Exam Objective 1

Explain Consul Architecture

Explain Consul Architecture

Objective 1a: Identify the components of Consul datacenter, including agents and communication protocols

Objective 1b: Prepare Consul for high availability and performance

Objective 1c: Identify Consul's core functionality

Objective 1d: Differentiate agent roles

1 [✓ 2] 3 4 5

Difficulty Level

HashiCorp Suite of Tools

HashiCorp Terraform

HashiCorp Vault

HashiCorp Consul

HashiCorp Nomad

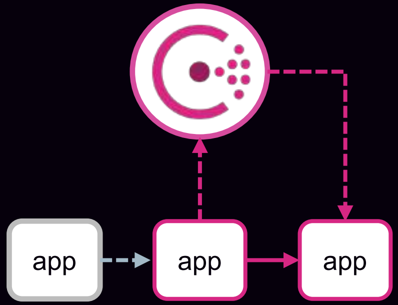
HashiCorp Boundary

HashiCorp Packer

HashiCorp Vagrant

HashiCorp Waypoint

Cloud networking automation for dynamic infrastructure



HashiCorp

Consul

Service Discovery

Service Segmentation

Service Configuration

Consul OSS vs. Enterprise

Open Source

- ✓ Service Discovery
- ✓ Service Segmentation
- ✓ Layer 7 Traffic Mgmt
- ✓ K/V Storage
- ✓ Mesh Gateways
- ✓ Application Aware Intentions

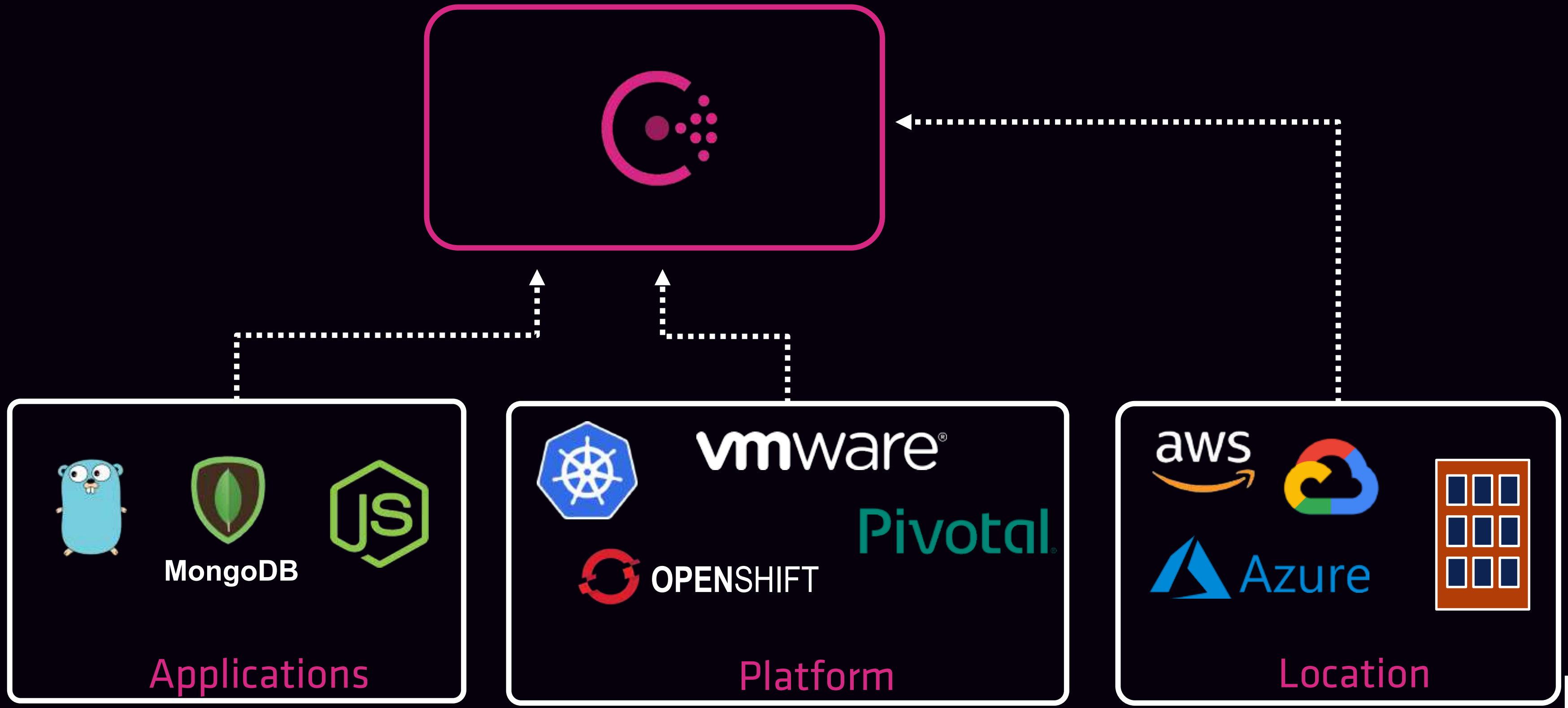
Enterprise

- ✓ Automated Backups
- ✓ Automated Upgrades

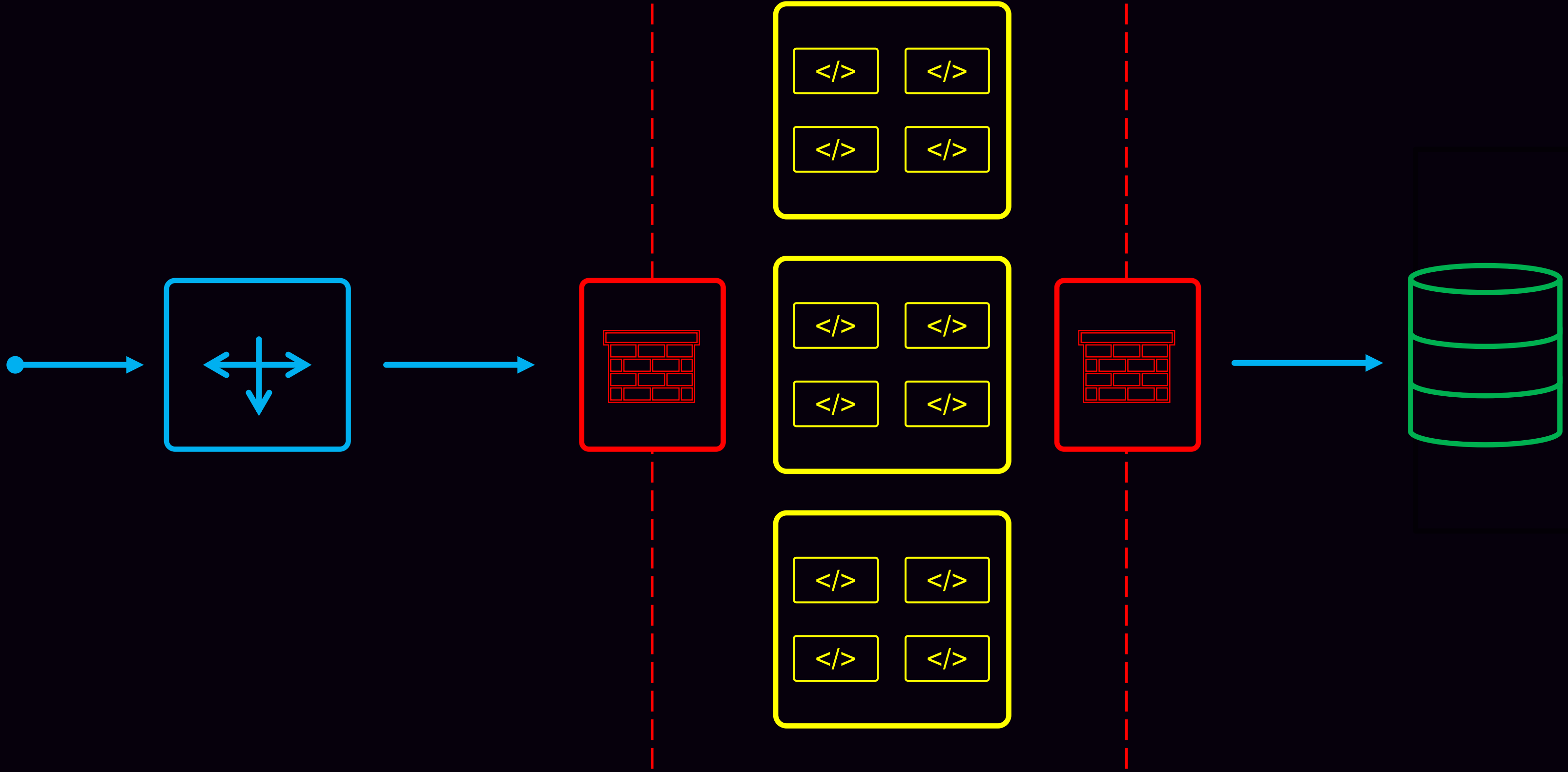
Optional Modules

- ✓ Network Segments
- ✓ Federation
- ✓ Enhanced Read Scalability
- ✓ Redundancy Zones
- ✓ Namespaces
- ✓ SSO
- ✓ Audit Logging

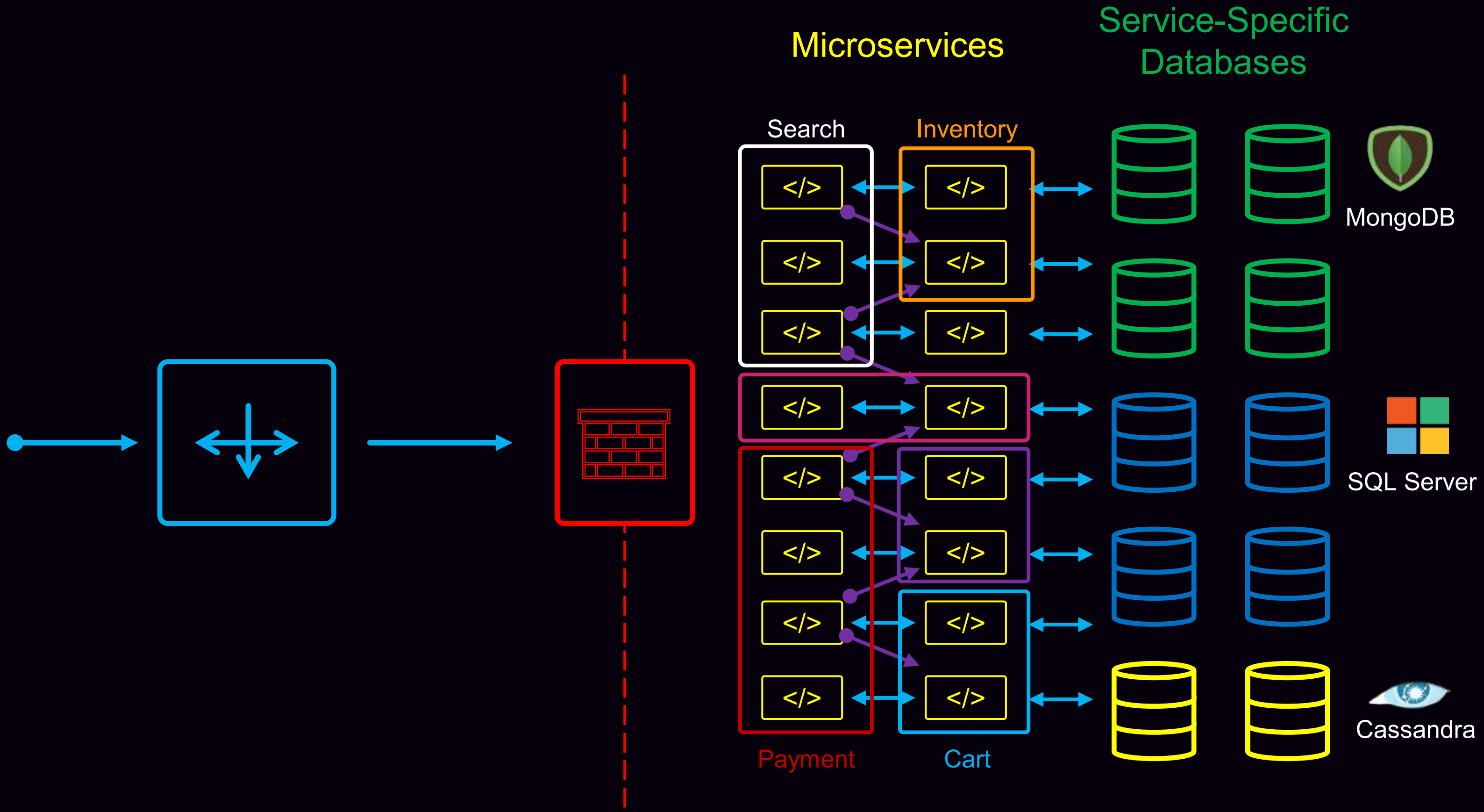
Why Use Consul?



Traditional Monolith



Shift to Microservices



Core Features of Consul

- Dynamic Service Registration
- Service Discovery
- Distributed Health Checks
- Centralized K/V Storage
- Access Control Lists
- Segmentation of Services
- Cross Cloud/Data Center Availability
- HTTP API, UI, and CLI Interfaces

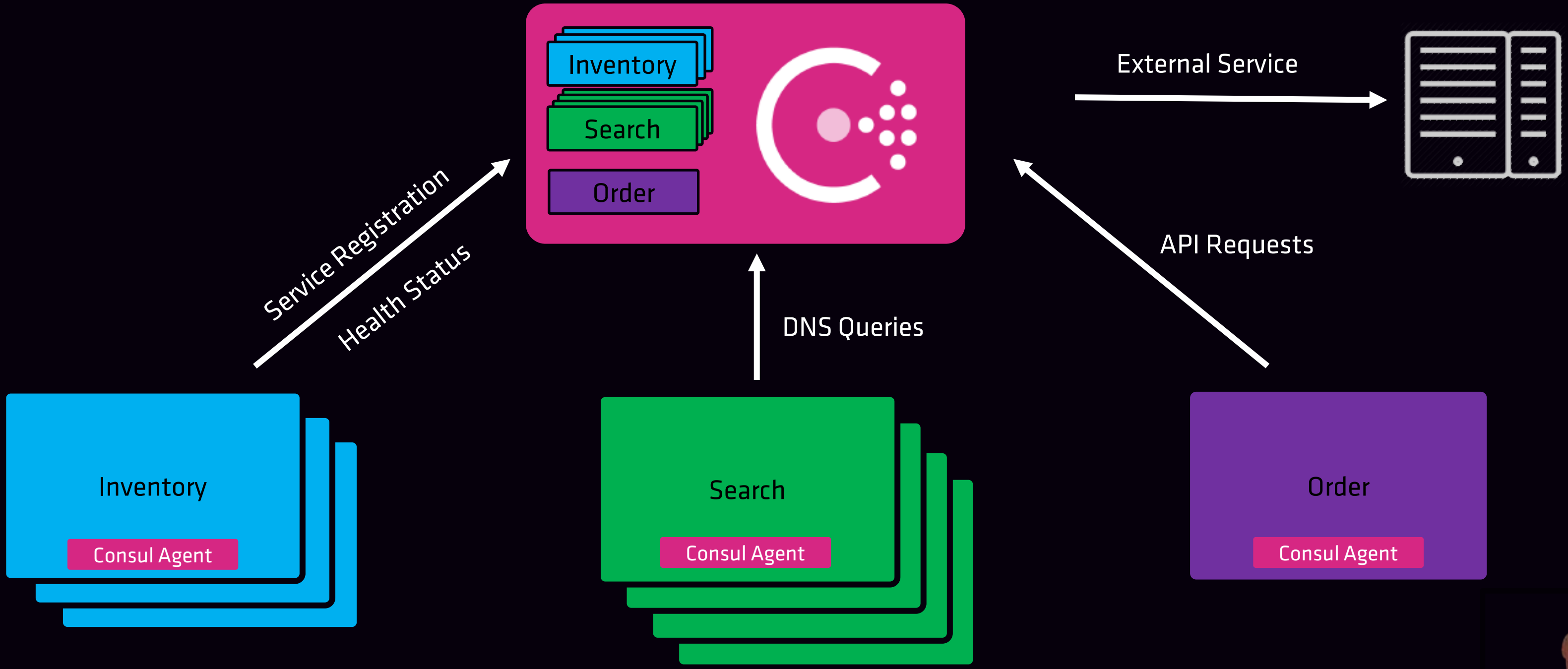
Service Discovery!

- Centralized Service Registry
  - Single point of contact for services to communicate to other services
  - Important for dynamic workloads (such as containers)
  - Especially important for a microservice architecture
- Reduction or elimination of load balancers to front-end services
  - Frequently referred to as east/west traffic

Service Discovery!

- Real-time health monitoring
  - Distributed responsibility throughout the cluster
  - Local agent performs query on services
    - Node-level health checks
    - Application-level health checks

Service Discovery



Inventory

Search

Order

External Service

Service Registration

Health Status

DNS Queries

API Requests

Inventory

Consul Agent

Search

Consul Agent

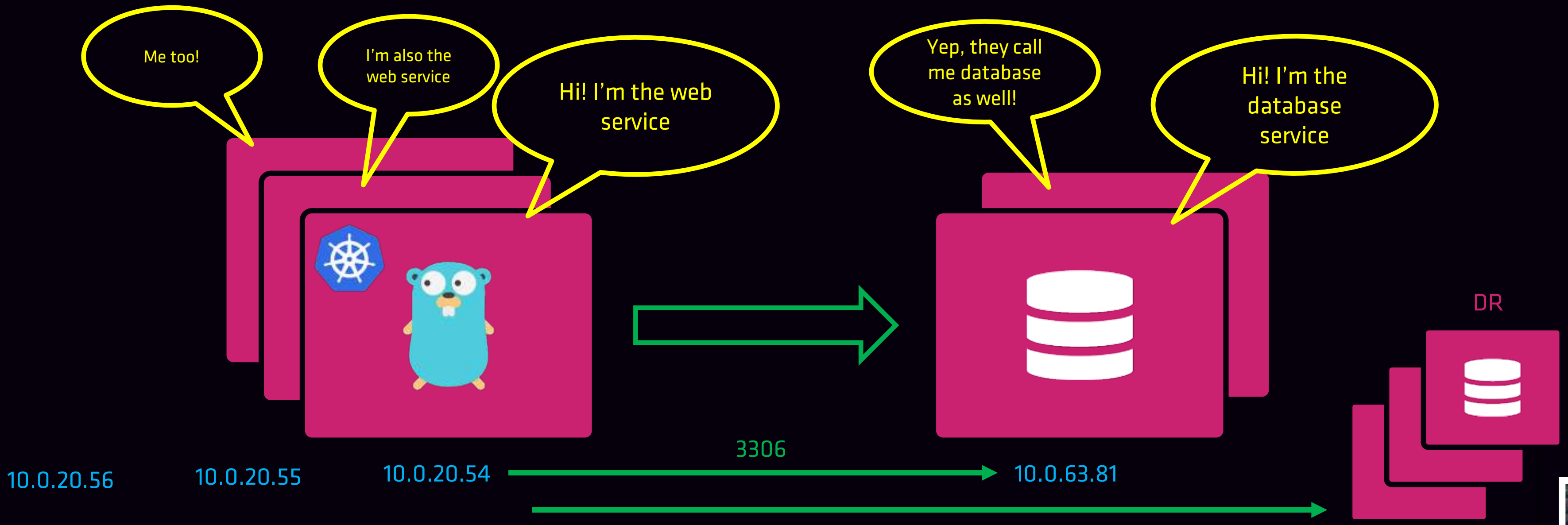
Order

Consul Agent

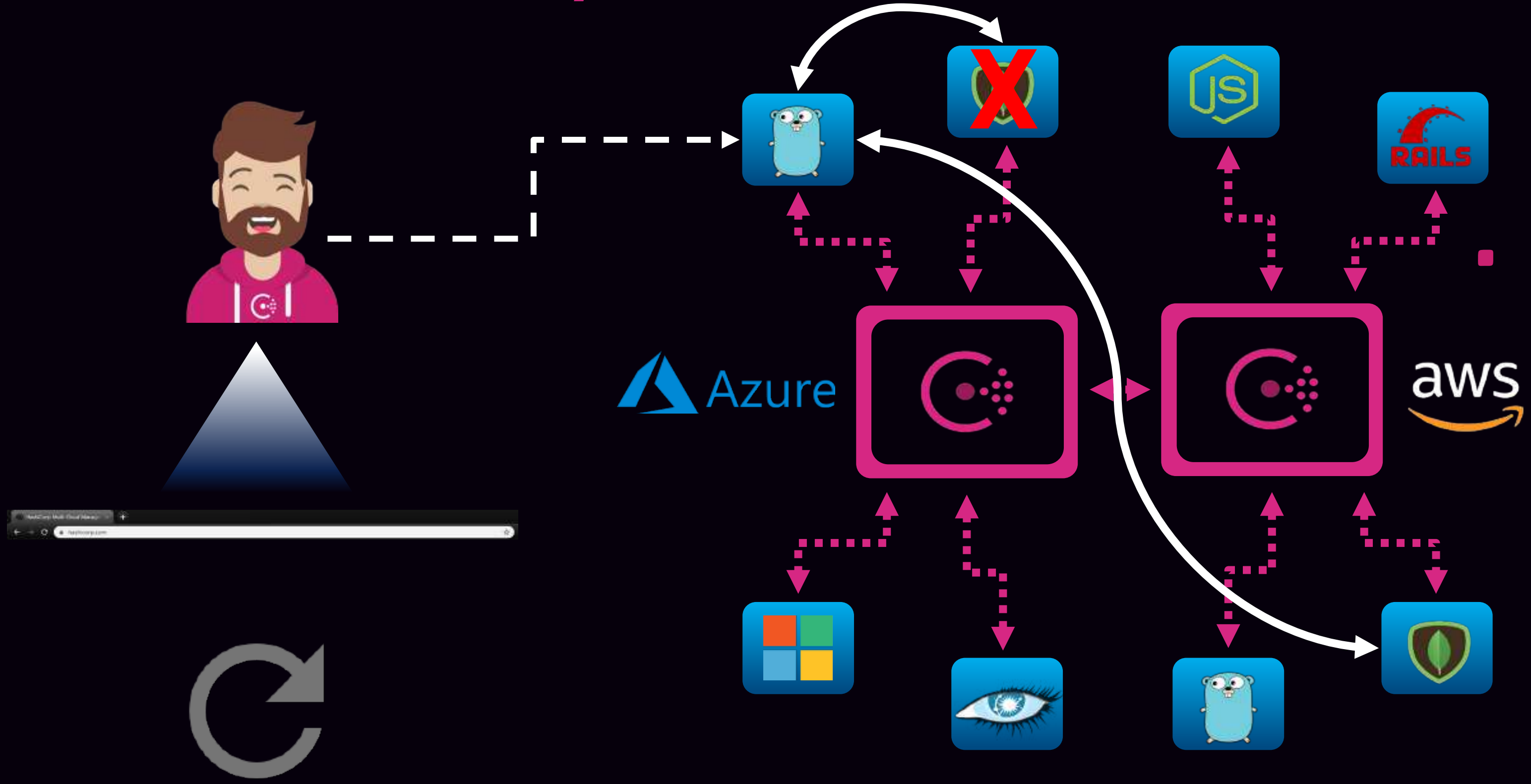
Scale to Thousands and Thousands of Nodes

Service Discovery

- Automate networking and security using identity-based authorization
  - no more IP-based or firewall-based security



Service Discovery – Multi-DC

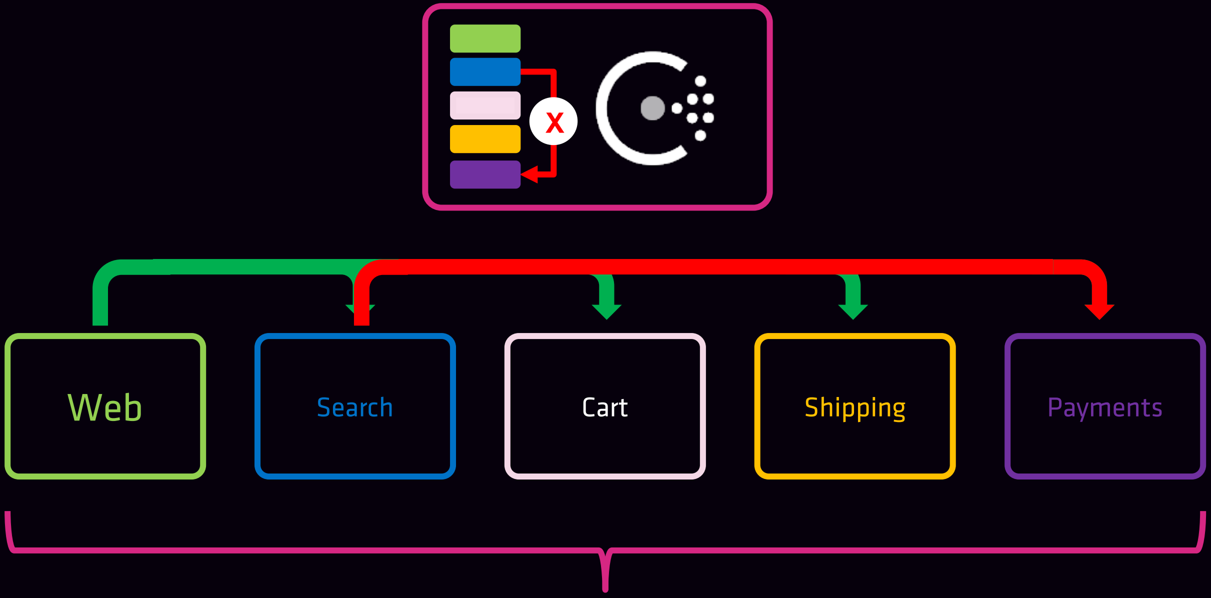


Service Mesh

- Enables secure communication between services
  - Integrated mTLS secures communication
  - Uses sidecar architecture that is placed alongside the registered service
  - Sidecar (Envoy, etc.) transparently handles inbound/outbound connections

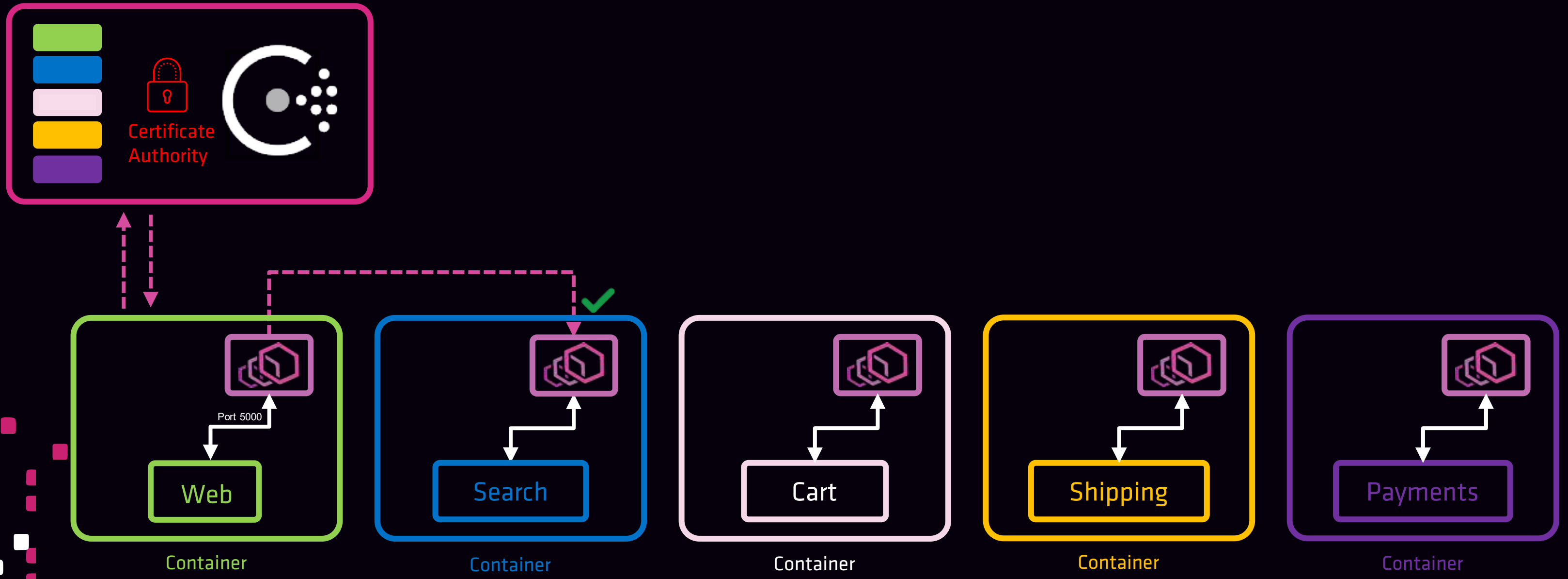
- Defined access control for services
  - Defines which service can establish connections to other service

Service Mesh



eCommerce Site

Service Mesh

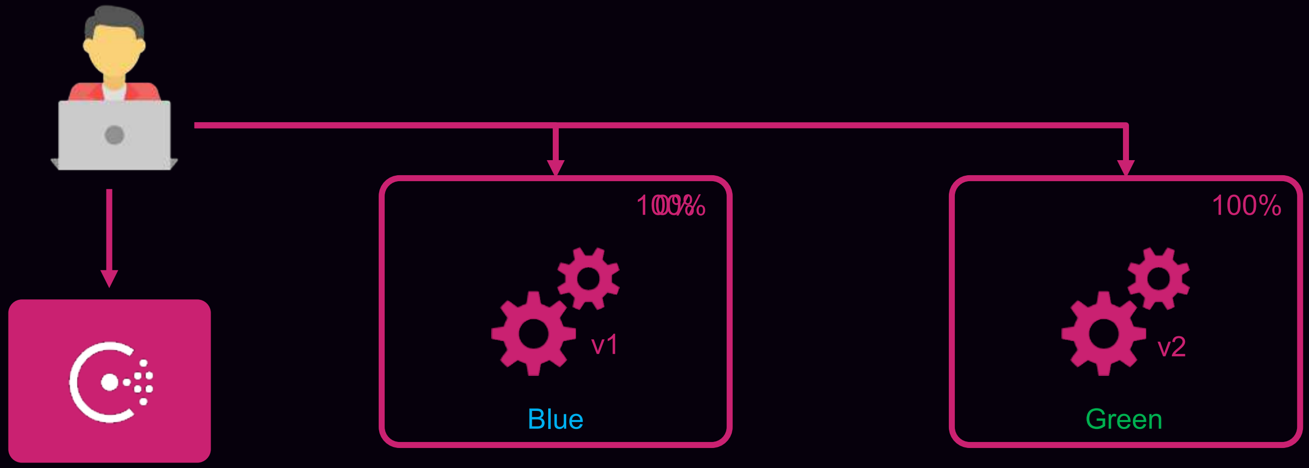


Network Automation

- Dynamic load balancing among services
  - Consul will only send traffic to healthy nodes & services
  - Use traffic-shaping to influence how traffic is sent
- Extensible through networking partners
  - F5, nginx, haproxy, Envoy
- Reduce downtime by using multi-cloud and failover for services

Network Automation

- L7 traffic management based on your workloads and environment
  - service failover, path-based routing, and traffic shifting capabilities

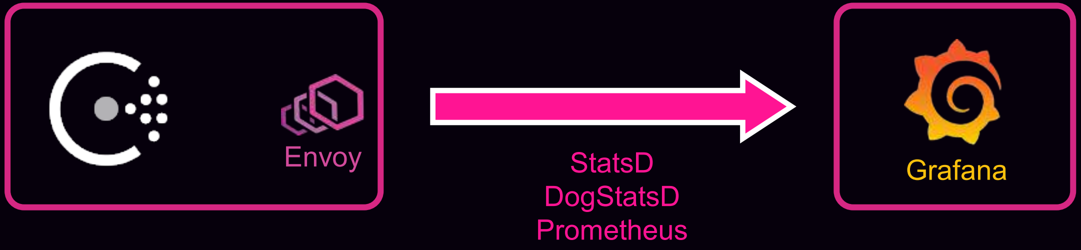


```
Kind = "service-splitter"
Name = “web-app"

Splits = [
 {
  Weight       = 100
  ServiceSubset = “v2"
 },
 {
  Weight       = 0
  ServiceSubset = "v1"
 },
]
```

Network Automation

- Increased L7 visibility between services
    - View metrics such as connections, timeouts, open circuits, etc.



Envoy

StatsD
DogStatsD
Prometheus

Grafana

Service Configuration

- Consul provides a distributed K/V store
- All data is replicated across all Consul servers
  - Can be used to store configuration and parameters
  - It is NOT a full featured datastore (like DynamoDB)

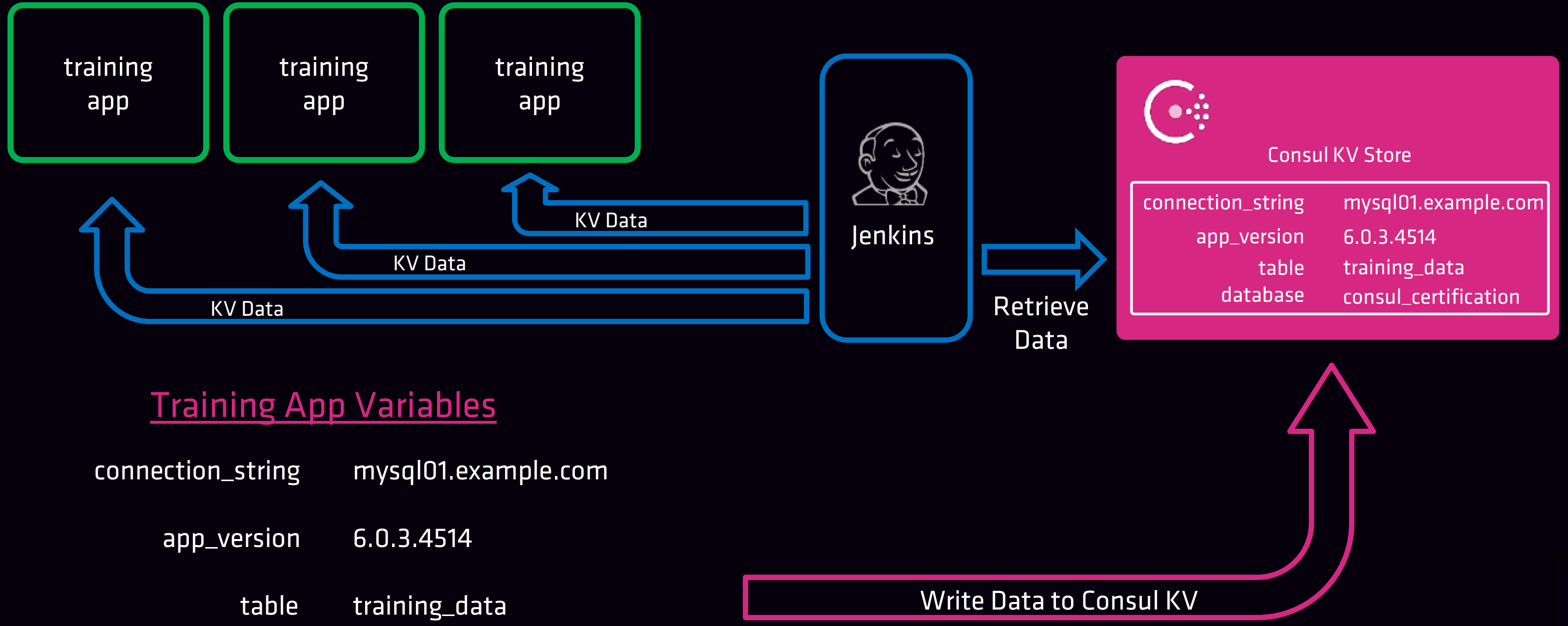
- Can be accessed by any agent (client or server)
  - Accessed using the CLI, API, or Consul UI
  - Make sure to enable ACLs to restrict access (Objective 8)

Service Configuration

- No restrictions on the type of object stored
- Primary restriction is the object size – capped at 512 KB

- Doesn't use a directory structure, although you can use / to organize your data within the KV store
  - / is treated like any other character
  - This is different than Vault where / signifies a path

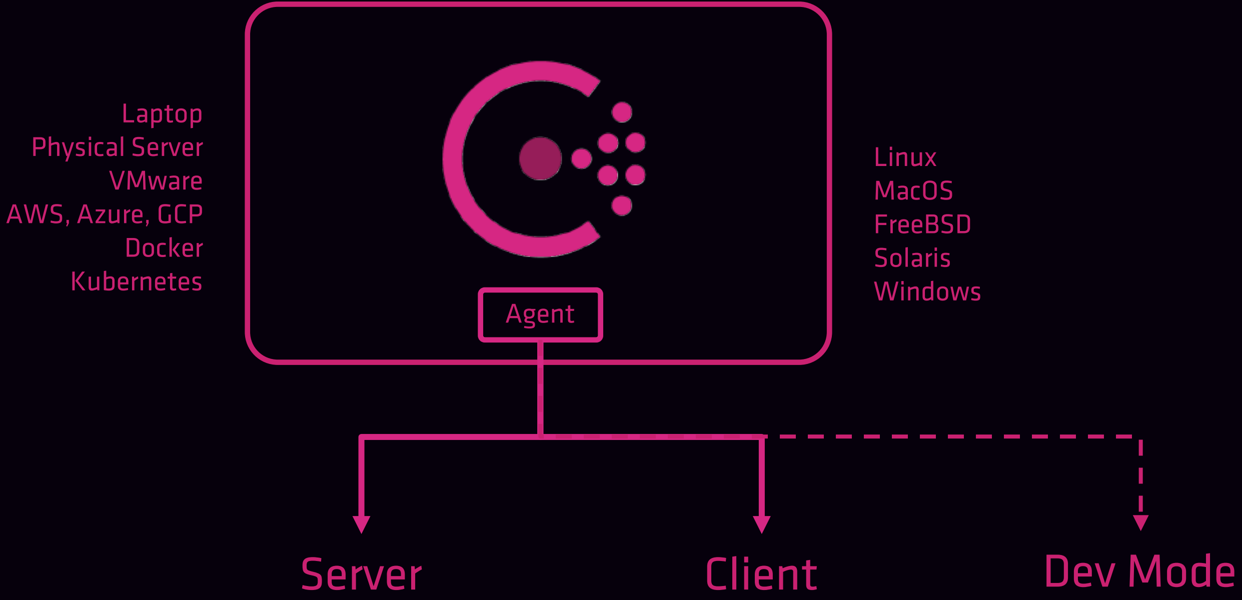
Service Configuration



Training App Variables

connection_string	mysql01.example.com
app_version	6.0.3.4514
table	training_data
database	consul_certification

Consul Basics



Laptop
Physical Server
VMware
AWS, Azure, GCP
Docker
Kubernetes

Agent

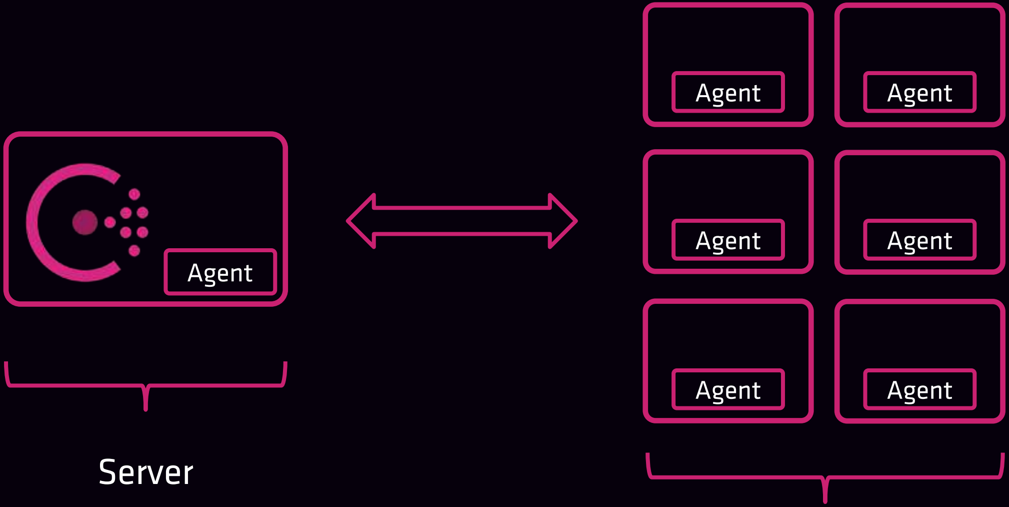
Linux
MacOS
FreeBSD
Solaris
Windows

Server

Client

Dev Mode

Agent Modes



Server

Also Known As:

- Server Mode
- Server Agent
- Consul Node

Client

Also Known As:

- Client Mode
- Client Agent

Server vs. Client Mode

Server

Agent

Consul (cluster) State
Membership
Responds to Queries
Registers Services
Maintains Quorum
Acts as Gateway to other DCs

vs

Client

Agent

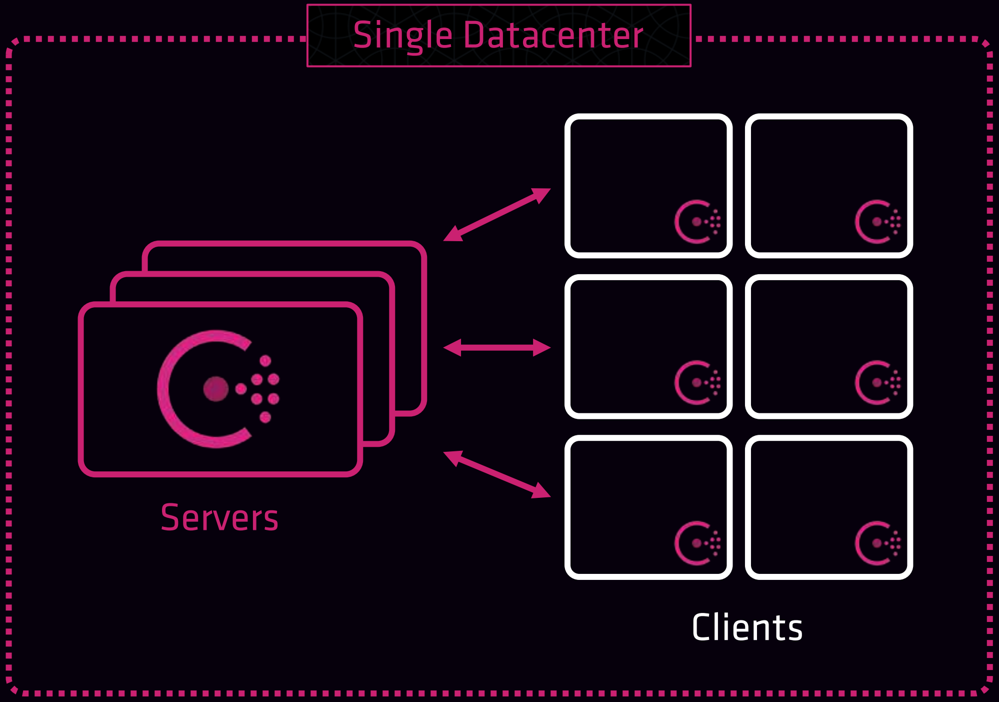
Register Local Services
Perform Health Checks
Forwards RPC calls to Servers
Takes Part in LAN Gossip Pool
Relatively Stateless

Dev

Agent

Used Only for Testing/Demo
Runs as a Consul Server
Not Secure or Scalable
Runs Locally
Stores Everything in Memory
Does Not Write to Disk

Single Datacenter



Single Datacenter

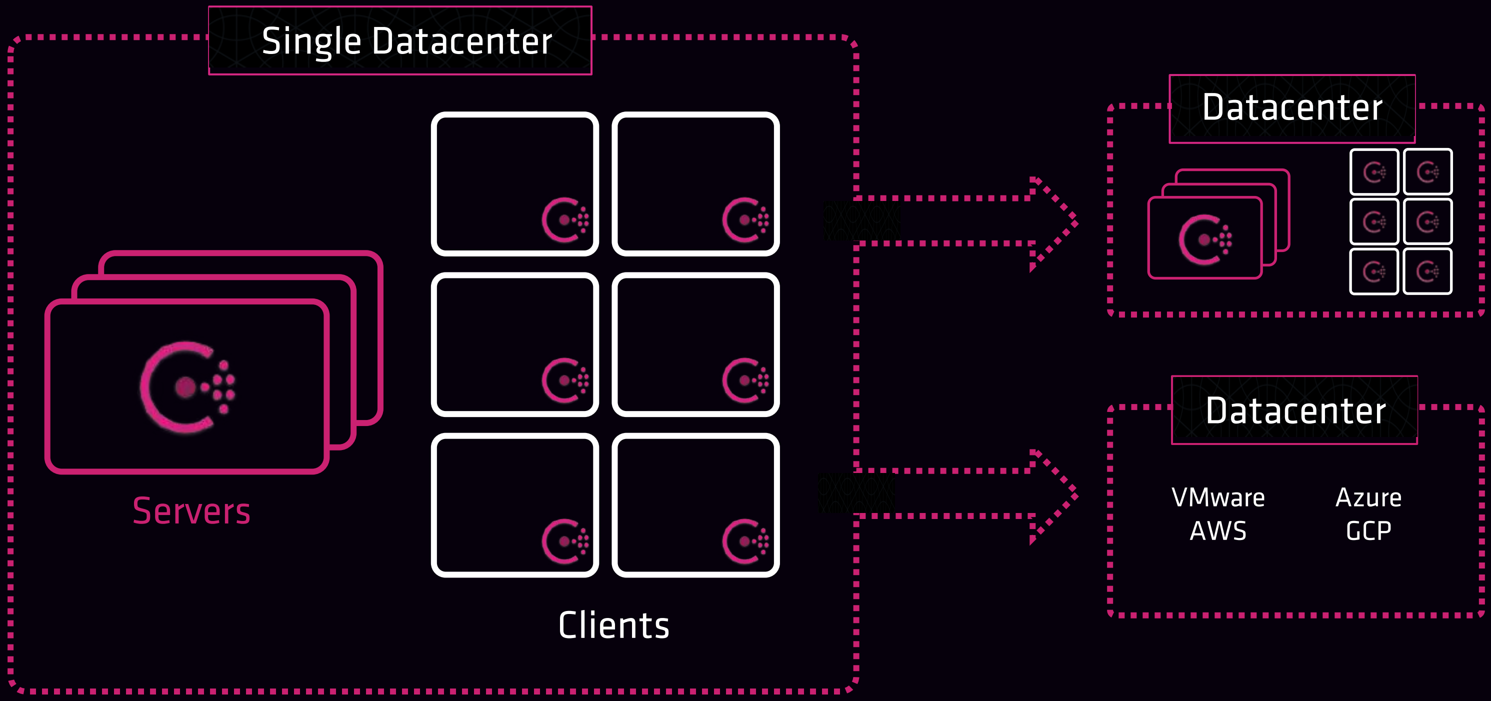
☑ What Is a Datacenter?

- single-cluster
- private
- low latency
- high bandwidth
- contained in a single location
- multi-AZ is acceptable
- uses the LAN gossip pool

☒ What a Datacenter Is Not!

- multi-cloud or location
- multiple Consul clusters
- uses the WAN gossip pool
- communicates via WAN or Internet

Multi-Datacenter

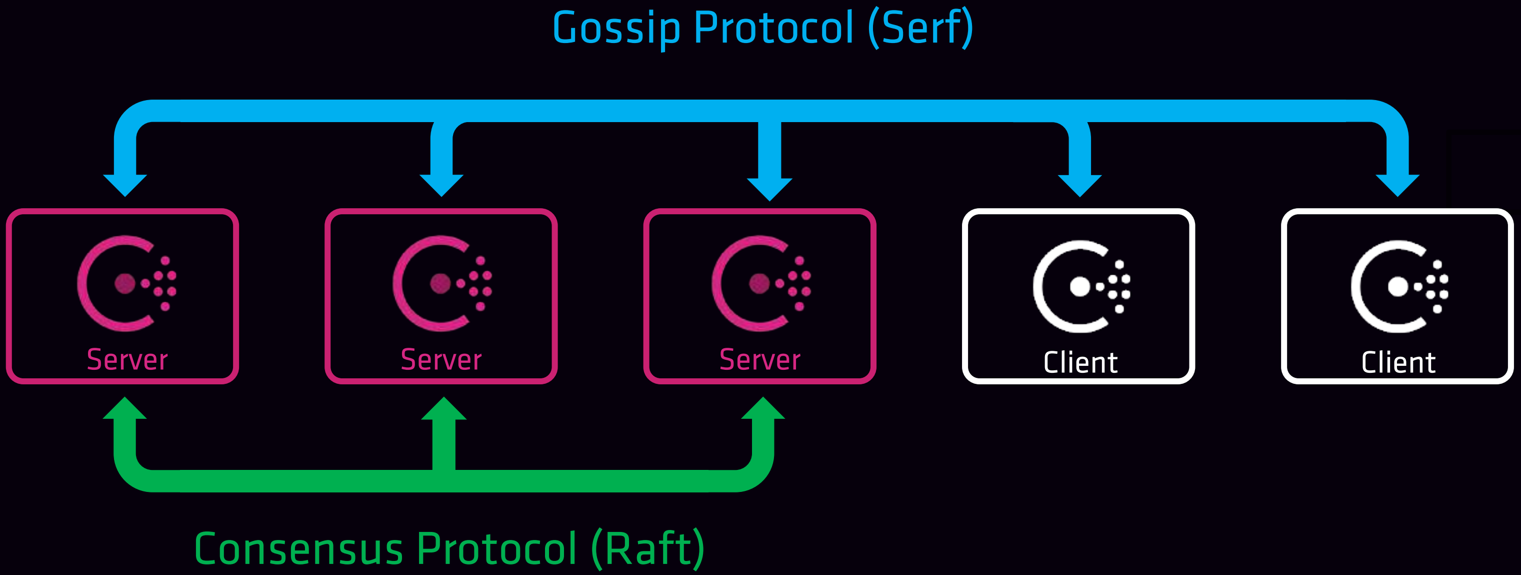


Multi-Datacenter

☑ What Is Multi-Datacenter?

- multi-cloud, multi-region, location, or cluster
- multiple Consul cluster federation
- uses the WAN gossip pool
- communicates via WAN or Internet
- WAN federation through mesh gateways

Key Protocols



Consensus Protocol

- Based on Raft
  - Used on only Server nodes (cluster) – <u>not clients</u>
  - Strongly consistent

- Responsible for:
  - Leadership elections
  - Maintaining committed log entries across server nodes
  - Establishing a quorum

Consensus Glossary

- Log
  - Primary unit of work – an ordered sequence of entries
  - Entries can be a cluster change, key/value changes, etc.
  - All members must agree on the entries and their order to be considered a consistent log

- Peer Set
  - All members participating in log replication
  - In Consul's case, all servers nodes in the local datacenter

Consensus Glossary

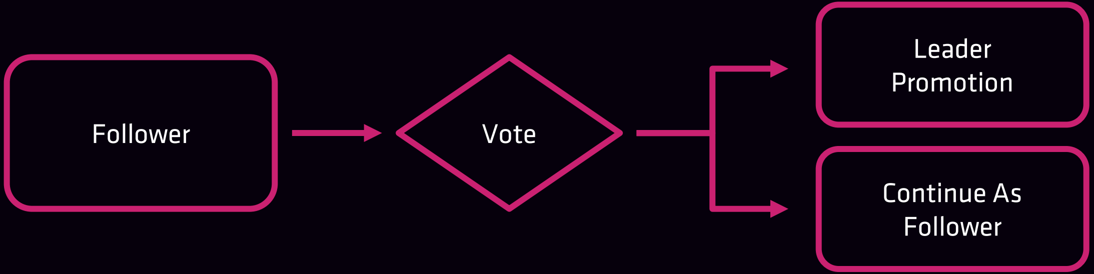
- Quorum
  - Majority of members of the peer set (servers)
  - No quorum = no Consul

  - A quorum requires at least (n+1)/2 members
    - Five-node cluster = (5+1)/2 = 3
    - Three-node cluster = (3+1)/2 = 2

Consensus Protocol

Raft nodes are always in one of three states:

- Leader
- Follower
- Candidate



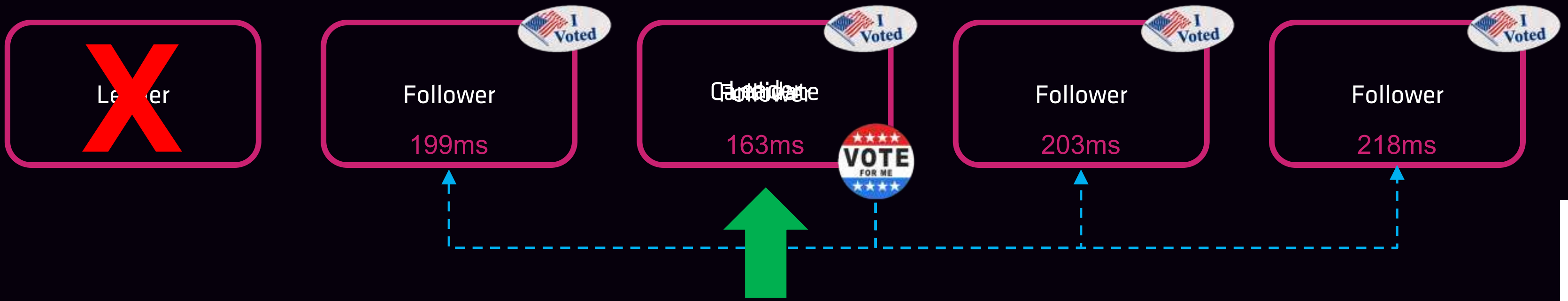
Consensus Protocol

- Leader is responsible for:
  - Ingesting new log entries
  - Processing all queries and transactions
  - Replicating to followers
  - Determining when an entry is considered committed

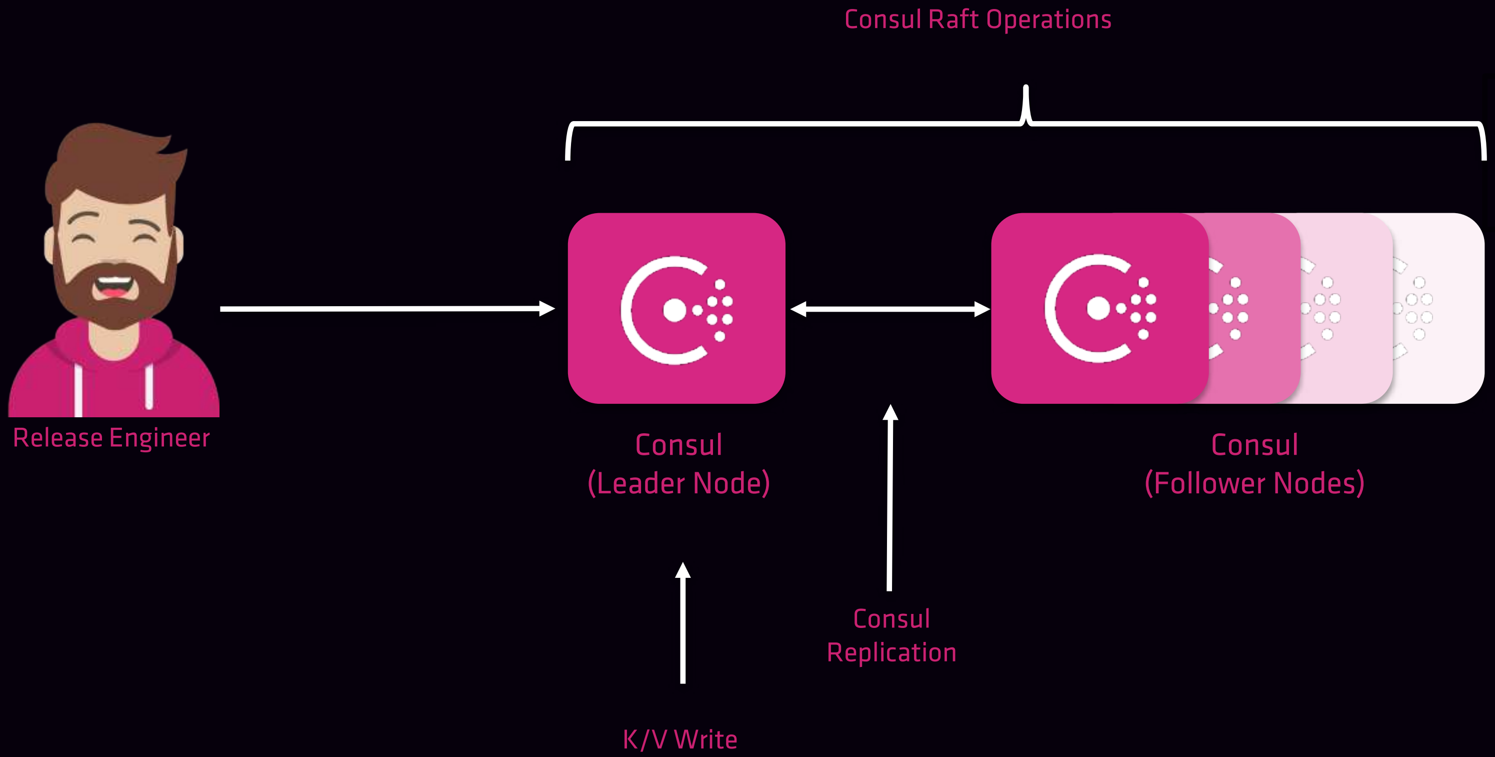
- Follower is responsible for:
  - Forwarding RPC request to the leader
  - Accepting logs from the leader
  - Casting votes for leader election

Consensus Protocol – Leader Election

- Leadership is based on randomized election timeouts
  - Leader sends out frequent heartbeats to follower nodes
  - Each server has a randomly assigned timeout (e.g., 150ms – 300ms)
  - If a heartbeat isn't received from the leader, an election takes place
  - The node changes its state to candidate, votes for itself, and issues a request for votes to establish majority



Consensus Protocol



Gossip Protocol

- Based on Serf
  - Used cluster wide – including multi-cluster
  - Used by clients and servers
- Responsible for:
  - Manage membership of the cluster (clients and servers)
  - Broadcast messages to the cluster such as connectivity failures
  - Allows reliable and fast broadcasts across datacenters
  - Makes use of two different gossip pools
    - LAN
    - WAN

Gossip Protocol

LAN Gossip Pool

- Each datacenter has its own LAN gossip pool
- Contains all members of the datacenter (clients & servers)

Purpose

- Membership information allows clients to discover servers
- Failure detection duties are shared by members of the entire cluster
- Reliable and fast event broadcasts

Gossip Protocol

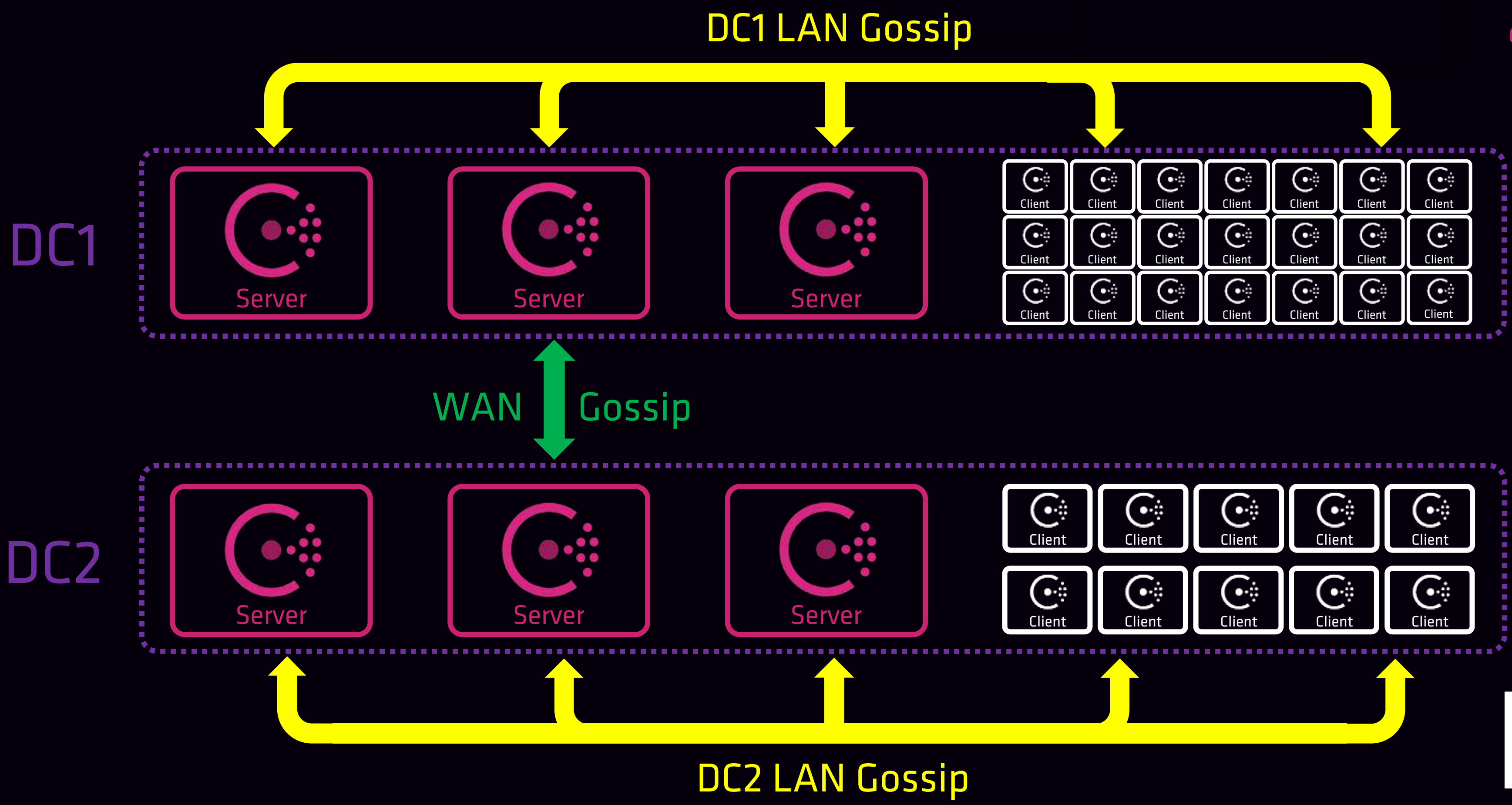
WAN Gossip Pool

- Separate, globally unique pool
- All servers participate in the WAN pool regardless of datacenter

Purpose

- Allows servers to perform cross datacenter requests
- Assists with handling single server or entire datacenter failures

Gossip Protocol

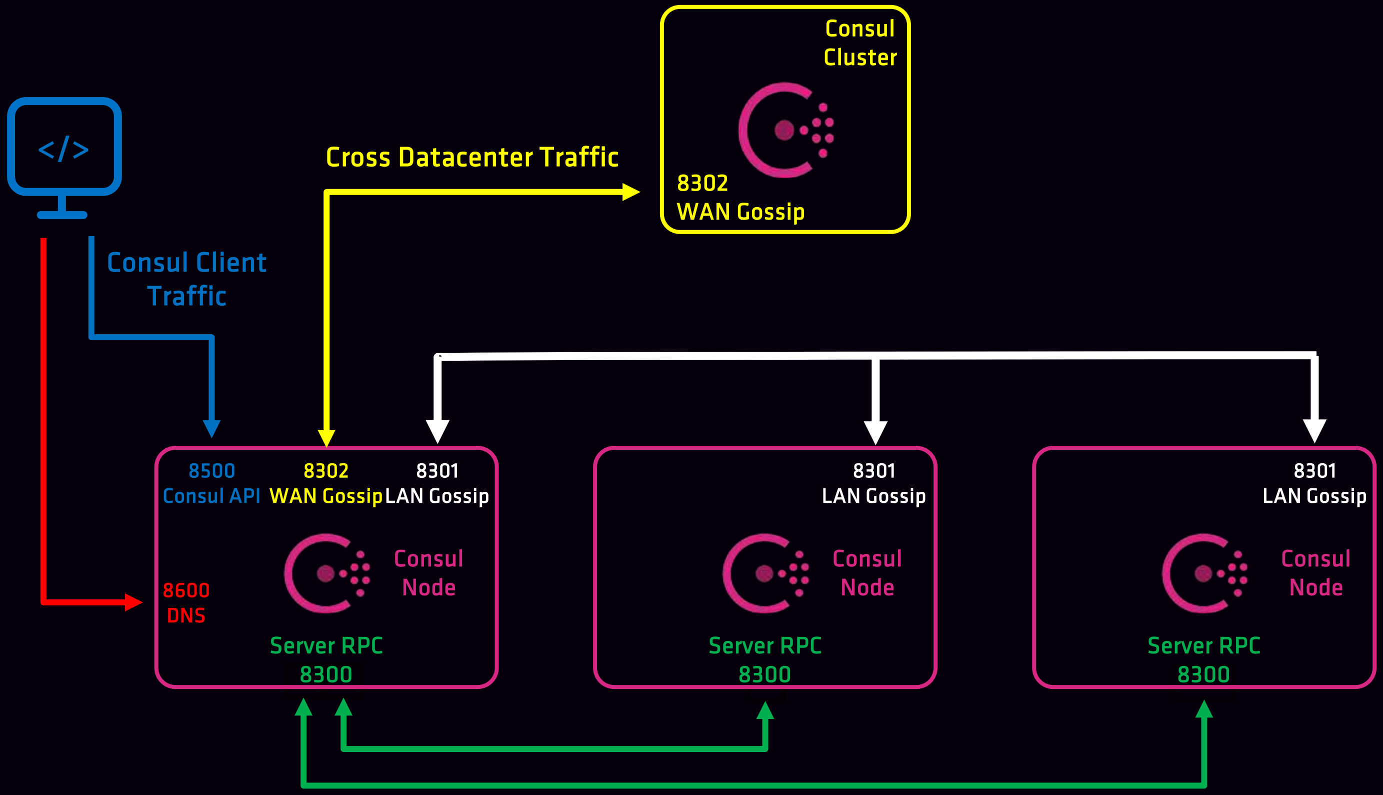


Network Traffic & Ports

- All communication happens over http and https
- Network communication protected by TLS and gossip key

- Ports (assumes default)
  - HTTP API and UI – tcp/8500
  - LAN Gossip – tcp & udp/8301
  - WAN Gossip – tcp & udp/8302
  - RPC – tcp/8300
  - DNS – tcp/8600
  - Sidecar Proxy – 21000 - 21255

Network Traffic & Ports



Accessing Consul

- Consul API can be accessed by any machine (assuming network/firewall)
- Consul CLI can be accessed and configured from any server node
- UI can be enabled in the configuration file and accessed from anywhere

Consul High Availability

- High availability is achieved using clustering
  - HashiCorp recommends 3-5 servers in a Consul cluster
  - Uses the Consensus protocol to establish a cluster leader
  - If a leader becomes unavailable, a new leader is elected

- General recommendation is to not exceed (7) server nodes
  - Consul generates a lot of traffic for replication
  - More than 7 servers may be negatively impacted by the network or negatively impact the network

Fault Tolerance

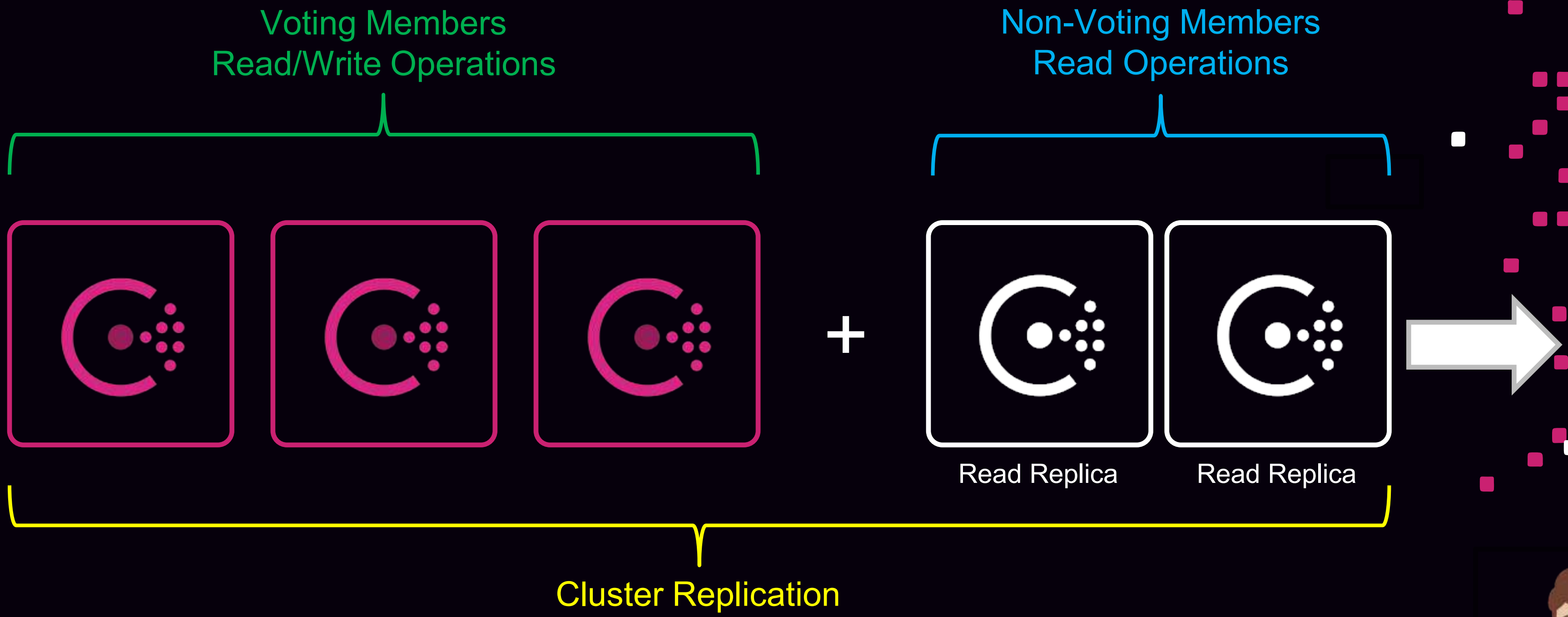
Consul Server Nodes	Quorum Size	Failure Tolerance	
1	1	0	Only! for testing
~~2~~	~~2~~	~~0~~	Don't!
3	2	1 ✓	Minimal! but acceptable
4	3	1	Meh. Maybe 3 + 1 read replica?
5	3	2 ✓	Yes! ideal for production
6	4	2 ✓	Great! use with redundancy zones
7	4	3 ✓	Wonderful! ideal for production

https://www.consul.io/docs/internals/consensus.html

Scaling for Performance

- Consul Enterprise supports Enhanced Read Scalability with Read Replicas
  - Scale your cluster to include read replicas to scale reads
  - Read replicas participate in cluster replication
  - They do NOT take part in quorum election operations (non-voting)

Scaling for Performance



Voting Members
Read/Write Operations

Non-Voting Members
Read Operations

+

Read Replica

Read Replica

Cluster Replication

Voting vs. Non-Voting Members

Enterprise Feature

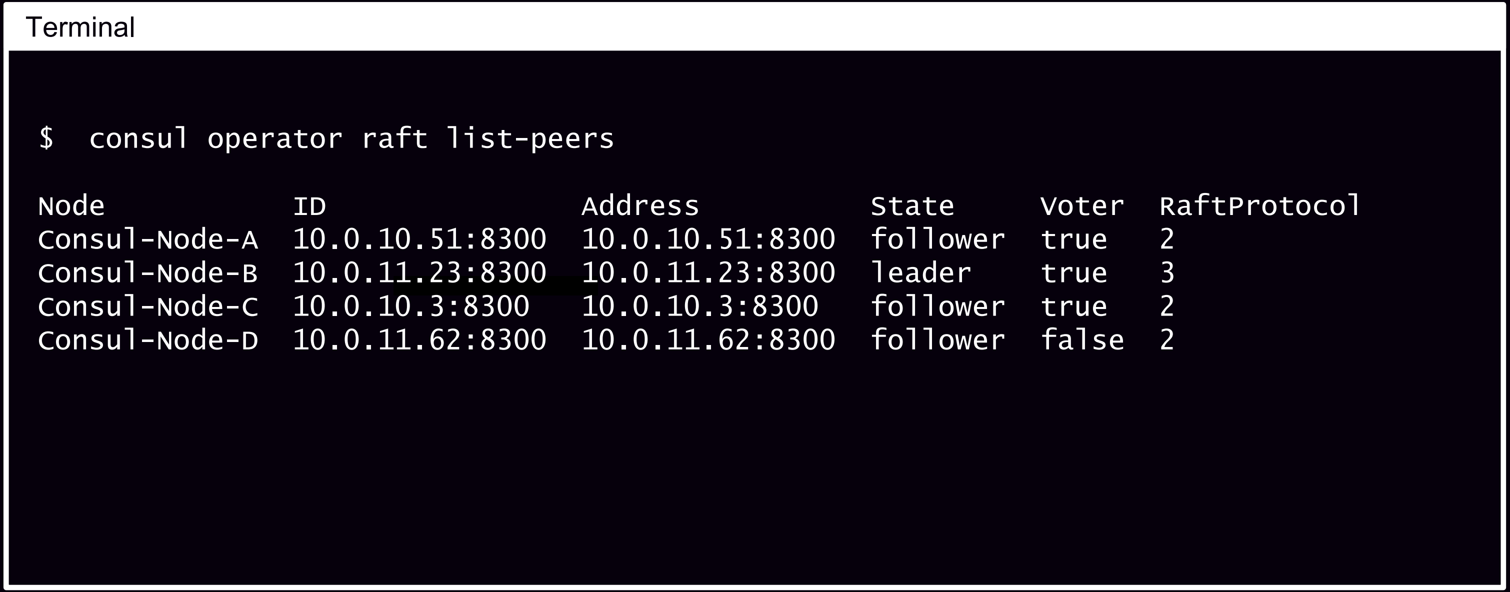
- Consul servers can be provisioned to provide read scalability
- Non-voting do not participate in the raft quorum (voting)
- Generally used in conjunction with redundancy zones

Configured using:

- non_voting_member setting in the config file
- the –non-voting-member flag using the CLI

Voting vs. Non-Voting Members

Enterprise Feature



Terminal

```
$  consul operator raft list-peers

Node           ID                Address           State     Voter  RaftProtocol
Consul-Node-A  10.0.10.51:8300   10.0.10.51:8300   follower  true   2
Consul-Node-B  10.0.11.23:8300   10.0.11.23:8300   leader    true   3
Consul-Node-C  10.0.10.3:8300    10.0.10.3:8300    follower  true   2
Consul-Node-D  10.0.11.62:8300   10.0.11.62:8300   follower  false  2
```

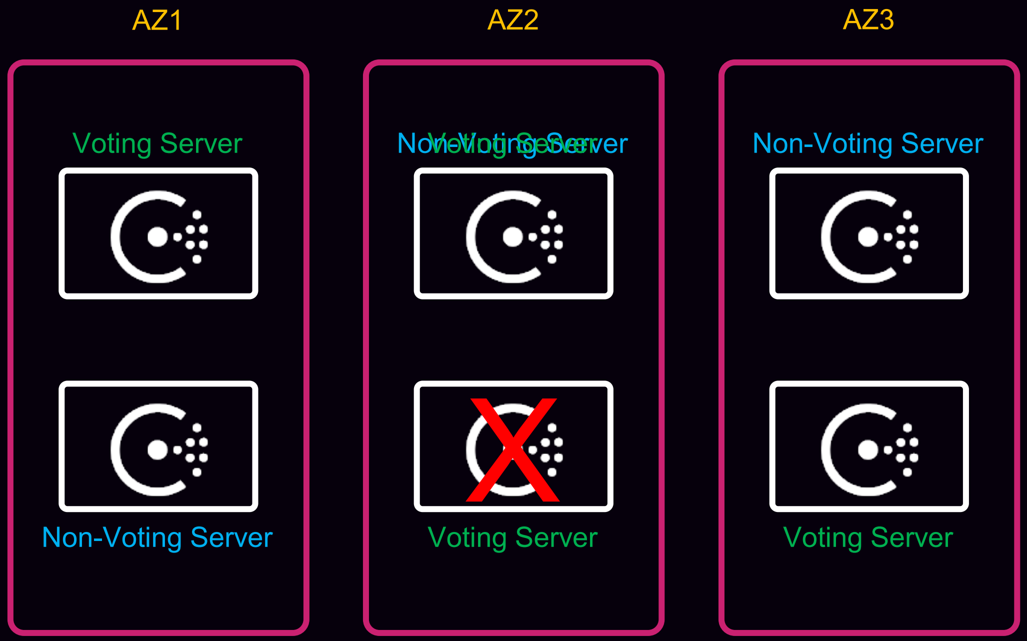
Redundancy Zones

Enterprise Feature

- Provides both scaling and resiliency benefits by using non-voting servers
- Each fault zone only has (1) voting member
    - All others are non-voting members
- If a voting member fails, a non-voting member in the same fault zone is promoted in order to maintain resiliency and maintain a quorum
- If an entire availability zone fails, a non-voting member in a surviving fault zone is promoted to maintain a quorum

Redundancy Zones

Enterprise Feature

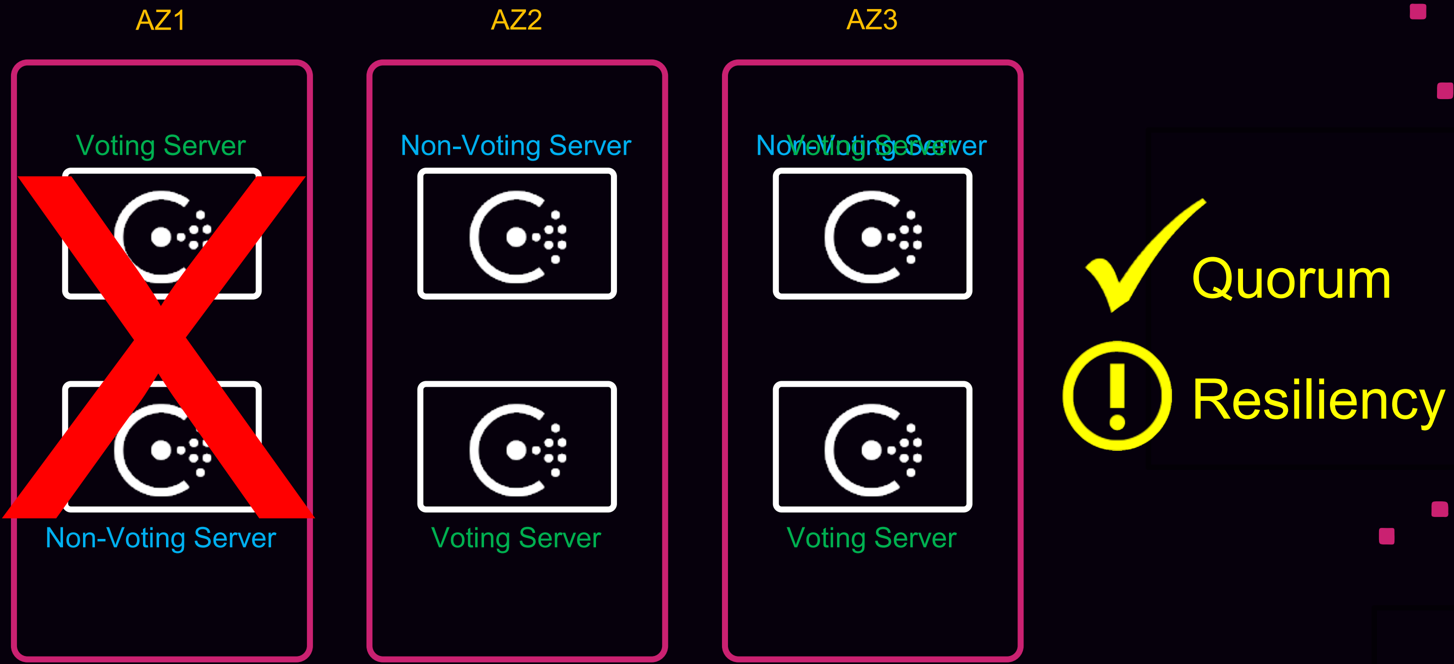


✓ Quorum

✓ Resiliency

Redundancy Zones

Enterprise Feature



AZ1

Voting Server

Non-Voting Server

AZ2

Non-Voting Server

Voting Server

AZ3

Non-Voting Server

Voting Server

Voting Server

✓ Quorum

! Resiliency

Autopilot

Enterprise Feature

Built-in solution to assist with managing Consul nodes

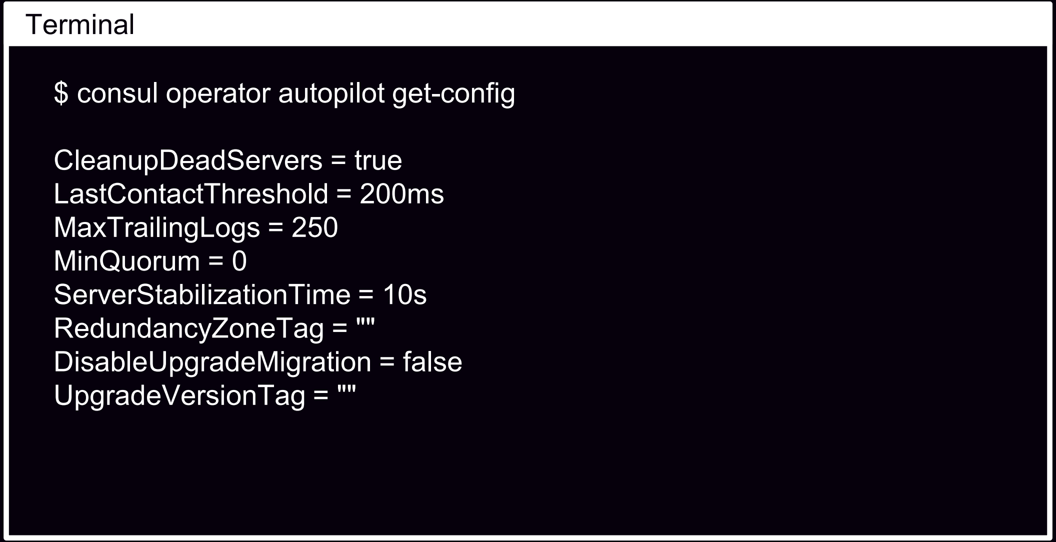
- Dead Server Cleanup
- Server Stabilization
- Redundancy Zone Tags
- Automated Upgrades

Autopilot is on by default – disable features you don't want

Autopilot

Enterprise Feature

View Configuration

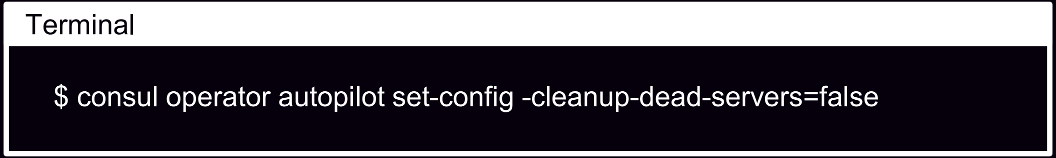


Terminal

```
$ consul operator autopilot get-config

CleanupDeadServers = true
LastContactThreshold = 200ms
MaxTrailingLogs = 250
MinQuorum = 0
ServerStabilizationTime = 10s
RedundancyZoneTag = ""
DisableUpgradeMigration = false
UpgradeVersionTag = ""
```

Change Configuration



Terminal

```
$ consul operator autopilot set-config -cleanup-dead-servers=false
```

Autopilot

Enterprise Feature

Dead Server Cleanup

- Dead server cleanup will remove failed servers from the cluster once the replacement comes online based on configurable threshold
- Cleanup will also be initialized anytime a new server joins the cluster

- Previously, it would take 72 hours to reap a failed server or it had to be done manually using consul force-leave.

Autopilot

Enterprise Feature

Server Stabilization

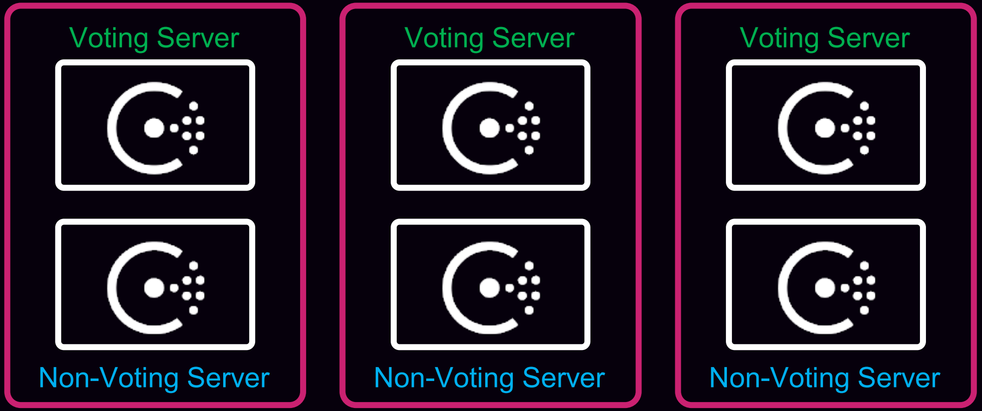
- New Consul server nodes must be healthy for x amount of time before being promoted to a full, voting member.
- Configurable time – default is 10 seconds

Autopilot

Enterprise Feature

Redundancy Zone Tags

- Ensure that Consul voting members will be spread across fault zones to always ensure high availability.
- Example: In AWS, you can create fault zones based upon Availability Zones

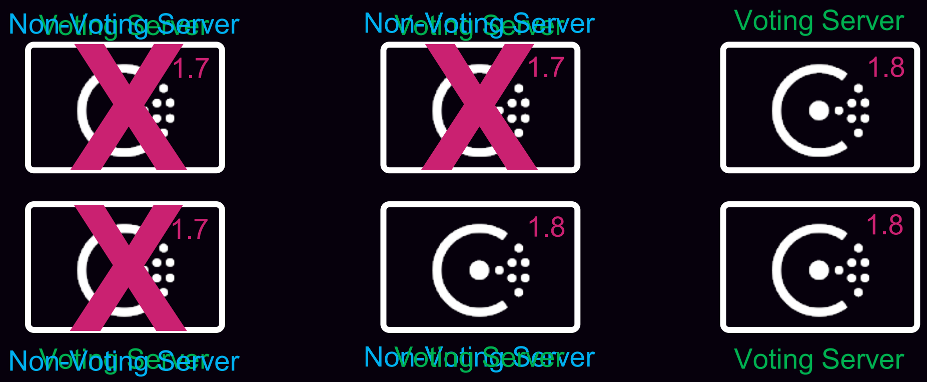


Autopilot

Enterprise Feature

Automated Upgrades Migrations

- New Consul Server version > current Consul Server version
- Consul won’t immediately promote newer servers as voting members
- Number of ‘new’ nodes must match the number of ‘old’ nodes



Explain Consul Architecture

Objective 1a: Identify the components of Consul datacenter, including agents and communication protocols

Objective 1b: Prepare Consul for high availability and performance

Objective 1c: Identify Consul's core functionality

Objective 1d: Differentiate agent roles

1 [✓ 2] 3 4 5

Difficulty Level

END OF SECTION