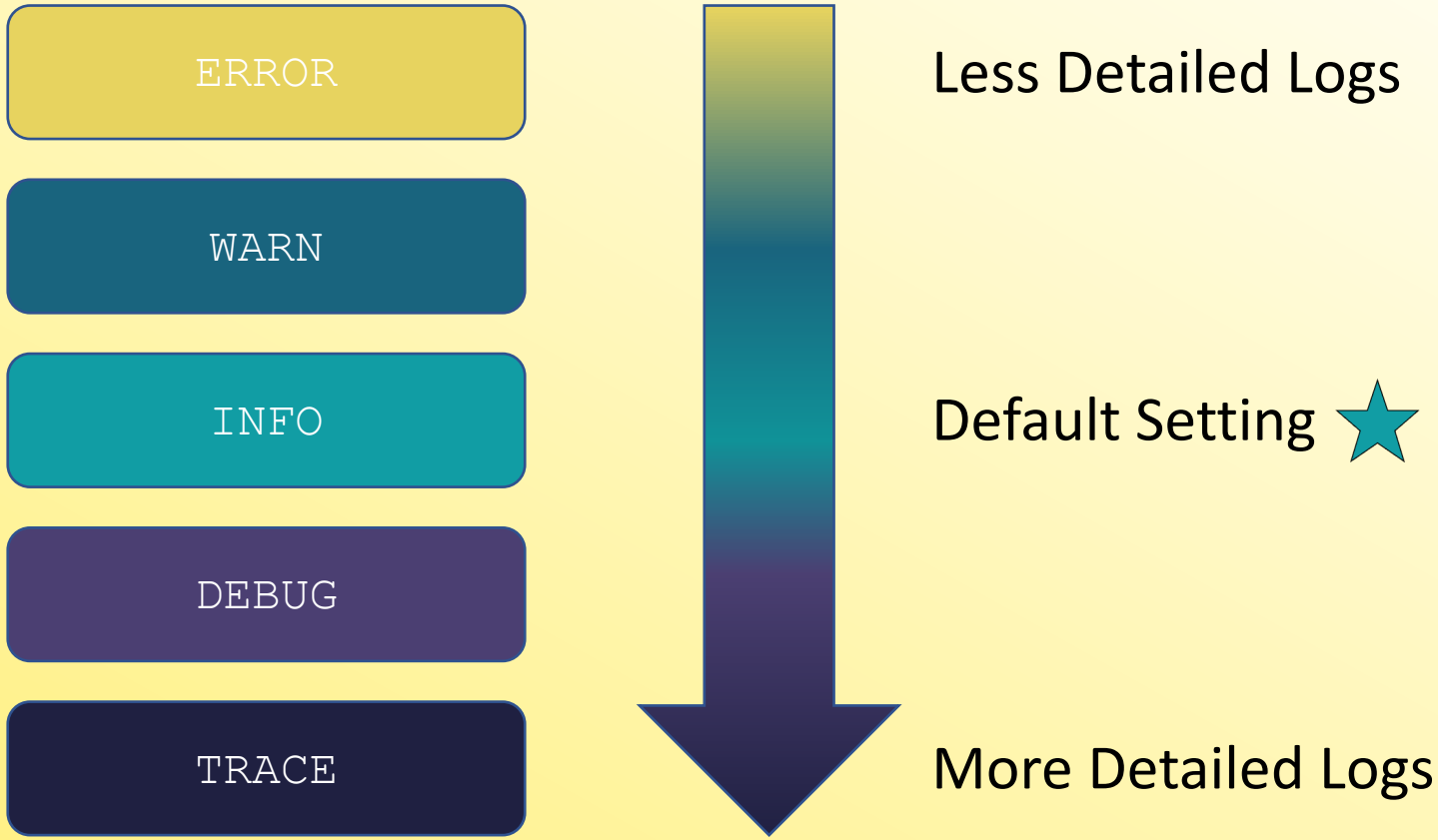


Monitor and Understand Vault Operational Logs

Vault Server Logs

- During startup, Vault will log configuration information to the log, such as listeners & ports, logging level, storage backend, Vault version, and much more….
- Once started, Vault will continue to log entries which are invaluable for troubleshooting
- The log level can be configured in multiple places in Vault, and include levels such as `err`, `warn`, `info`, `debug`, and `trace`

Vault Log Levels



Specifying the Log Level

1. Use the CLI flag **`-log_level`** when starting the Vault service

```
$ vault server -config=/opt/vault/vault.hcl -log-level=debug
```

2. Set the environment variable **`VAULT_LOG_LEVEL`**
   - Change takes effect after Vault server is restarted

```
$ export VAULT_LOG_LEVEL=trace
```

3. Set the **`log_level`** configuration parameter in the Vault configuration file
   - Change takes effect after Vault server is restarted

```
log_level=warn
```

Where Can I Get the Vault Logs?

On modern Linux distributions using systemd, you can use `journalctl` to view the logs

Terminal

```
# Read Vault logs captured by journald
$ journalctl -b --no-pager -u vault
...
Dec 25 17:01:47 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:47.950Z [DEBUG] replication.index.local: saved
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.907Z [DEBUG] rollback: attempting rollback:
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.907Z [DEBUG] rollback: attempting rollback:
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.907Z [DEBUG] rollback: attempting rollback:
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.907Z [DEBUG] rollback: attempting rollback:
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.907Z [DEBUG] rollback: attempting rollback:
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.947Z [DEBUG] replication.index.perf: saved
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.950Z [DEBUG] replication.index.local: saved
```

Page Up/Down - Scroll

Shift-G – Go to the bottom of logs

CTRL-C – Exit from journalctl

Where Can I Get the Vault Logs?

On modern Linux distributions using systemd, you can use `journalctl` to view the logs

Terminal

```
# Read Vault logs captured by journald
$ journalctl -b --no-pager -u vault
...
Dec 25 17:01:47 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:47.950Z [DEBUG] replication.index.local: saved
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.907Z [DEBUG] rollback: attempting rollback:
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.907Z [DEBUG] rollback: attempting rollback:
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.907Z [DEBUG] rollback: attempting rollback:
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.907Z [DEBUG] rollback: attempting rollback:
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.907Z [DEBUG] rollback: attempting rollback:
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.947Z [DEBUG] replication.index.perf: saved
Dec 25 17:01:52 ip-10-42-0-27 vault[7954]: 2022-12-25T17:01:52.950Z [DEBUG] replication.index.local: saved
```

Page Up/Down - Scroll   Shift-G – Go to the bottom of logs   CTRL-C – Exit from journalctl

Where Can I Get the Vault Logs?

Logs from Docker containers can be retrieved using the `docker logs` command

Terminal

```
# Read Vault logs captured by Docker
$ docker logs vault0

Couldn't start vault with IPC_LOCK. Disabling IPC_LOCK, please use --cap-add IPC_LOCK
==> Vault server configuration:

             Api Address: http://0.0.0.0:8200
                     Cgo: disabled
         Cluster Address: https://0.0.0.0:8201
              Go Version: go1.17.9
              Listener 1: tcp (addr: "0.0.0.0:8200", cluster address: "0.0.0.0:8201", max_request_duration:
"1m30s", max_request_size: "33554432", tls: "disabled")
               Log Level: info
                   Mlock: supported: true, enabled: false
           Recovery Mode: false
                 Storage: inmem
                 Version: Vault v1.10.3
             Version Sha: af866591ee60485f05d6e32dd63dde93df686dfb
```

And on the Exam?

Make sure to check out the "About the Exam" video at the end of the course to learn more about exam-specific information regarding logs