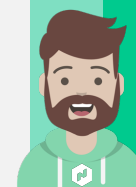
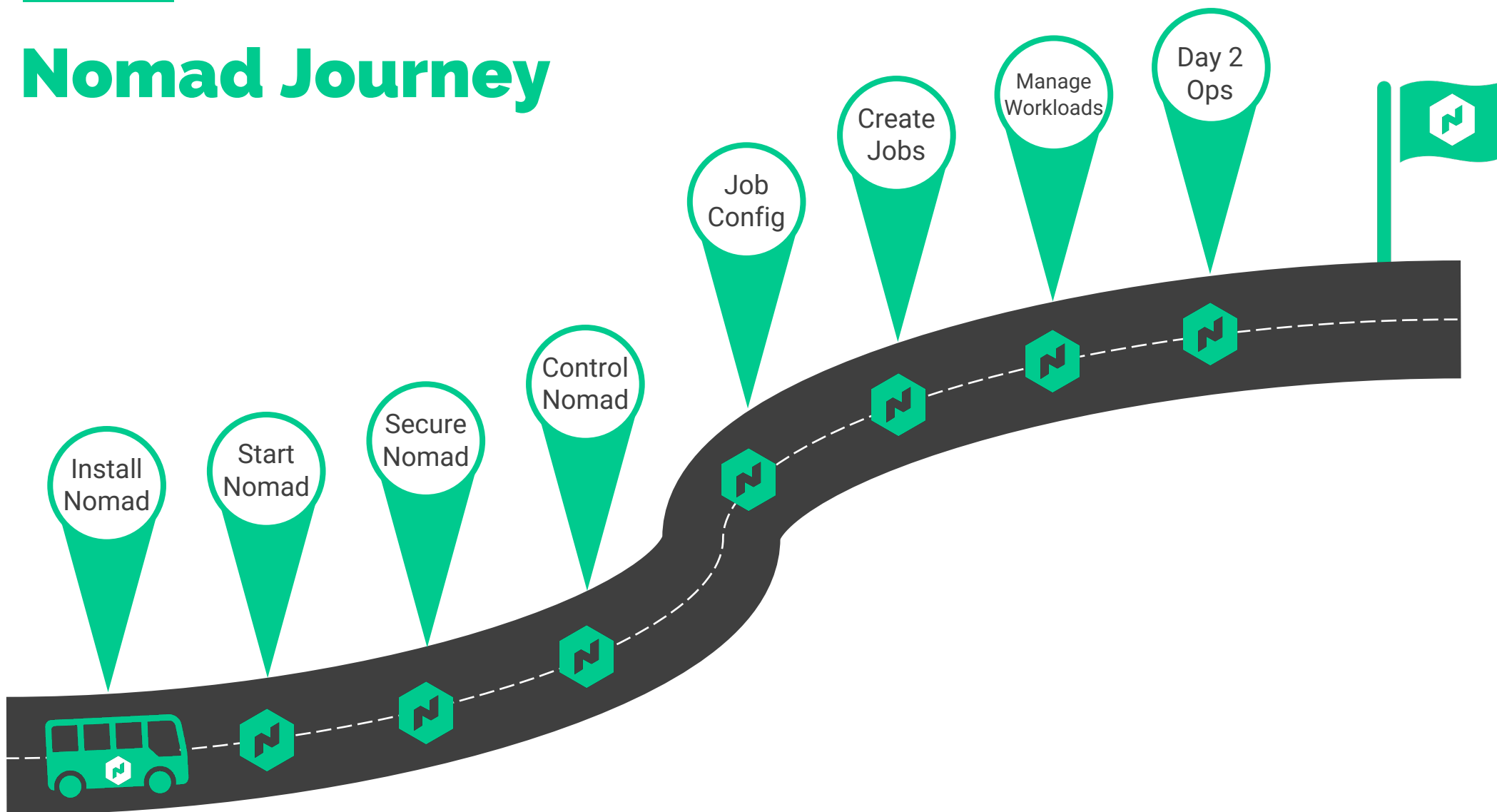




Introduction to Nomad



Nomad Journey

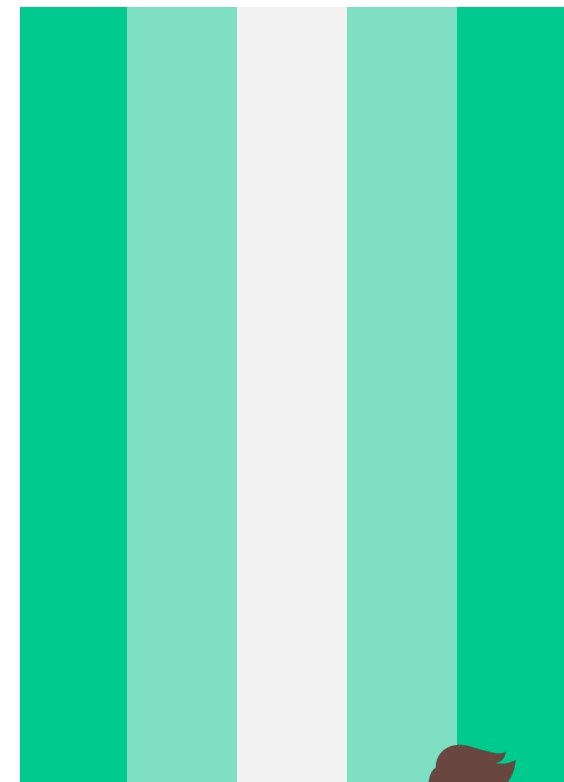


Nomad is a Container Orchestrator

(Ok, it can do more than just containers.....)



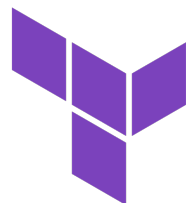
- Much simpler alternative to Kubernetes
- Nomad can deploy, manage, and scale containers for your environment
- You can use Nomad to deploy containers on-prem, at the edge, and on any cloud platform. Multi-region and multi-datacenter support.
- Nomad can scale to thousands of nodes in a single cluster
- Supports non-containerized workloads as well, such as VMs, binaries, and more...
- Tight integration into the HashiCorp ecosystem





Where Does Nomad Fit into the HashiCorp Ecosystem?

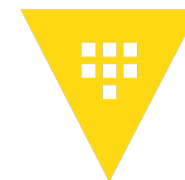
Nomad powers the application deployment and placement



HashiCorp
Terraform



HashiCorp
Consul



HashiCorp
Vault



HashiCorp
Nomad



HashiCorp
Waypoint



HashiCorp
Vagrant



HashiCorp
Boundary



HashiCorp
Packer

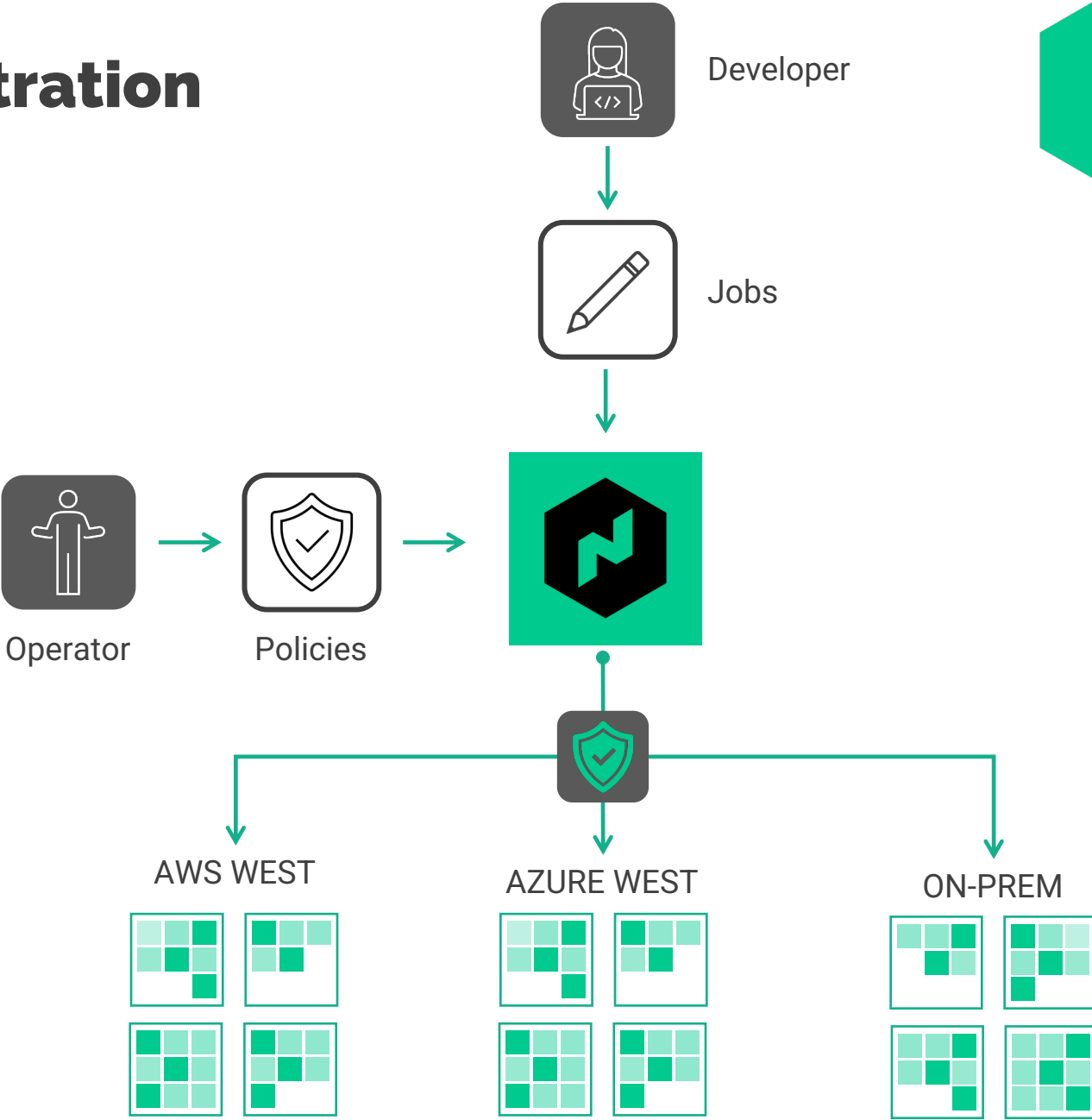


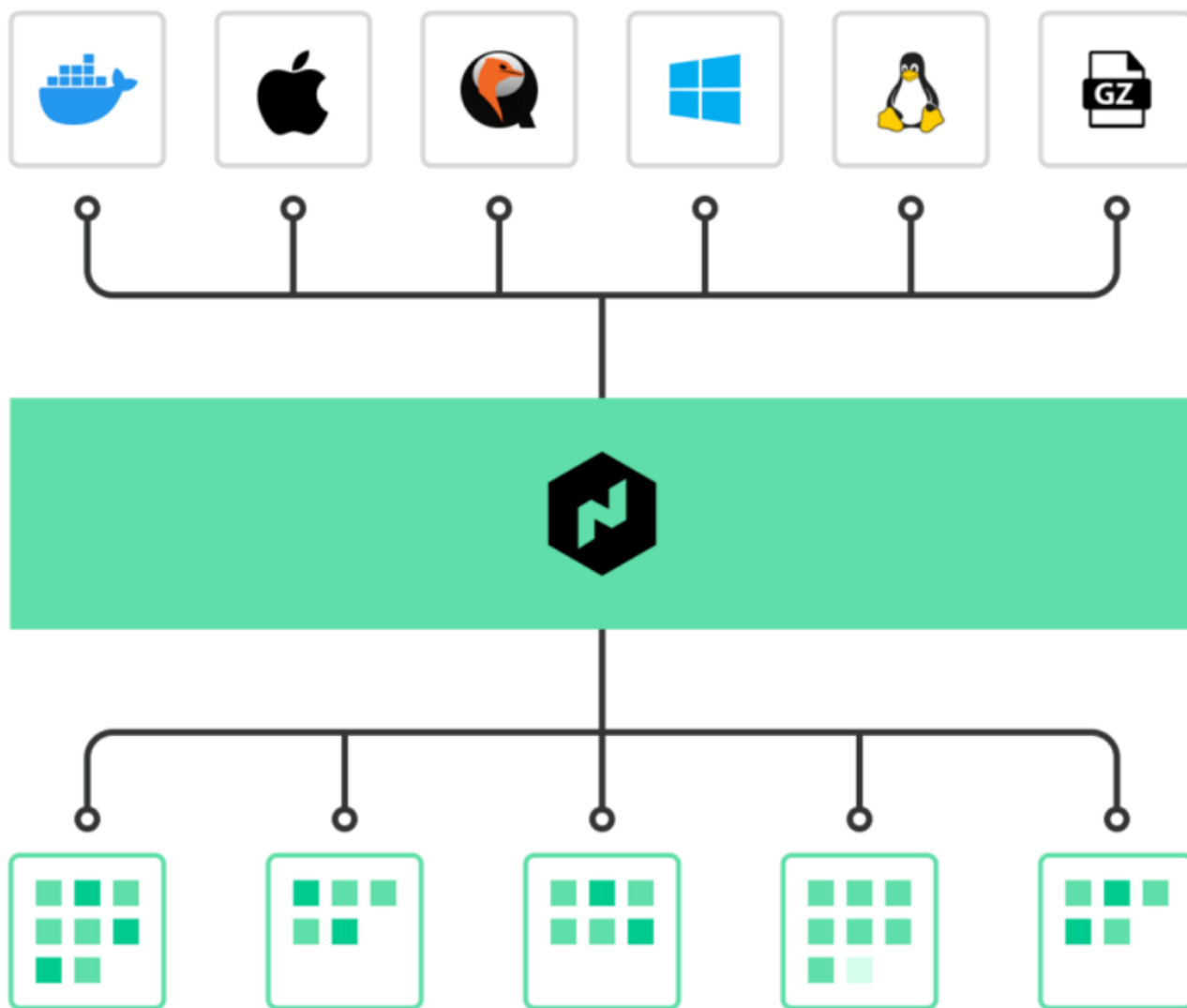
Simple Container Orchestration

The Foundation for Cloud Application Automation



COPYRIGHT – BRYAN KRAUSEN – DO NOT DISTRIBUTE





Orchestrate Any Application

- Containerized
- Non-Containerized
- Batch Applications
- Java Apps



"Nomad is an emerging
disruptor in the container
orchestrator world"

-NetApp





"Nomad does an amazing job of scheduling both containers and WebAssembly, and we think it's the future of cloud orchestrators.

I think that we can imagine a world where Kubernetes fades, and Nomad takes its place."

-Matt Butcher

Author of Go in Practice





Nomad Components



Nomad Components



Node

A physical or virtual machine in the cluster. A node is a machine running the Nomad agent.

Agent

Long-running daemon running on every member of the Nomad cluster. Agent can run in either client or server mode. This is essentially the binary that is downloaded from HashiCorp.

Server Mode

An agent running on a server that holds the global state of the cluster and participates in scheduling decisions.

Client Mode

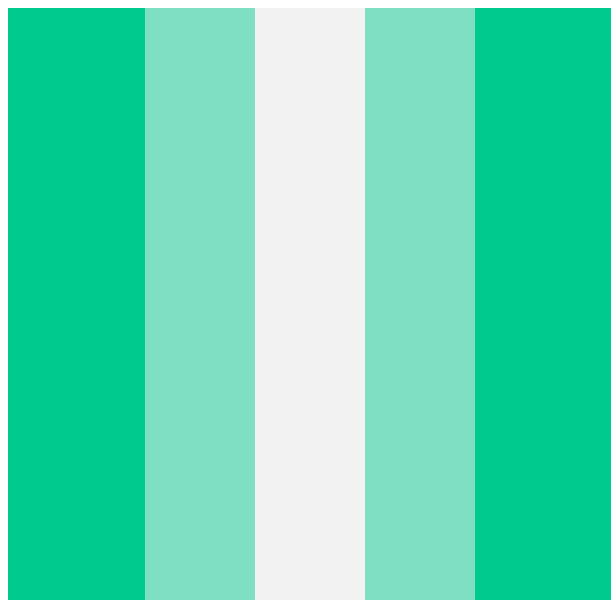
An agent that fingerprints the host to determine the capabilities, resources, and available drivers. Responsible for running workloads, such as containers.



Nomad Components



Jobs



- Definition of how a workload should be scheduled. The job specification (spec) is composed of one or more task groups, and each task defines a series of resource configurations and constraints.
- A job is submitted to Nomad and represents a desired state. For example, run 3 instances of my application using this Docker image and spread them across three nodes to ensure high-availability.



Nomad Components

Job Specification (jobspec)



- An HCL configuration file on disk which describes how a workload should be scheduled.
- It contains multiple stanzas that define configurations such as jobs, groups, tasks, services, and resources for the application



Job Specification File



```
vault-cluster.nomad

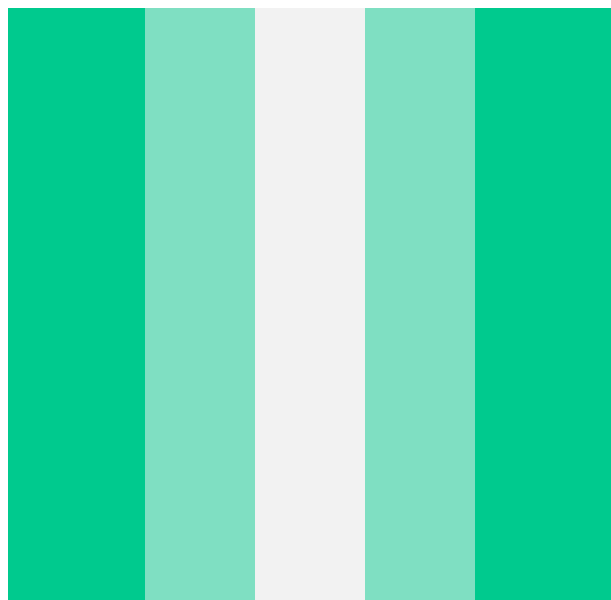
1  job "vault" {
2    datacenters = ["dc1"]
3
4    group "vault_cluster" {
5      count = 3
6      spread {
7        attribute = "{node.datacenter}"
8        target "dc1" {
9          percent = 100
10       }
11    }
12
13    network {
14      mode = "host"
15      port "vault_api" {
16        to = 8200
17      }
18    }
19
20    service {
21      name = "vault"
```



Nomad Components



Driver



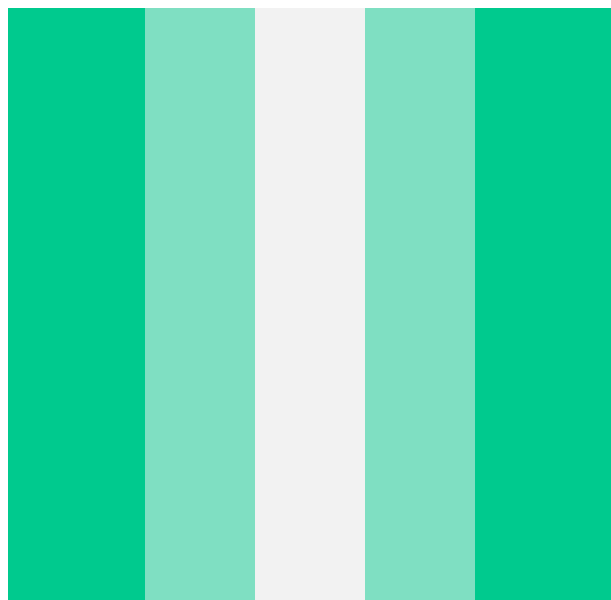
- Pluggable components that execute a task and provide resource isolation.
- Example drivers include: *docker*, *java*, *podman*, and *raw-exec*
- Drive must be installed/available before tasks can be executed. For example, if you submit a job to schedule Docker containers, Docker must be available on the Nomad clients to use...



Nomad Components



Task



- A command, service, application, or "set of work" to be executed by Nomad.
- Tasks are executed by their driver.
- Examples include:
 - Run these containers
 - Execute these commands
 - Run this Java application from a `.jar` file



Task Example



```
vault-cluster.nomad

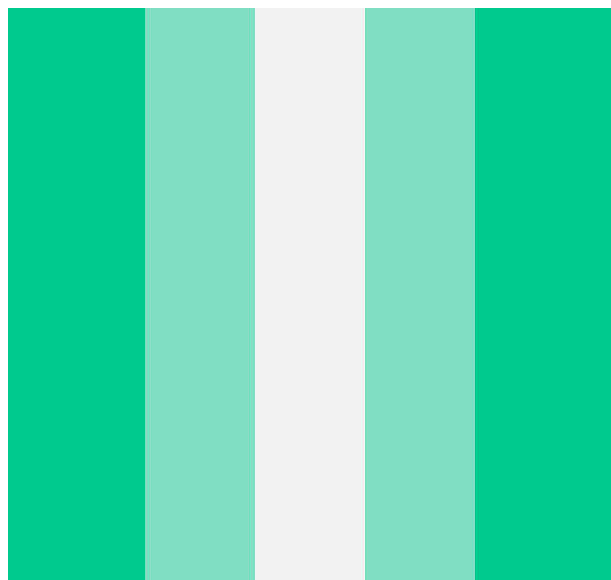
1 task "vault" {
2   driver = "docker"
3
4   config {
5     image      = "hashicorp/vault"
6     ports      = ["vault_api"]
7     auth_soft_fail = true
8   }
9
10  resources {
11    cpu    = 500
12    memory = 256
13  }
14 }
```



Nomad Components



Task Group



- A collection of individual task that should be co-located on the same node
- Any task within the defined group will be placed on the same Nomad client
- This is especially useful for applications that require low latency or have high throughput to another application in the task group.





Task Group - Example

```
1 group "hpc_application" {  
2   constraint {  
3     attribute = "${attr.cpu.arch}"  
4     value     = "amd64"  
5   }  
6  
7   task "cache" {  
8     # ...  
9   }  
10  
11   task "server" {  
12     # ...  
13   }  
14 }
```

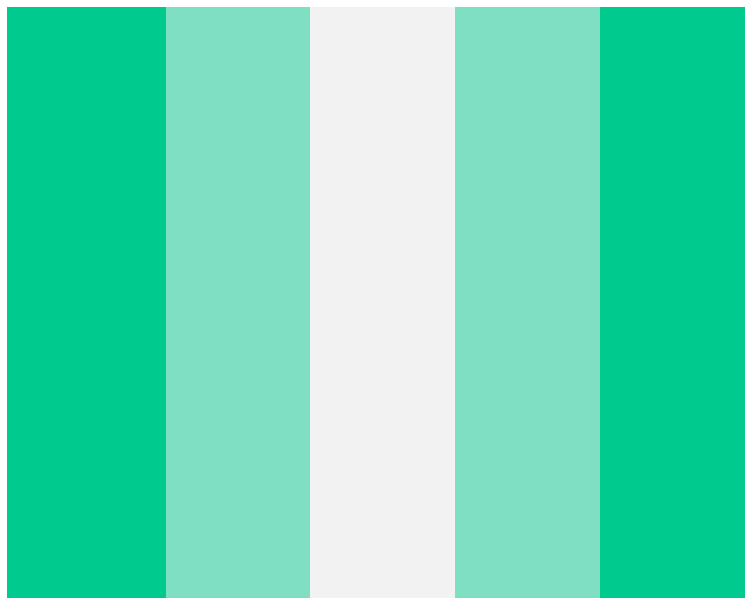
Resources defined in both *cache* and *server* will be scheduled on the same Nomad client



Nomad Components



Evaluation



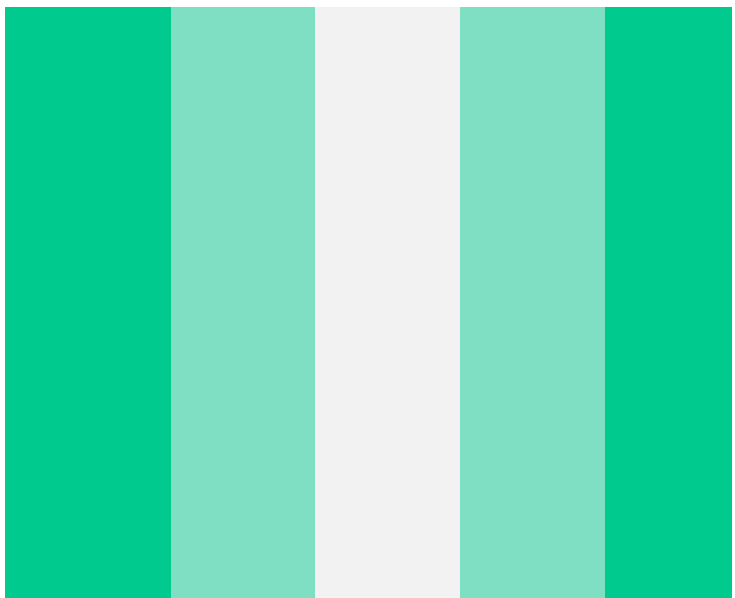
- A calculation performed by the Nomad servers to determine what action(s) need to take place to execute a job.
- Nomad performs evaluations whenever jobs are submitted or client state changes to determine if what changes need to be made to ensure the desired state



Nomad Components



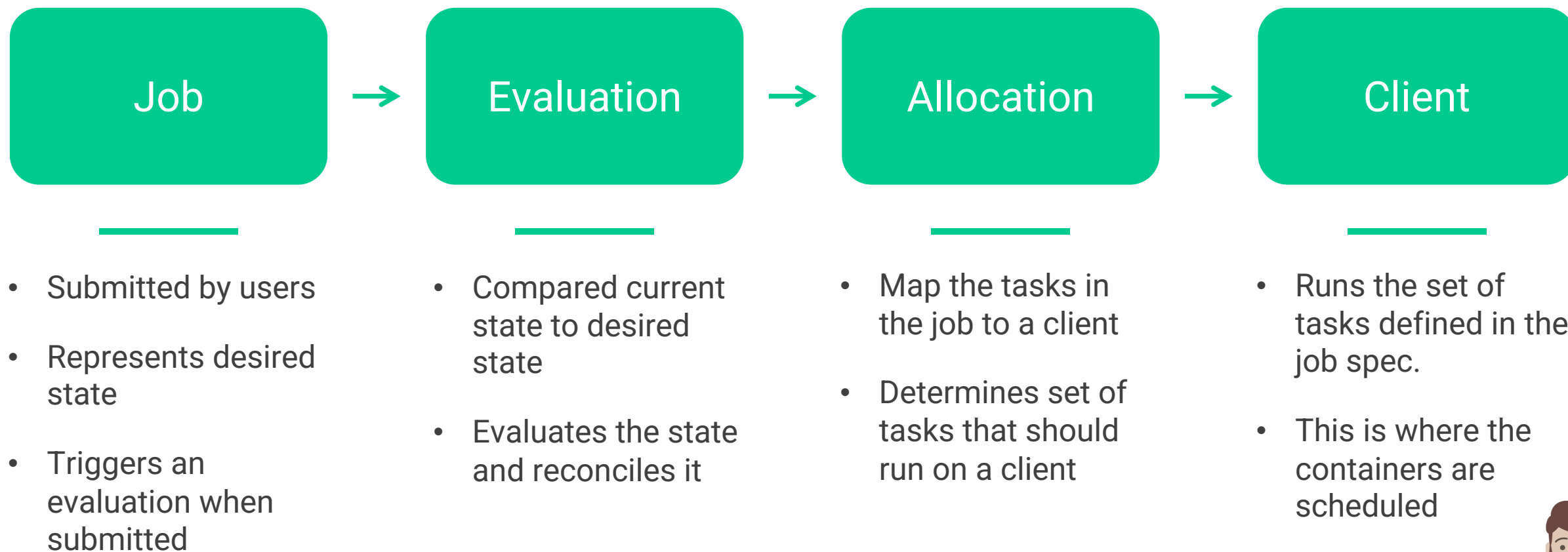
Allocation



- The mapping of tasks in a job to clients is done using allocations.
- An allocation is used to declare that a set of tasks in a job should be run on a particular node.
- Allocations can fail if there are not enough resources to execute the task, a node is down, etc.



Scheduling in Nomad





Nomad Architecture



Nomad Architecture



Datacenter

Environment that has private, low latency and high-bandwidth between nodes.



Consensus

Agreement of an elected leader in the cluster

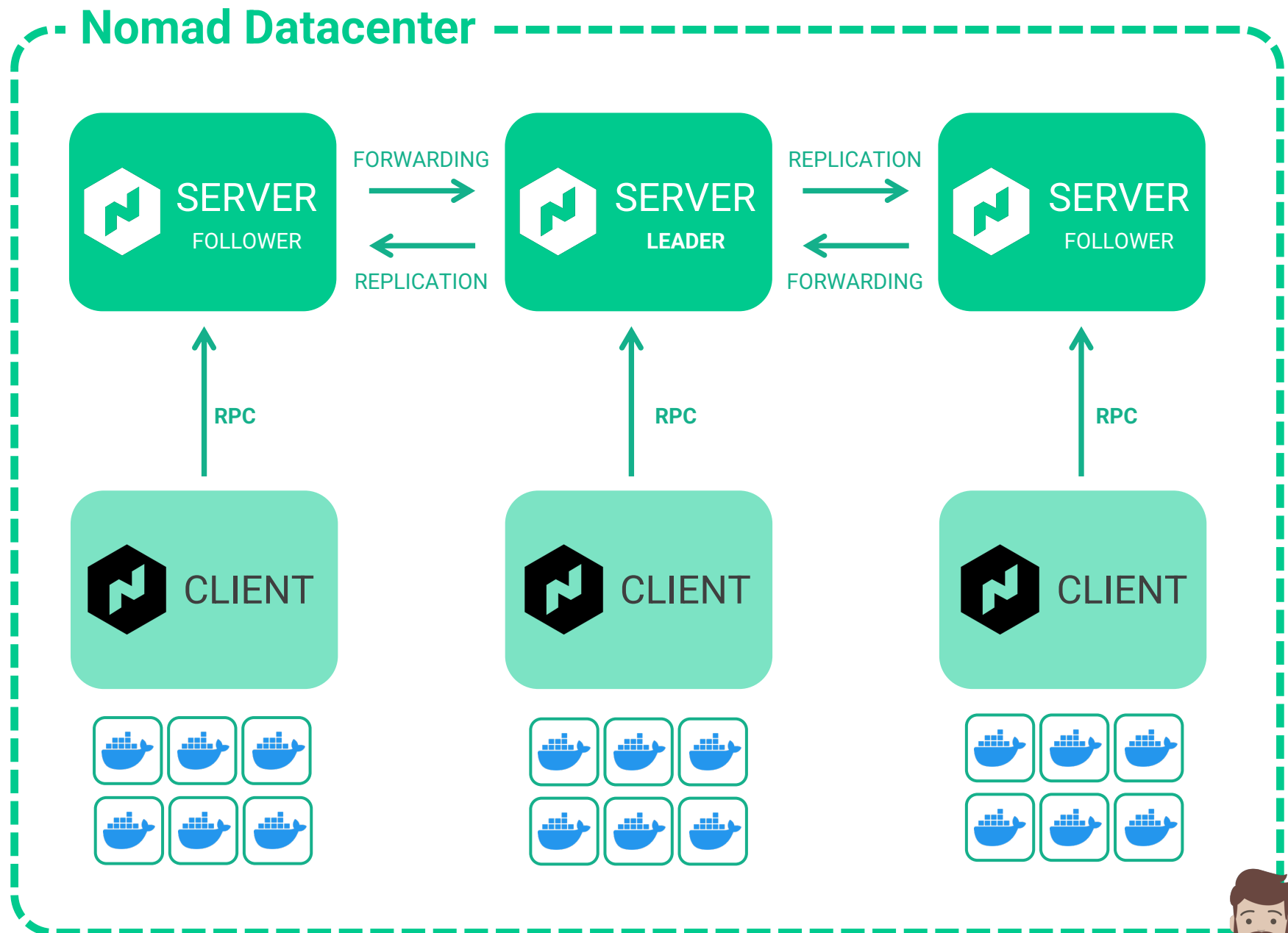


Region

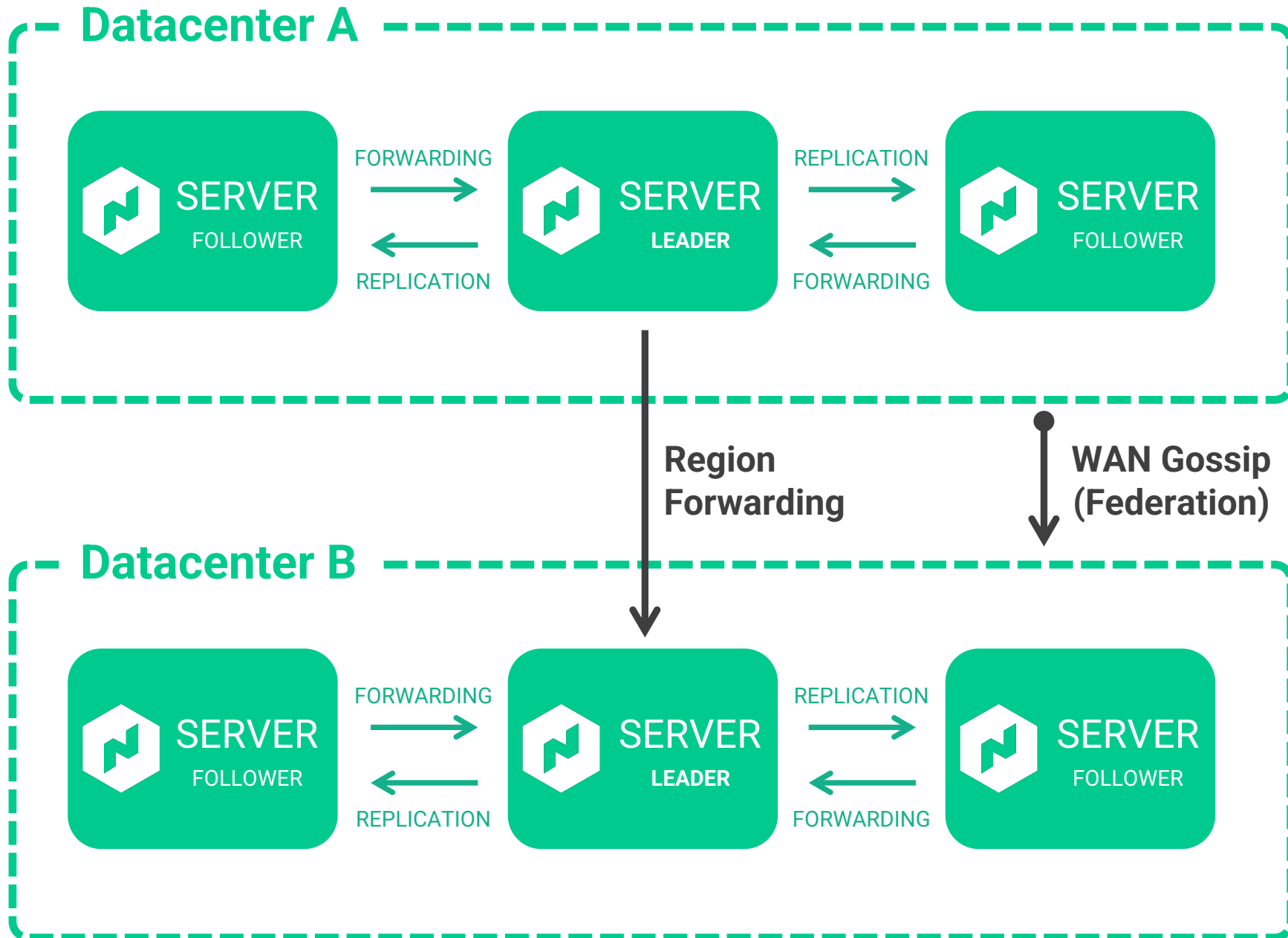
Collection of multiple datacenters, often grouped geographically.



Single-Region Architecture

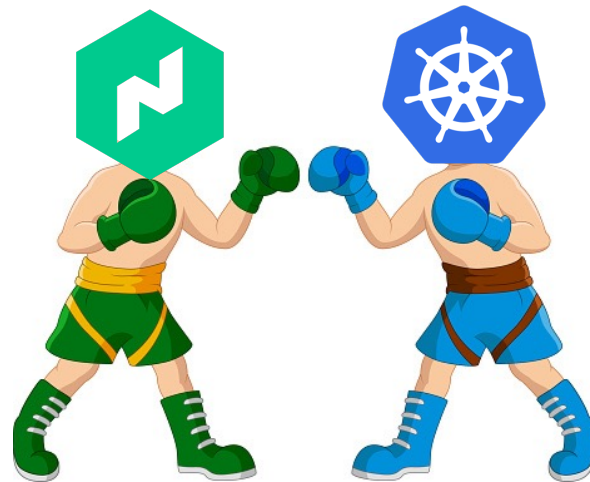


Multi-Region Architecture





Comparing Nomad to Kubernetes



How Does **Nomad** Compare to Kubernetes?

- Nomad is not based on Kubernetes, and Kubernetes is not based on Nomad, they are separate products
- Both solutions are open-source, although HashiCorp does currently offer an Enterprise version of Nomad
- Nomad supports more types of workloads than Kubernetes, such as VMs, Java JAR, Qemu, Raw Executables, Firecracker microVMs, and Wasm
- Nomad is a single binary for everything Nomad-related
- Kubernetes is much more architecturally complex than a Nomad environment
- Nomad can scale to millions of containers (check out the 1 million and 2 million container challenge HashiCorp did)
- Kubernetes is considered the industry standard, but it doesn't mean it's always the right tool





How Does **Nomad** Compare to Kubernetes?

- Nomad doesn't have the concept of Ingress Controllers for managing network connectivity
- Nomad is primarily a task-scheduling platform and can't orchestrate load balancing, config management, or routing
- Most cloud platforms provide a managed service for Kubernetes, simplifying the deployment and management of the platform. There is currently no managed service for Nomad on any cloud platform, including HCP
- Kubernetes has an extensive community for support, but it comes with the increased complexity of deployment
- Nomad is an emerging disruptor in the container orchestrator world



Comparing Terminology



COPYRIGHT – BRYAN KRAUSEN – DO NOT DISTRIBUTE

HashiCorp Nomad	Kubernetes
Job	Job
Tasks/Group	Pods
Client Agent	Kubelet
Server Node	Controllers/Control Plane
Client Node	Worker Node
Task Driver	Container Runtime
Service Discovery/Reverse Proxy	Load Balancer/Ingress
Integrated API	API Server
Raft	etcd
HCL	YAML





End of Section

