# Production Monitoring of Vault and Consul

November 2019

HashiCorp

# Document Information

## Revision History

| Date | Version | Notes |
|---|---|---|
| 21 Nov, 2019 | 1.0 | Initial publication |

## Contributors

| Name | Position |
|---|---|
| Kyle Rarey | Implementation Services Engineer |
| Johnny Carlin | Sr. Implementation Services Engineer |
| Peter Zelinski | Sr. Technical Account Manager |
| Chas Setchell | Sr. Technical Account Manager |
| James Anderton | Sr. Enterprise Architect |
| Adam Melong | Sr. DevOps Engineer - DigitalOnUs |

# Overview

There are many options to monitor in Vault and Consul. Vault and Consul provide reference documentation for all of the metrics they export in their telemetry:

- Vault - https://www.vaultproject.io/docs/internals/telemetry.html
- Consul - https://www.consul.io/docs/agent/telemetry.html

Those pages list and describe every metric that is available. Below is a list of the more important ones and what they signify. A fuller explanation is available here: https://s3-us-west-2.amazonaws.com/hashicorp-education/whitepapers/Vault/Vault-Consul-Monitoring-Guide.pdf

# Important Metrics

While it is typical to monitor for individual metrics on infrastructure components (CPU, mem, disk) as these are always good indicators of issues if they are behaving outside of expected norms, there are several more that are available via the telemetry components of Vault and Consul including but not limited to Networking, File, Backend, Replication, Leadership, and Cluster Health Metrics.

Below is a breakdown of interesting metrics and what to look for.

## Memory Usage

Memory usage should be a constant when your system is in steady state as leases are revoked and started in balance. If you see memory usage starting to grow then you should try and tie to a reason. Memory usage should not be spiky.

Look for memory usage increases starting to climb beyond your growth baseline, this can indicate something may be wrong.

## CPU Utilization

Encryption can place a heavy demand on CPU. If the CPU is too busy, Vault may have trouble keeping up with the incoming request load. You may also want to monitor each CPU individually to make sure requests are evenly balanced across all CPUs. Consul is not particularly demanding of CPU time, but a spike in CPU usage might indicate too many operations taking place at once.

- **cpu.user_cpu** Percentage of CPU being used by user processes (such as Vault or Consul)
- **cpu.iowait_cpu** Percentage of CPU time spent waiting for I/O tasks to complete

**iowait_cpu** is critical -- it means Consul is waiting for data to be written to disk, a sign that Raft might be writing snapshots to disk too often.
Look for **cpu.iowait_cpu** greater than 10%

## Network Utilization

A sudden spike in network traffic to Vault might be the result of a misconfigured client causing too many requests, or additional load you didn't plan for.

- **net.bytes_recv** Bytes received on each network interface
- **net.bytes_sent** Bytes transmitted on each network interface

Look for sudden large changes to the net metrics (greater than 50% deviation from baseline).

## File Descriptor Usage

Heavy utilization of the network due to lots of connections can often cause high utilization of file-descriptors.

- **linux_sysctl_fs.file-nr** Number of file handles being used across all processes on the host
- **linux_sysctl_fs.file-max** Total number of available file handles

Look for file-nr reaching 80% of the file-max, then you should alert and investigate

## Slow Consul response time - Vault

These metrics indicate how long it takes for Consul to handle requests from Vault

- **vault.consul.get** Count and duration of GET operations against the Consul storage backend
- **vault.consul.put** Count and duration of PUT operations against the Consul storage backend
- **vault.consul.list** Count and duration of LIST operations against the Consul storage backend
- **vault.consul.delete** Count and duration of DELETE operations against the Consul storage backend

Look for large deltas in the count, upper, or 90_percentile fields.

## Vault WAL Processing

The Vault write-ahead logs (WALs) are used to replicate Vault between clusters. Surprisingly, the WAL's are kept even if replication is not currently enabled. The WAL is purged every few seconds by a garbage collector. But if Vault is under heavy load, the WAL may start to grow, putting pressure on Consul.

- **vault.wal.persist** WALs Amount of time required to persist the Vault write-ahead logs (WAL) to the Consul backend.
- **vault.wal.flushReady** Amount of time required to flush the Vault write-ahead logs (WAL) to the persist queue

Look for flushReady over 500ms, or persistWALs is over 1000ms.

## Consul Leadership Metrics

Normally, your Consul cluster should have a stable leader. If there are frequent elections or leadership changes, it would likely indicate network issues between the Consul servers, or that the Consul servers themselves are unable to keep up with the load.

- **consul.raft.leader.lastContact** Measures the time since the leader was last able to contact the follower nodes when checking its leader lease.
- **consul.raft.state.candidate** This increments whenever a Consul server starts an election

- **consul.raft.state.leader** This increments whenever a Consul server becomes a leader

Look for candidate > 0, or leader > 0, or lastContact greater than 200ms

## Consul Cluster Health

**consul.autopilot.healthy** This tracks the overall health of the local server cluster. If all servers are considered healthy by Autopilot, this will be set to 1.

Look for values equal to 0. If any are unhealthy, this will be 0

## Consul Garbage collection

GC pause is a "stop-the-world" event, meaning that all runtime threads are blocked until GC completes. Normally these pauses last only a few nanoseconds. But if memory usage is high, the Go runtime may GC so frequently that it starts to slow down Consul.

**consul.runtime.total_gc_pause_ns** Number of nanoseconds consumed by stop-the-world garbage collection (GC) pauses since Consul started. This would be considered a warning if total_gc_pause_ns exceeds 2 seconds/minute and critical if it exceeds 5 seconds/minute.

Look for total_gc_pause_ns exceeding 2 seconds/minute

## Vault Replication Monitoring

Your Vault configuration must define the telemetry stanza to collect the telemetry. For example:

...

telemetry {

```
  dogstatsd_addr = "localhost:8125"
  disable_hostname = true
}
```

- **logshipper.streamWALs.missing_guard -** Number of incidences where the starting Merkle Tree index used to begin streaming WAL entries is not matched/found
- **logshipper.streamWALs.guard_found** - Number of incidences where the starting Merkle Tree index used to begin streaming WAL entries is matched/found
- **replication.fetchRemoteKeys -** Time taken to fetch keys from a remote cluster participating in replication prior to Merkle Tree based delta generation
- **replication.merkleDiff** - Time taken to perform a Merkle Tree based delta generation between the clusters participating in replication
- **replication.merkleSync -** Time taken to perform a Merkle Tree based synchronization using the last delta generated between the clusters participating in replication
- **Vault.wal_persistwals -** Time taken to persist a WAL to storage
- **Vault.wal_flushready -** Time taken to flush a ready WAL to storage
- **Wal.gc.total -** Total number of WAL on disk
- **Wal.gc.deleted** - Number of WAL deleted during each garbage collection run

The WAL is purged every few seconds by a garbage collector, but if Vault is under heavy load, the WAL may start to grow, putting a lot of pressure on the storage backend (Consul). To detect back pressure from a slow storage backend, monitor the **vault.wal_flushready** and **vault.wal_persistwals** metrics.


USA Headquarters

101 Second St., Suite 575, San Francisco, Ca, 94105