LINBIT

×

HashiCorp
Nomad

STORAGE DAY

HashiCorp Nomad Technical Introduction

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

LINBIT

Brian Hellman

COO

HashiCorp

Thomas Harris

Sr. Solutions Engineer

www.linbit.com | www.nomadproject.io

Introducing Nomad

Cloud application automation

Leading cloud infrastructure automation

Our software stack enables the **provisioning, securing, connecting,** and **running** of apps and the infrastructure to support them.

We **unlock the cloud operating model** for every business and enable their digital transformation strategies to succeed.

INFRASTRUCTURE			NETWORKING	SECURITY		APPLICATION	
Packer	Vagrant	Terraform	Consul	Boundary	Vault	Nomad	Waypoint

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

2400

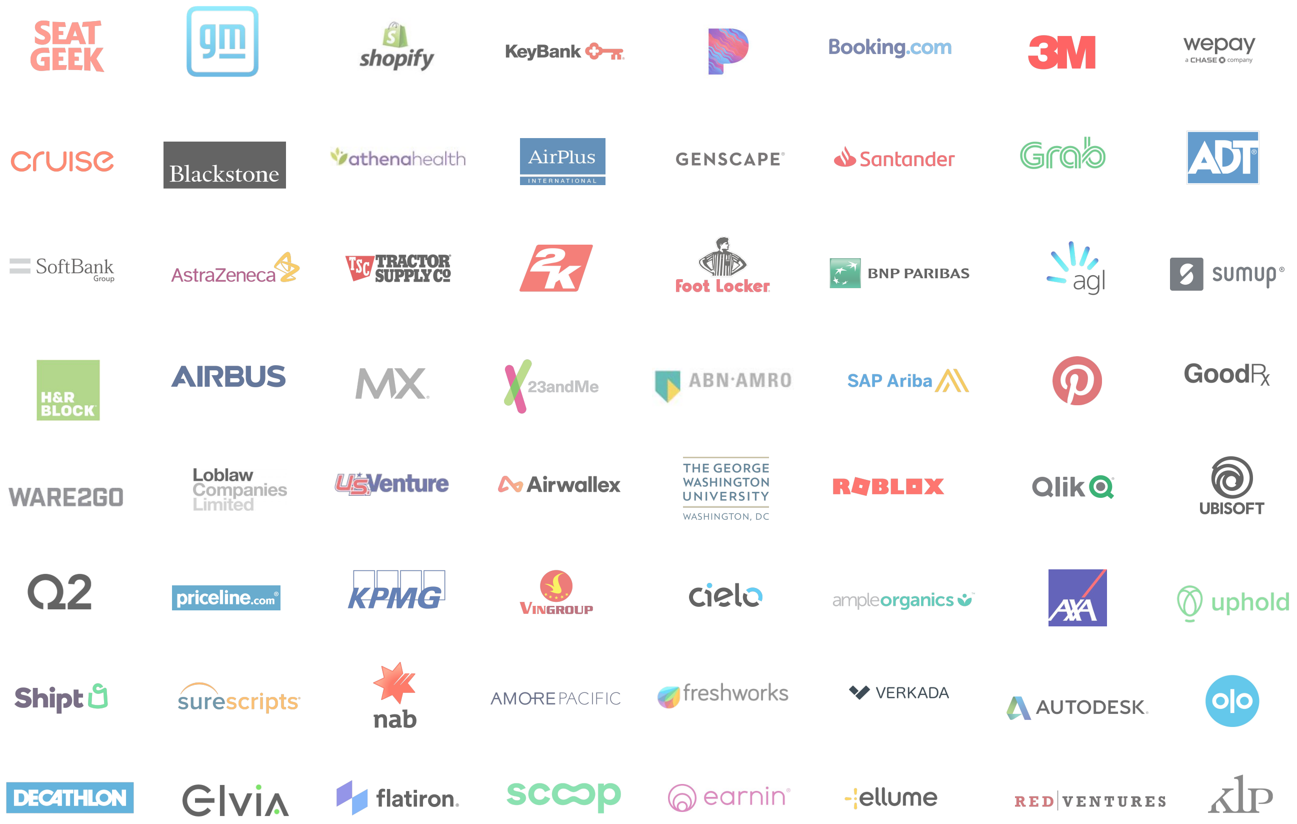
Enterprise Customers

340

of the Global 2000

158

of the Fortune 500



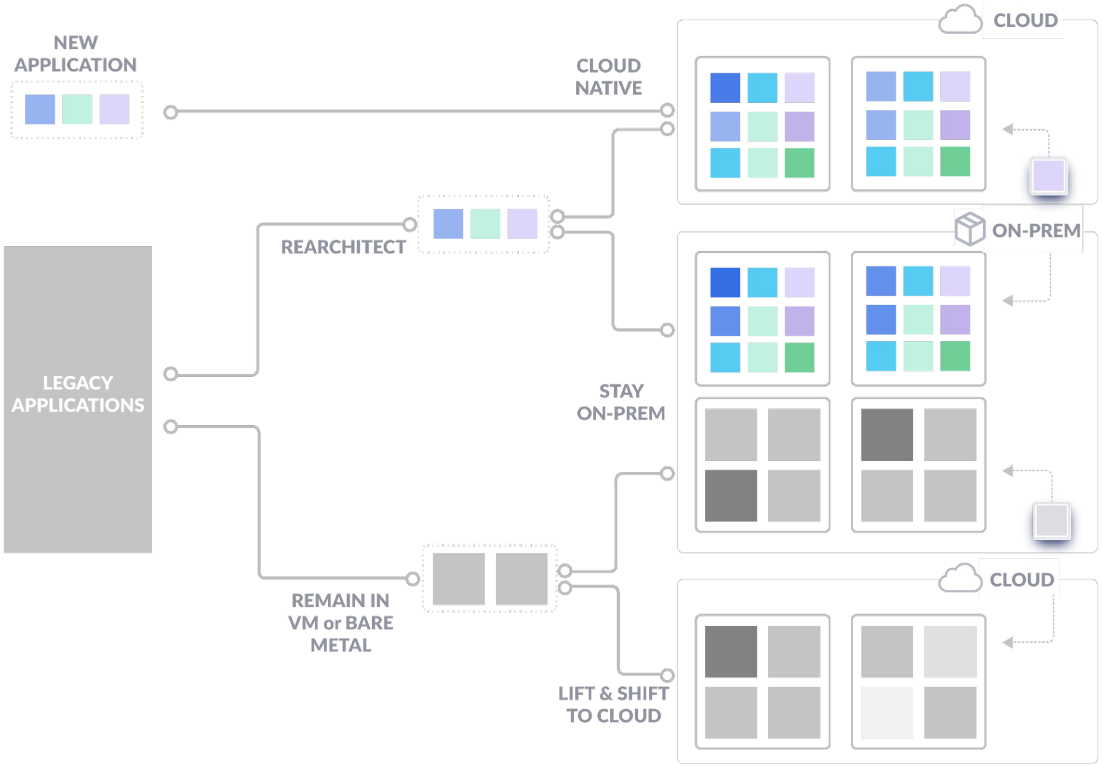
Why Orchestration?

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

The Move to Containers and Clouds

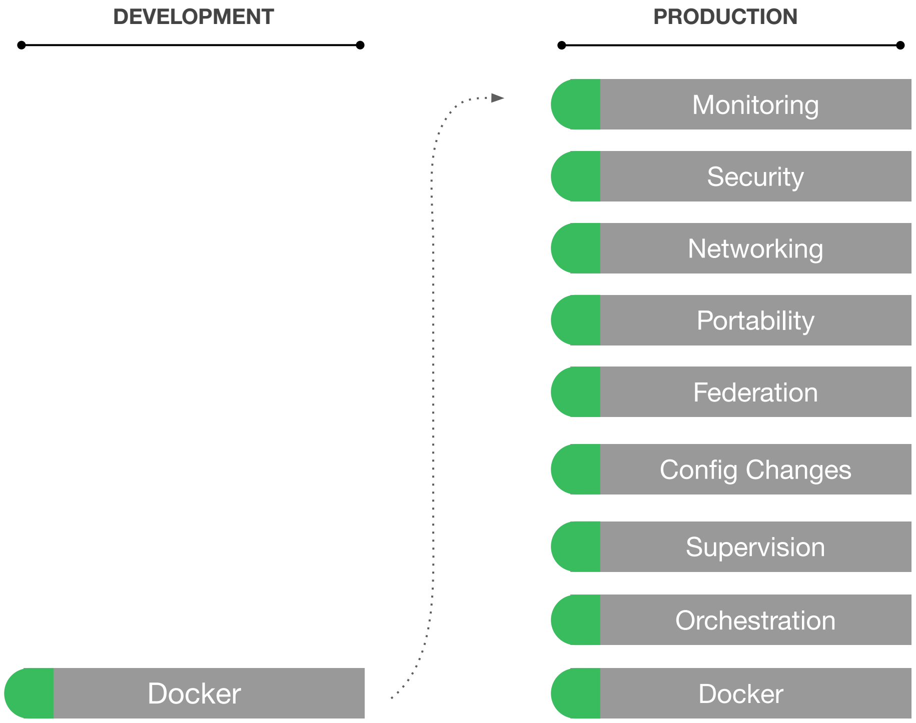
- Adopt containers and microservices for new applications
- Re-architect & migrate existing applications
- Segment of legacy applications remains the same



www.linbit.com | www.nomadproject.io

Complexity is a top challenge in using and deploying containers

40% of surveyed organizations cited “complexity” as the number one challenge in container deployment.



DEVELOPMENT

Docker

PRODUCTION

Monitoring

Security

Networking

Portability

Federation

Config Changes

Supervision

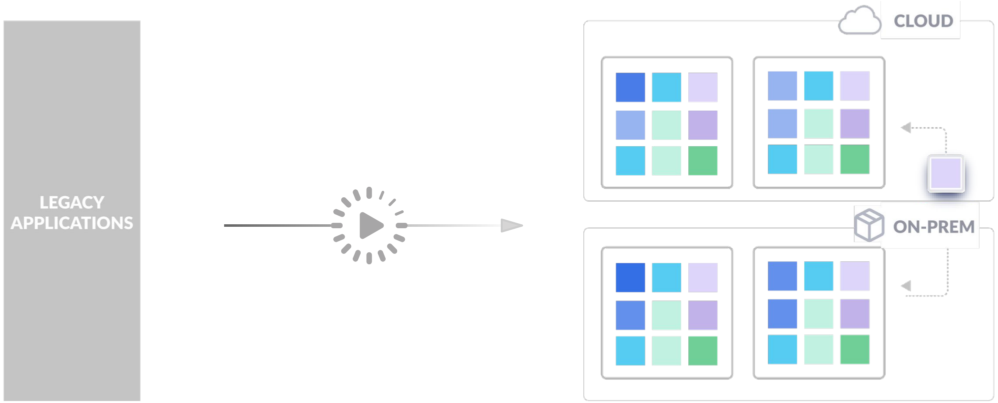
Orchestration

Docker

https://www.cncf.io/blog/2018/08/29/cncf-survey-use-of-cloud-native-technologies-in-production-has-grown-over-200-percent/

Challenges of Moving to Containers

- Lack of budget or time to refactor existing applications
- Increased complexity to support mixed systems and workflows
- Limited time to meet IT/Business requirements - Incremental vs Overhaul



Why Nomad?

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

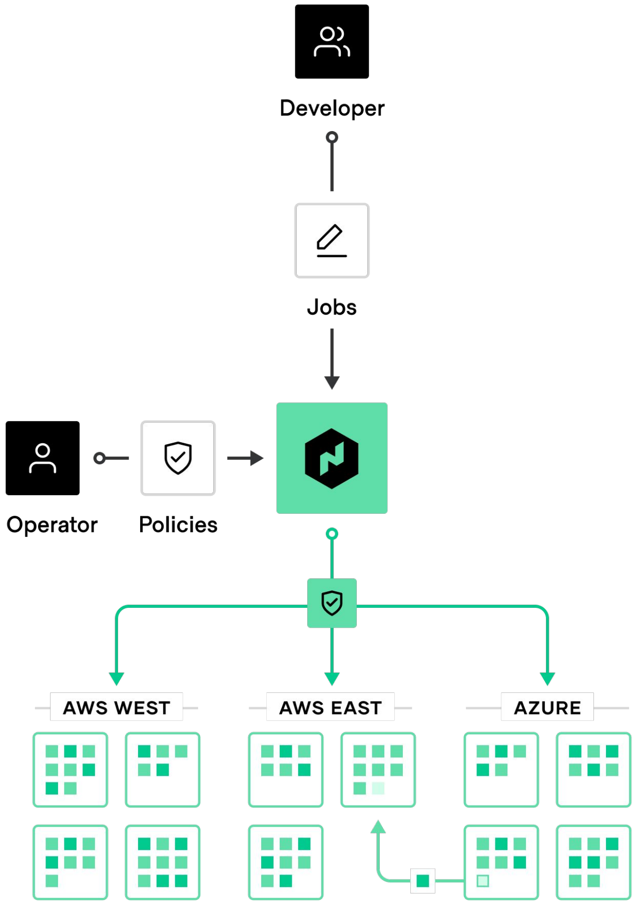
Nomad

Provides the foundation for cloud application automation by enabling simple and flexible workload orchestration in the cloud operating model.

- ✓ **Simple Container Orchestration** to easily deploy and manage containers at any scale.
- ✓ **Non-Containerized Application Orchestration** to modernize legacy applications without rewrites.

20k+
Monthly D/Ls

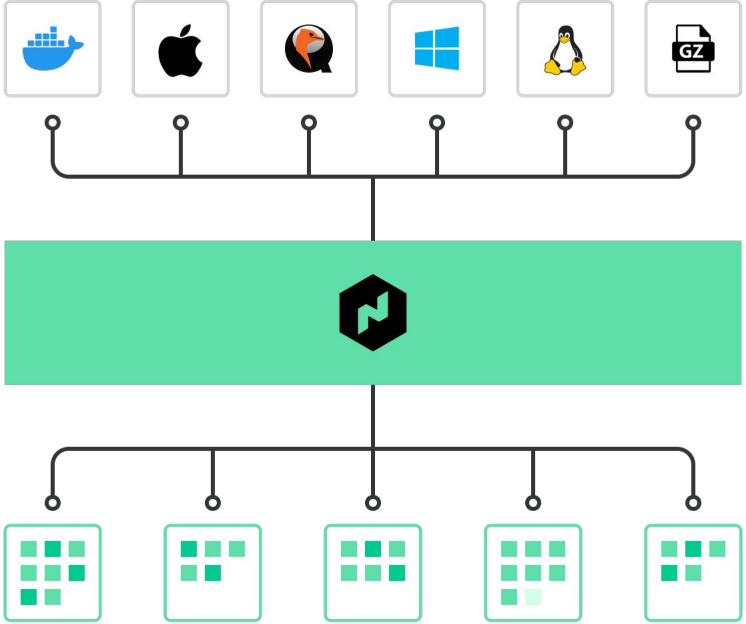
10k+
GitHub Stars



www.linbit.com | www.nomadproject.io

Guiding principle: **Orchestrate any application**

- Bringing modern orchestration benefits to all, containerized, non-containerized and batch applications.
- A simple, lightweight layer that can be integrated with any existing infrastructure.
- A single, unified workflow to accelerate incremental application modernization.

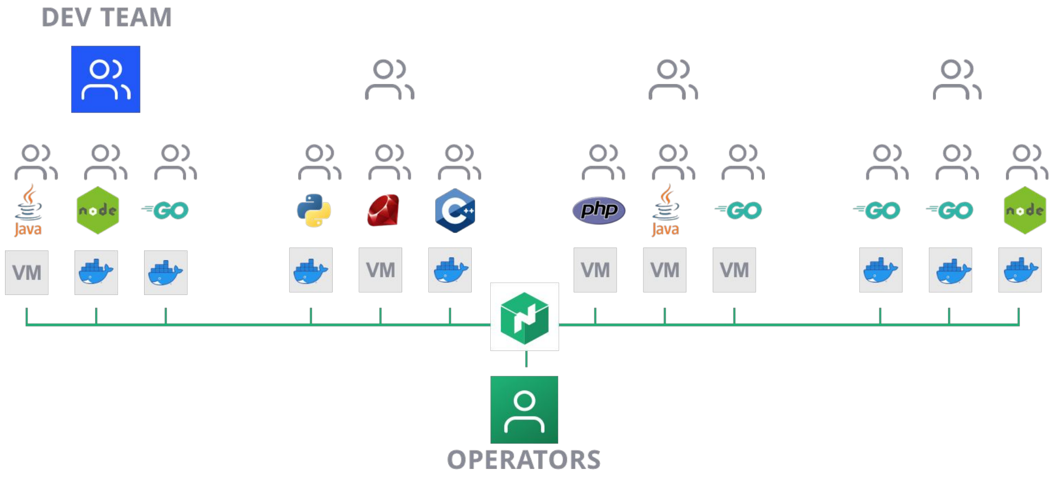


LINBIT SDS × HashiCorp Nomad

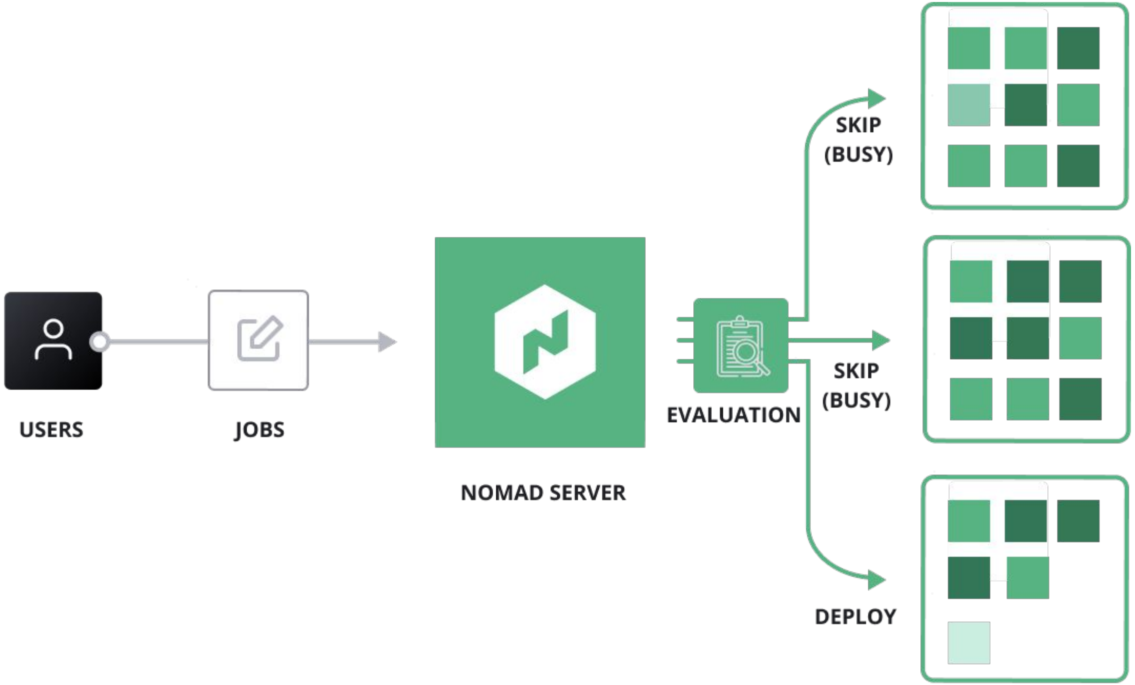
STORAGE DAY

Easy to deploy and maintain by system operators

- **Accelerate time-to-cluster** by using your normal software deployment methodology
- **Reduce cost** by utilising the skills you already have to run and maintain the clusters.
- **Don’t reinvent the wheel** by using your current infrastructure.

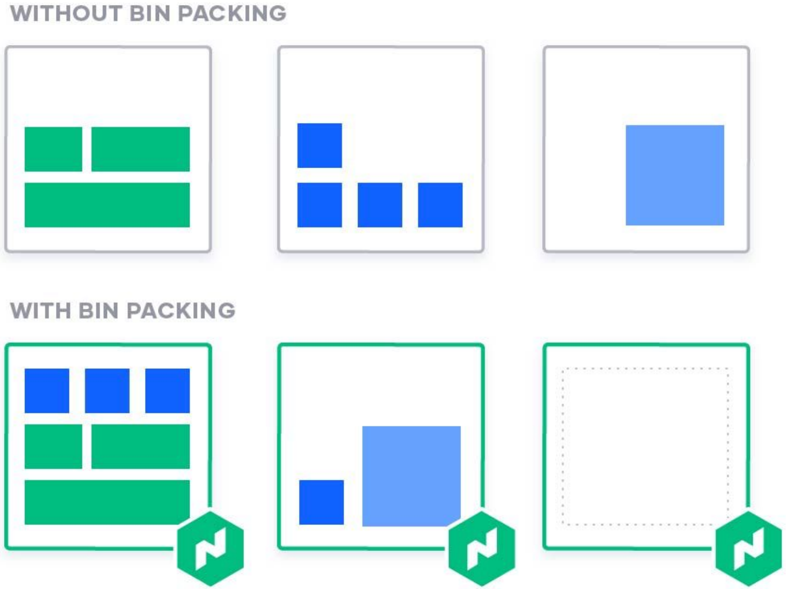


Automated Deployment Workflow



Improved resource utilization

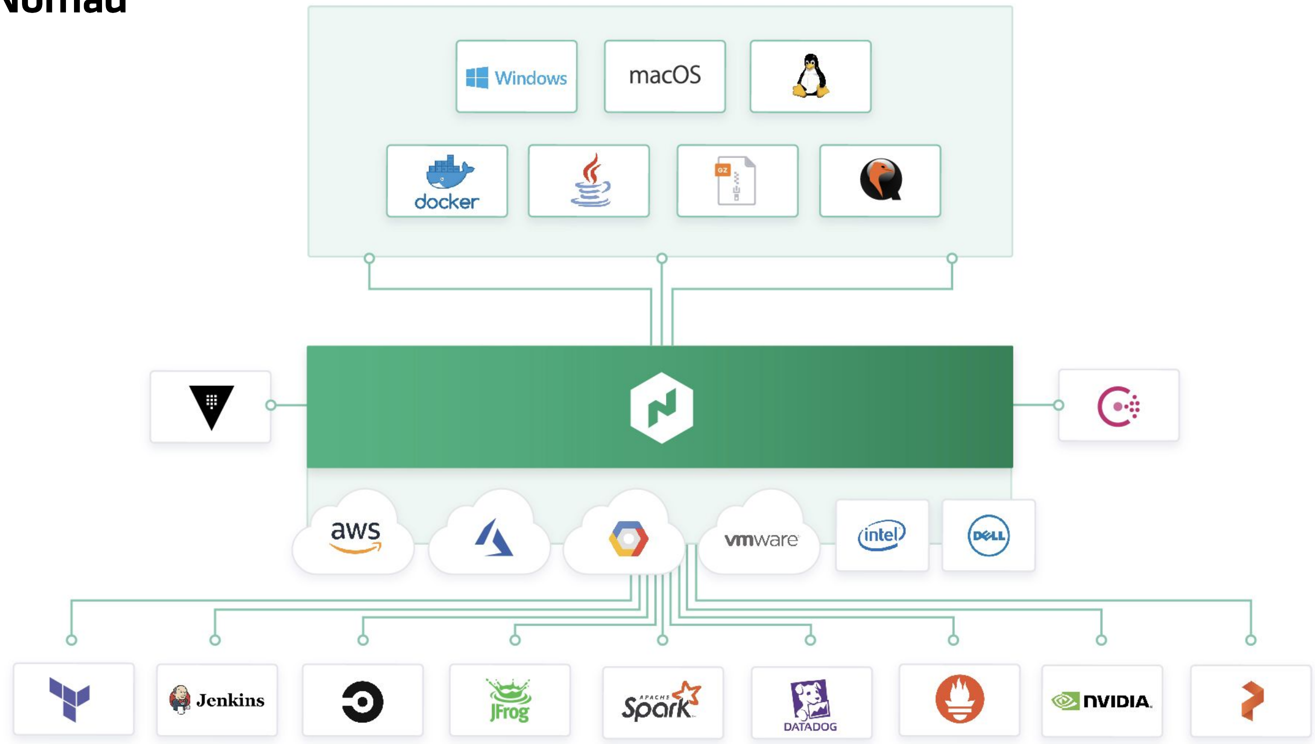
Improve resource utilization by densely scheduling applications over underutilized resources.



LINBIT SDS × HashiCorp Nomad

STORAGE DAY

Broad Ecosystem Integration



Public Users of Nomad

AUTODESK

pagerduty

pandora

ROBLOX

ebay

jet

Q2

circleci

deluxe

POLITIE

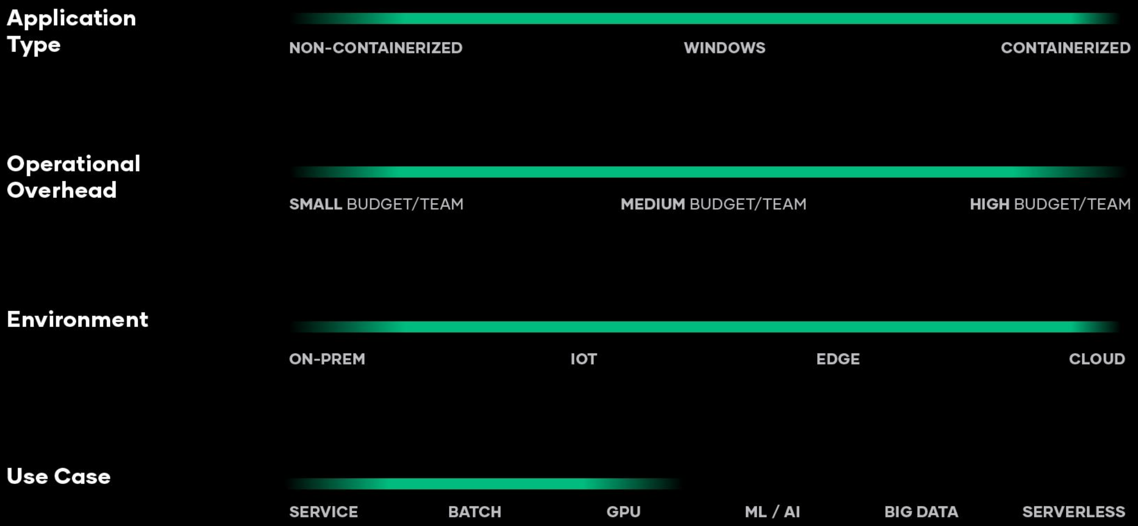
CLOUDFLARE

betterhelp

01

How organizations leverage Nomad

Multiple orchestrators for multiple use cases



Application Type

NON-CONTAINERIZED — WINDOWS — CONTAINERIZED

Operational Overhead

SMALL BUDGET/TEAM — **MEDIUM** BUDGET/TEAM — **HIGH** BUDGET/TEAM

Environment

ON-PREM — IOT — EDGE — CLOUD

Use Case

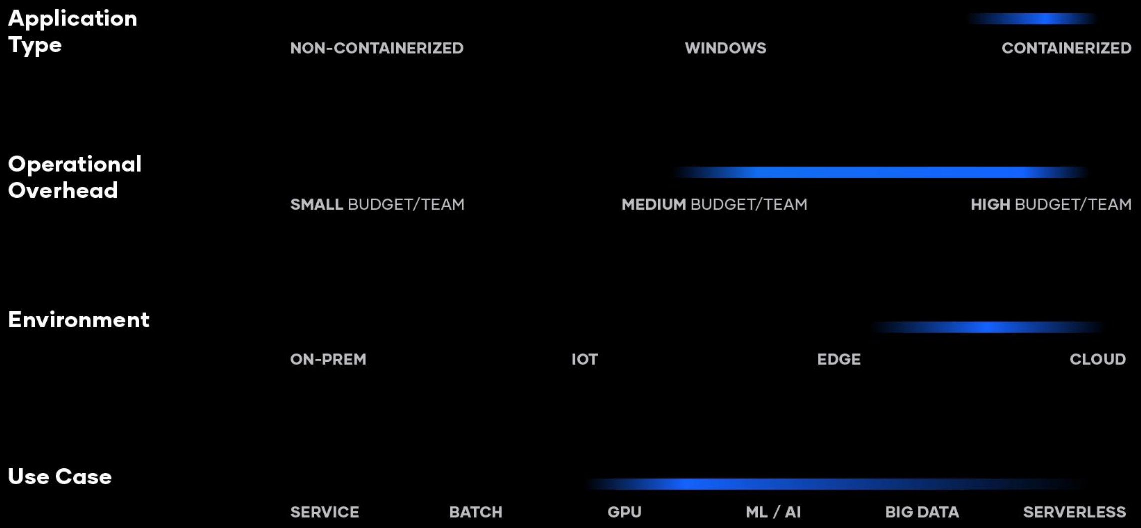
SERVICE — BATCH — GPU — ML / AI — BIG DATA — SERVERLESS

Based on interview data from HashiCorp users and customers running self-hosted Kubernetes and do not apply to managed Kubernetes services (EKS, AKS, GKE, etc.)

02

How organizations leverage Kubernetes

Multiple orchestrators for multiple use cases



Application Type

NON-CONTAINERIZED WINDOWS CONTAINERIZED

Operational Overhead

SMALL BUDGET/TEAM **MEDIUM** BUDGET/TEAM **HIGH** BUDGET/TEAM

Environment

ON-PREM IOT EDGE CLOUD

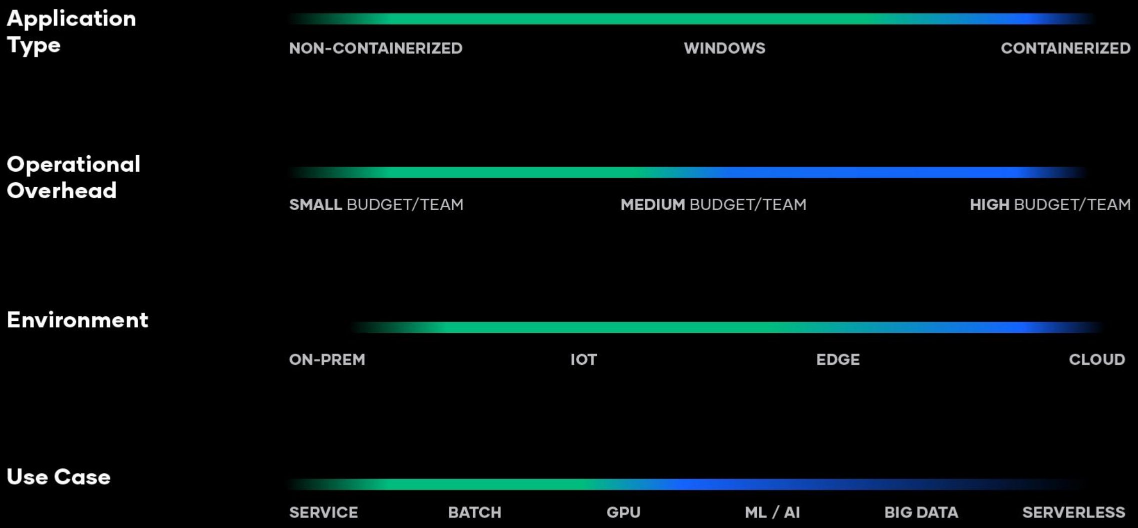
Use Case

SERVICE BATCH GPU ML / AI BIG DATA SERVERLESS

Based on interview data from HashiCorp users and customers running self-hosted Kubernetes and do not apply to managed Kubernetes services (EKS, AKS, GKE, etc.)

How organizations leverage Nomad and Kubernetes

Multiple orchestrators for multiple use cases



Application Type

NON-CONTAINERIZED — WINDOWS — CONTAINERIZED

Operational Overhead

SMALL BUDGET/TEAM — **MEDIUM** BUDGET/TEAM — **HIGH** BUDGET/TEAM

Environment

ON-PREM — IOT — EDGE — CLOUD

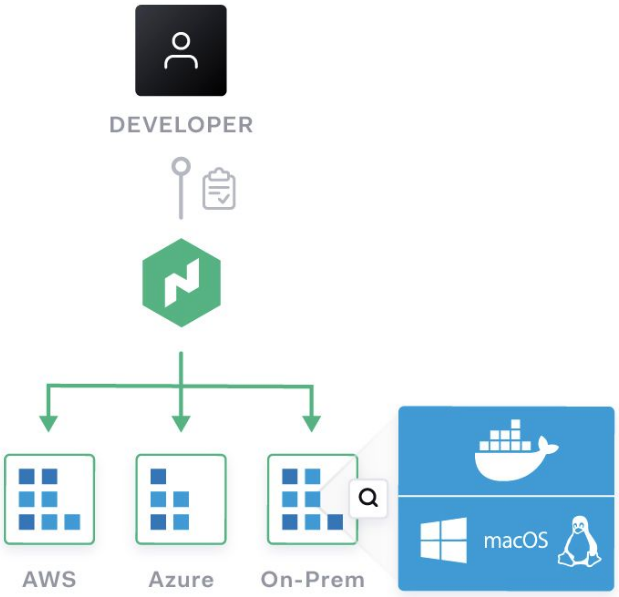
Use Case

SERVICE — BATCH — GPU — ML / AI — BIG DATA — SERVERLESS

Based on interview data from HashiCorp users and customers running self-hosted Kubernetes
and do not apply to managed Kubernetes services (EKS, AKS, GKE, etc.)

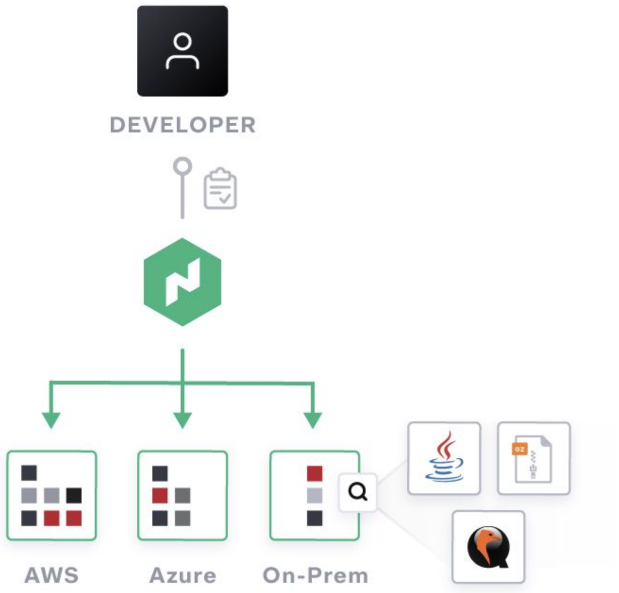
Container Orchestration

- **Accelerate time-to-value** with fast time-to-deploy and seamless integration with Consul and Vault
- **Increase adoption** by providing simplest onboarding experience to developers
- **Reduce operational overhead** with a lightweight, robust, and easy-to-manage platform



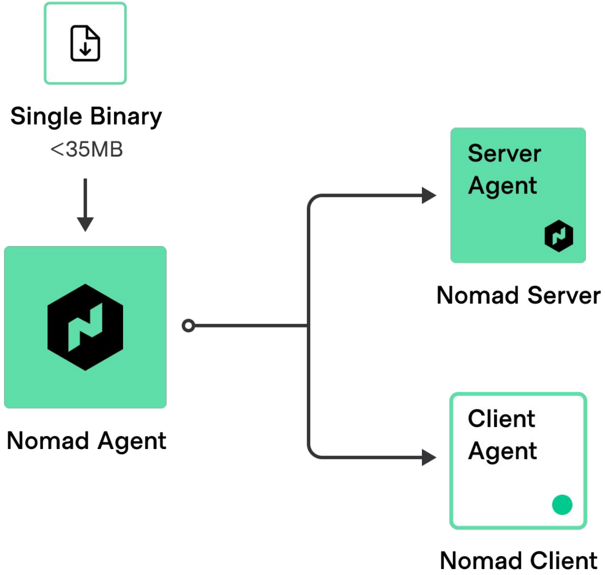
Non-Containerized Application Orchestration

- **Accelerate time-to-value** by integrating Nomad with existing infrastructure
- **Reduce cost** by densely scheduling applications to underutilized resources.
- **Zero downtime deployment** by bringing the benefits of modern deployment strategies to legacy applications



Technical Overview

Lightweight single binary



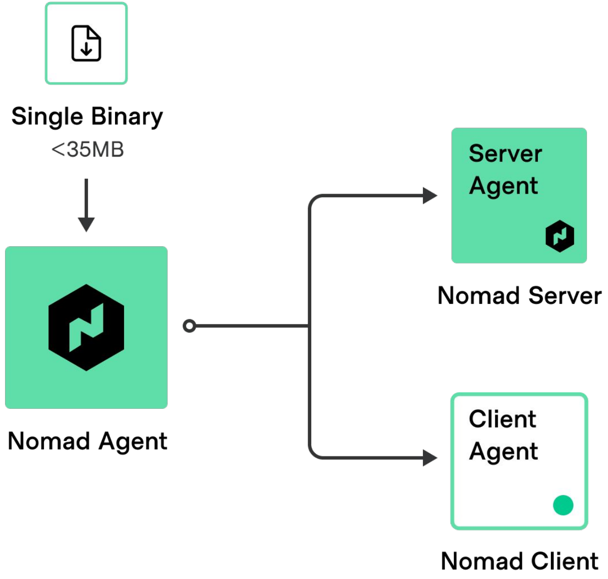
Nomad Server forms the control plane for scheduling.

```
$ .nomad -config=server.hcl
```

Nomad Client runs on the node which registers with the servers, watching for any work to be assigned and execute tasks

```
$ .nomad -config=client.hcl
```

Nomad Architecture - Laptop



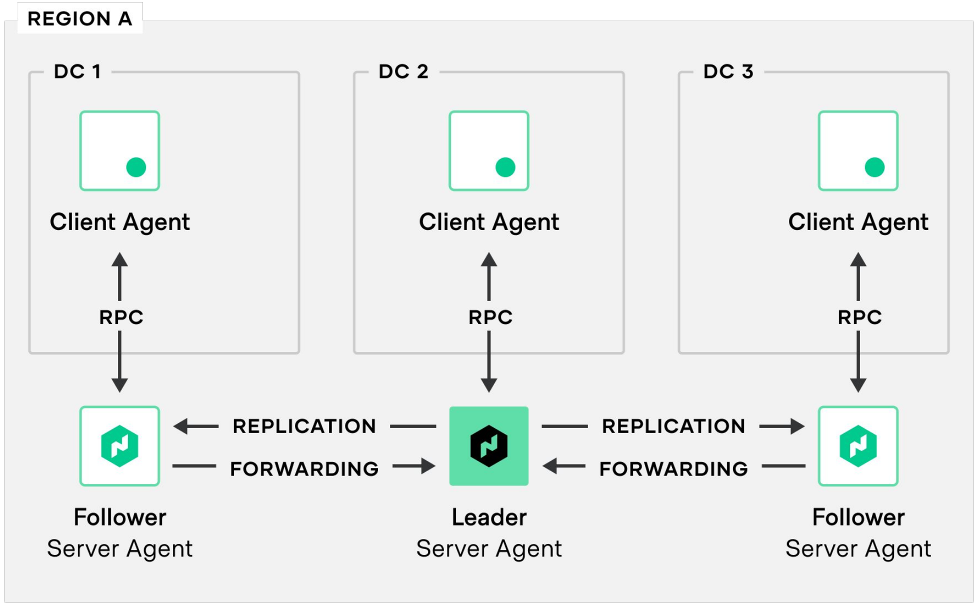
Nomad Agent runs as a Server and a Client on your local laptop.

```
$ nomad agent -dev
```

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

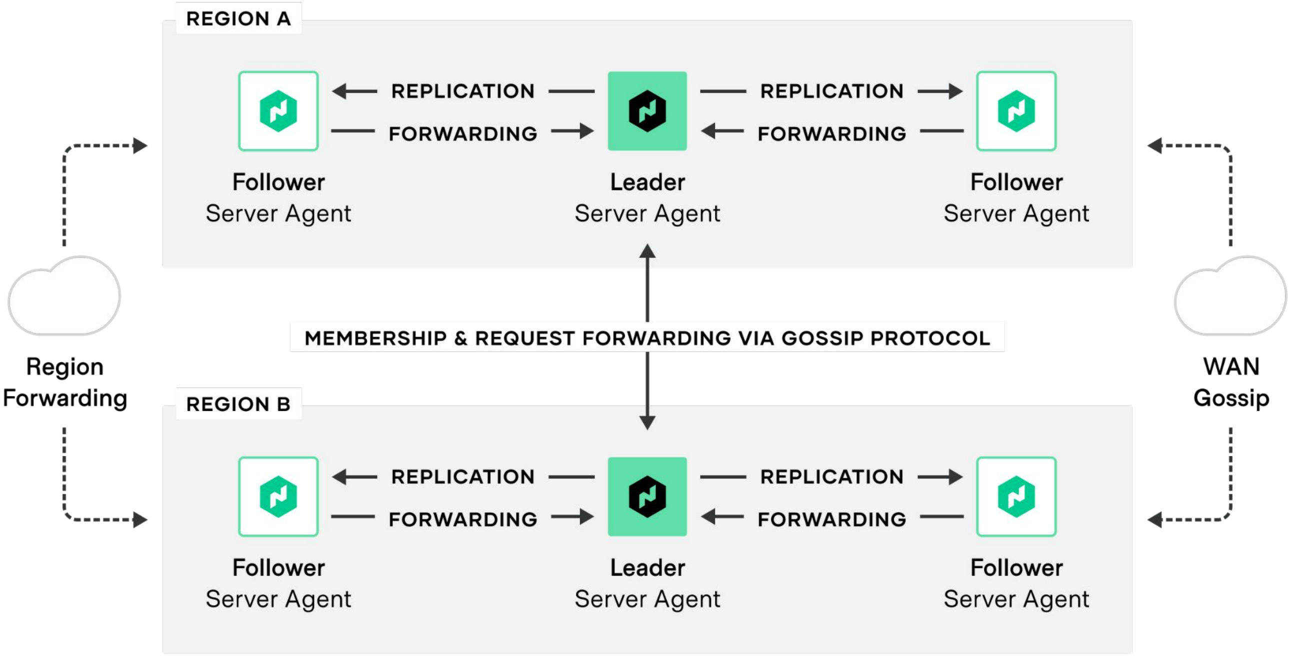
Nomad architecture - single region



LINBIT SDS × HashiCorp Nomad

STORAGE DAY

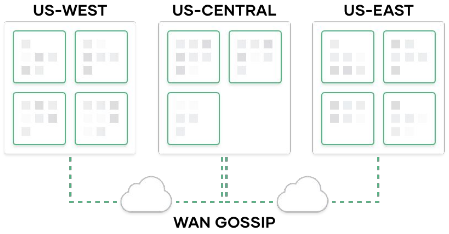
Nomad architecture - multi-region

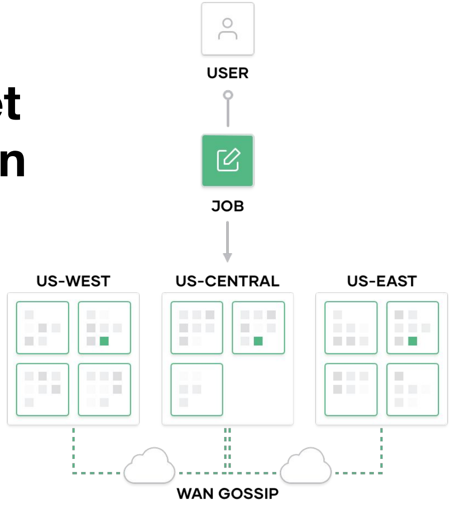


LINBIT SDS × HashiCorp Nomad
STORAGE DAY

Nomad federation

The first and only orchestrator on the market with complete and fully-supported federation capabilities for production.





Multi-Cluster Visibility

- Single command to join Nomad clusters together.
- Single interface to monitor job status across any cluster.

Multi-Cluster Deployment (ENT)

- Single job spec to deploy applications across multiple clusters.
- Orchestrate configurable rollout and rollback across multiple clusters.
- New in 0.12

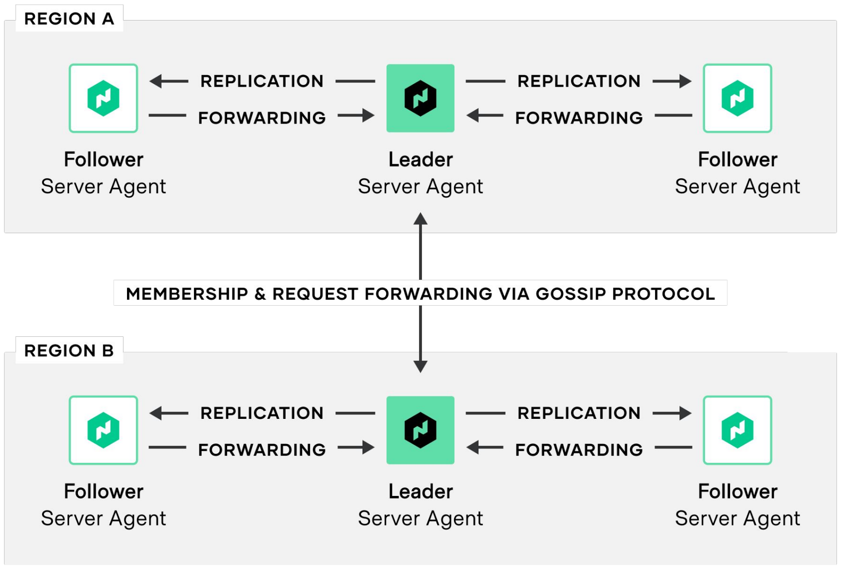
Native federation capabilities

- Federating is as simple as joining the servers.
- A single command to join a server in a remote region.

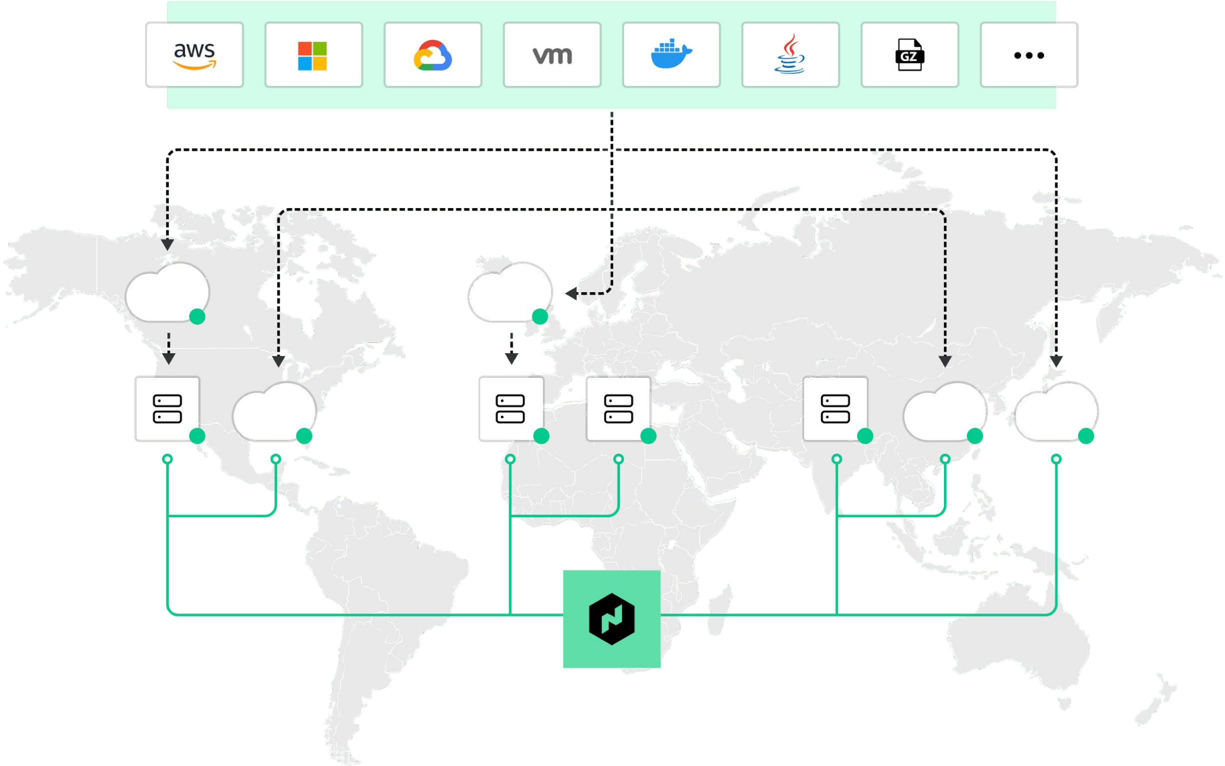
```
$ nomad server join
X.X.X.X:XXXX
```

- Submit jobs to any region or query the state through any server in any region transparently.

Nomad Multi-Region Architecture



Single monitoring and control plane with federation



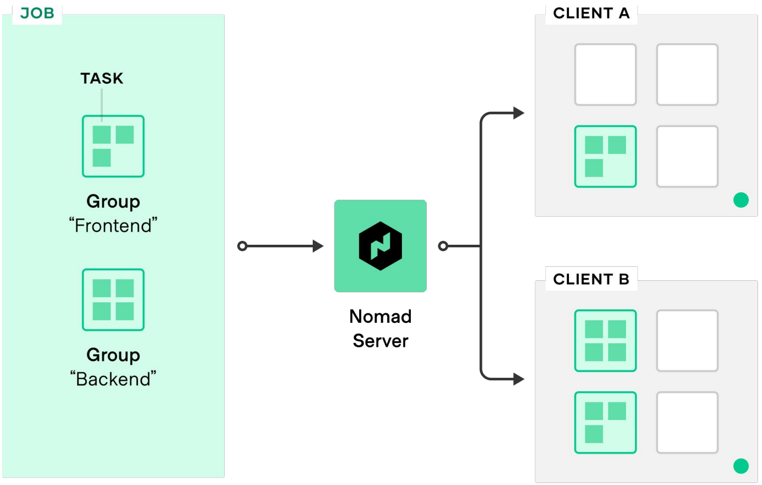
Core Concepts

Task The smallest unit of scheduling work. It could be a Docker container, a Java application, or batch processing.

Group A series of tasks that should be co-located on the same Nomad client. Tightly-coupled tasks in the same group can share the same network/storage

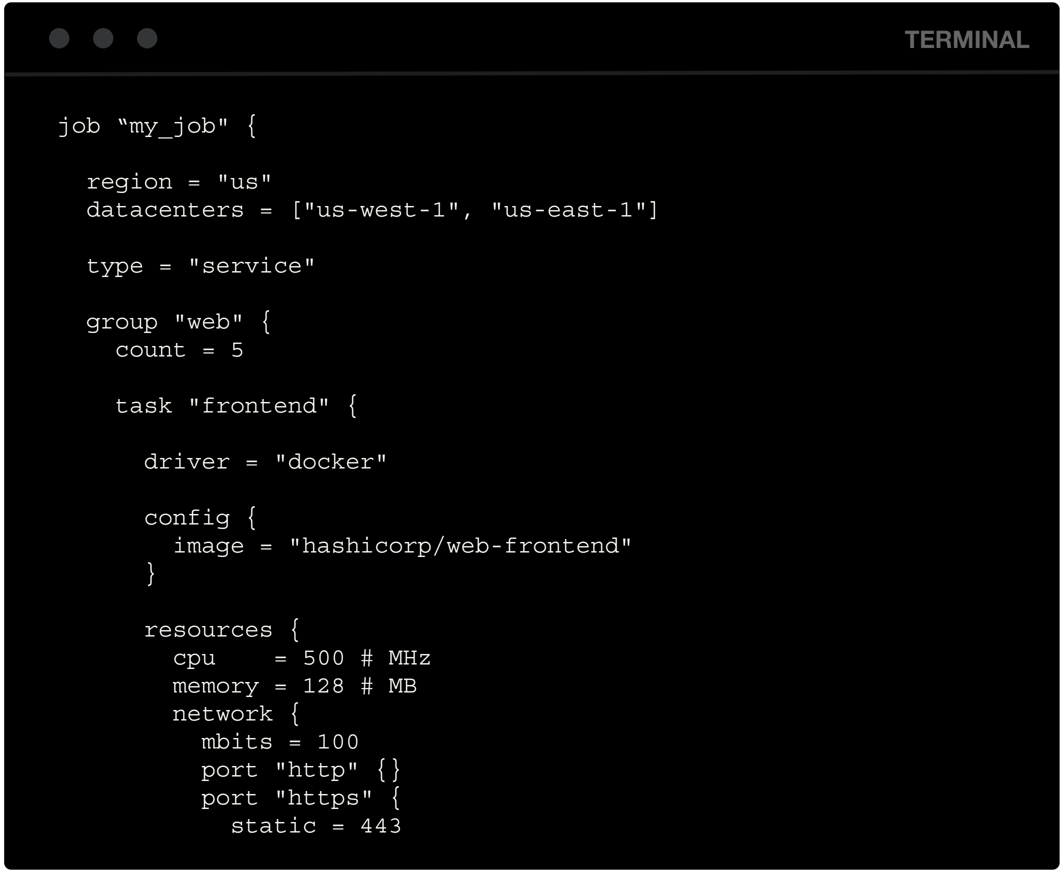
Job The declarative that defines the deployment rules for applications

Application The instance of a task group that are running on client



Application deployment as code

- Nomad **Job** - Declarative specification using HCL (like Terraform) or JSON.
- Set the deployment rules for applications fast and intuitively. Define tasks, images, resources, priorities, constraints, service registrations, secrets and other information required to deploy the application.



```
job “my_job" {

  region = "us"
  datacenters = ["us-west-1", "us-east-1"]

  type = "service"

  group "web" {
    count = 5

    task "frontend" {

      driver = "docker"

      config {
        image = "hashicorp/web-frontend"
      }

      resources {
        cpu    = 500 # MHz
        memory = 128 # MB
        network {
          mbits = 100
          port "http" {}
          port "https" {
            static = 443
```

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

Intuitive deployment rules

`job`

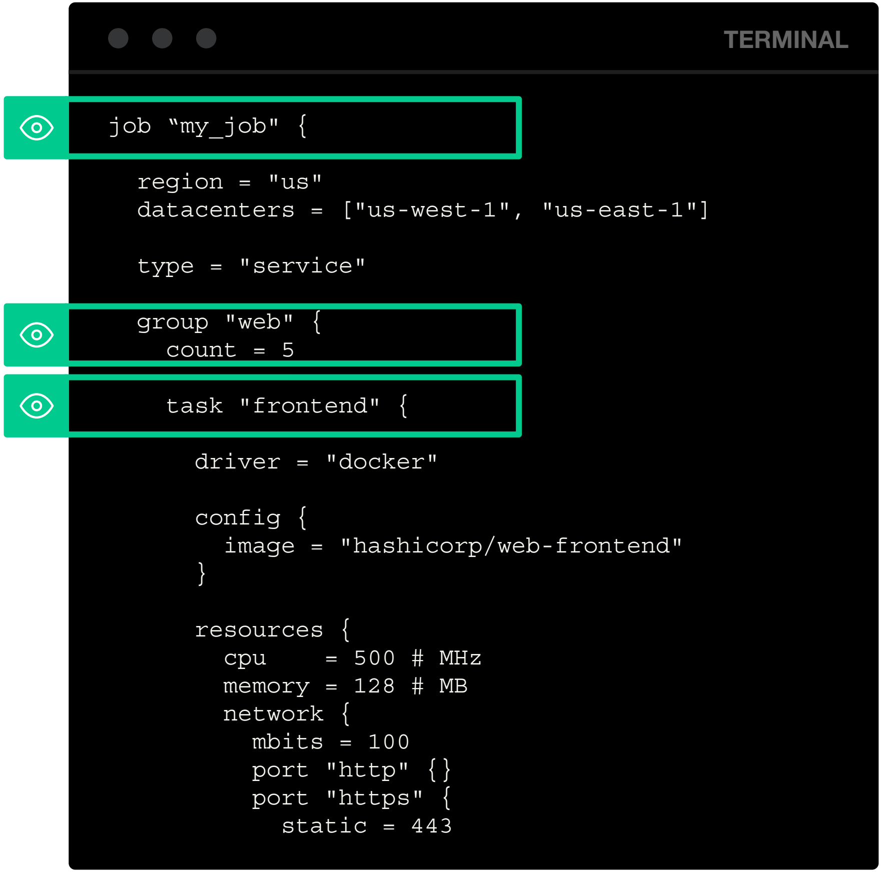
# define the deployment rules for applications

  `\_ group`

  # defines a series of tasks that should be co-located on the same Nomad client

    `\_ task`

    # defines a command, service, application or "set of work" to execute, such as a docker container, webapp or batch processing. Tasks are executed by their driver

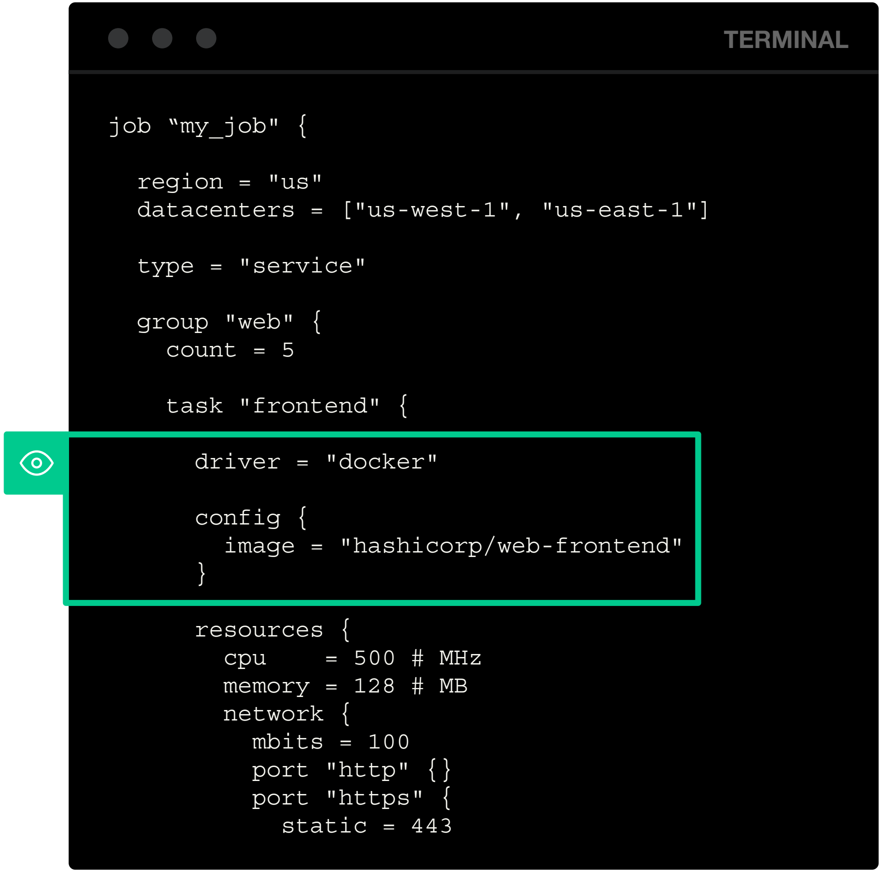


```
job “my_job" {

  region = "us"
  datacenters = ["us-west-1", "us-east-1"]

  type = "service"

  group "web" {
    count = 5

    task "frontend" {

      driver = "docker"

      config {
        image = "hashicorp/web-frontend"
      }

      resources {
        cpu    = 500 # MHz
        memory = 128 # MB
        network {
          mbits = 100
          port "http" {}
          port "https" {
            static = 443
```

Enable flexibility with extensible drivers

- **Task Drivers** execute tasks on the Nomad Client and provide resource isolation.
- First-class support of a broad set of workloads across all major operating systems.

containerd macOS



```
job “my_job" {

  region = "us"
  datacenters = ["us-west-1", "us-east-1"]

  type = "service"

  group "web" {
    count = 5

    task "frontend" {

      driver = "docker"

      config {
        image = "hashicorp/web-frontend"
      }

      resources {
        cpu    = 500 # MHz
        memory = 128 # MB
        network {
          mbits = 100
          port "http" {}
          port "https" {
            static = 443
```

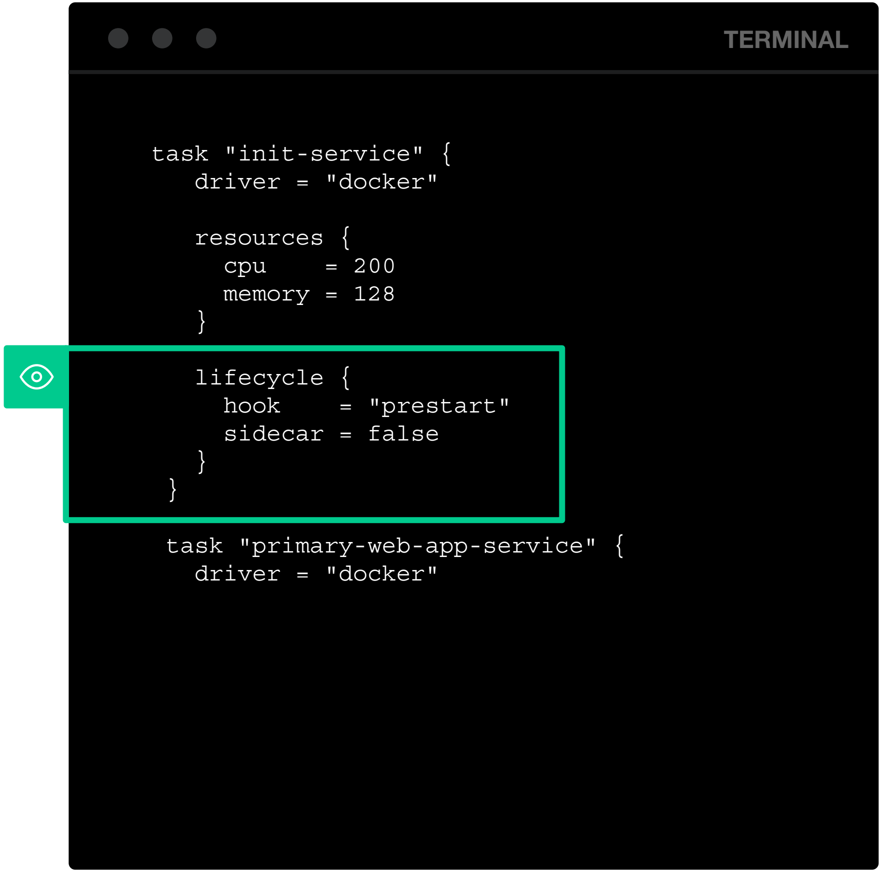
LINBIT SDS × HashiCorp Nomad

STORAGE DAY

Deploy interdependent applications

Deploy interdependent applications in their sequential orders with `lifecycle` hooks in the `Task` stanza:

- Startup & Initialization
  - `PreStart`
  - `PostStart`
- Sidecar
  - Boolean parameter
  - Deploy long-lived tasks within batch job
  - Run for duration of allocation
- Post deployment
  - `PostStop`



```
task "init-service" {
   driver = "docker"

   resources {
     cpu    = 200
     memory = 128
   }

   lifecycle {
     hook    = "prestart"
     sidecar = false
   }
 }

 task "primary-web-app-service" {
   driver = "docker"
```

www.linbit.com | www.nomadproject.io

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

Schedulers to run all types of workloads

- **Batch Scheduler** is optimized for fast placement for short-lived workload.

  Example: *Daily reports, transactions, billing invoices.*

- **Service Scheduler** is optimized for long-running workloads.

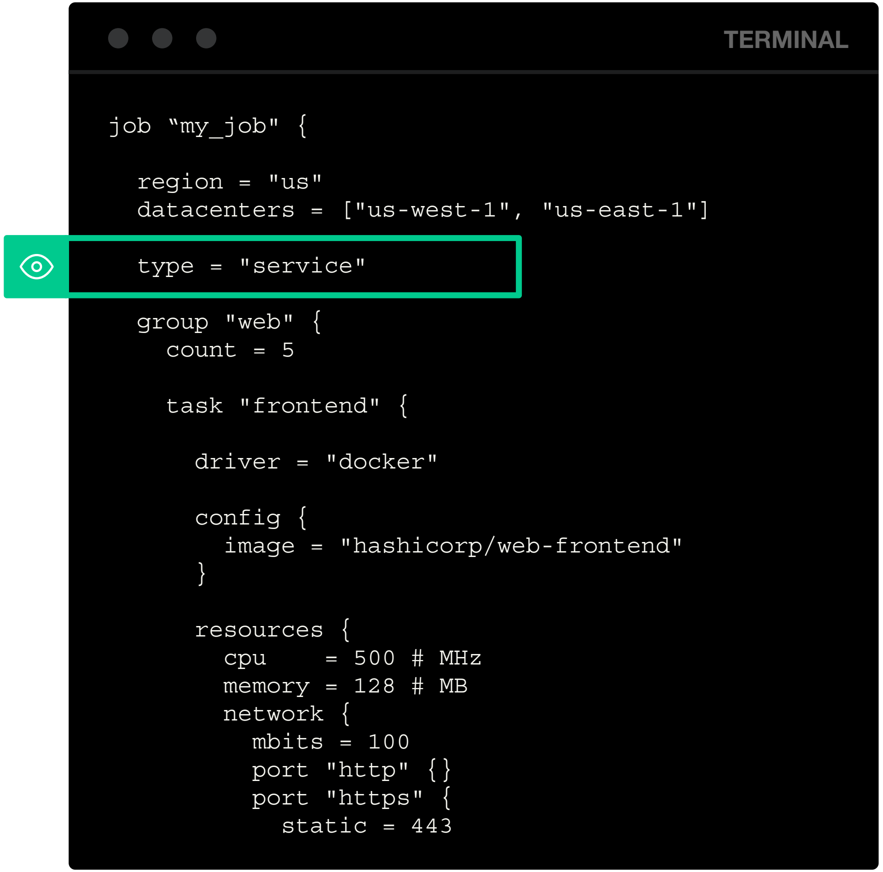
  Example: *Business-critical applications, customer facing webapps, database.*

- **System Scheduler** is optimized for background tasks.

  Example: *Logging/monitoring, security, background processes.*

- **System Batch** is optimized for background tasks.

  Example: *Same as System but in Batch Style.*

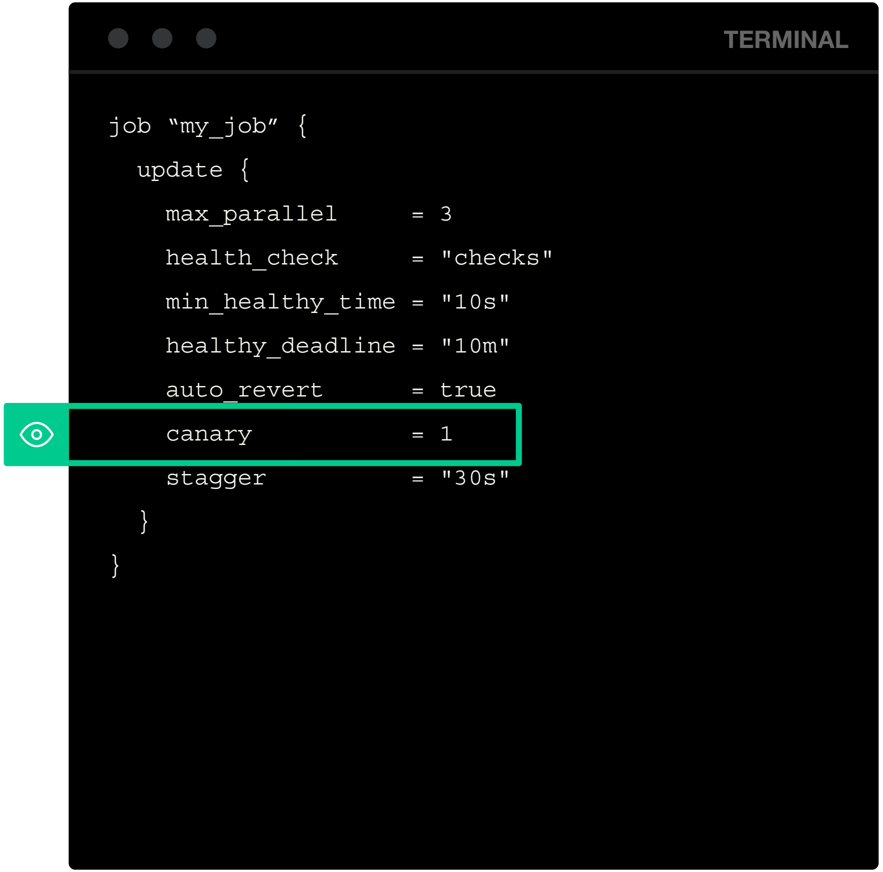


```
job “my_job" {

  region = "us"
  datacenters = ["us-west-1", "us-east-1"]

  type = "service"

  group "web" {
    count = 5

    task "frontend" {

      driver = "docker"

      config {
        image = "hashicorp/web-frontend"
      }

      resources {
        cpu    = 500 # MHz
        memory = 128 # MB
        network {
          mbits = 100
          port "http" {}
          port "https" {
            static = 443
```

Robust application update strategy

Automate job update and migration to minimize down time.

- **Rolling** Updates
- **Blue/Green** deployments
- **Canaries** deployments
- Updates can be gated on Consul health checks and automatically reverted.



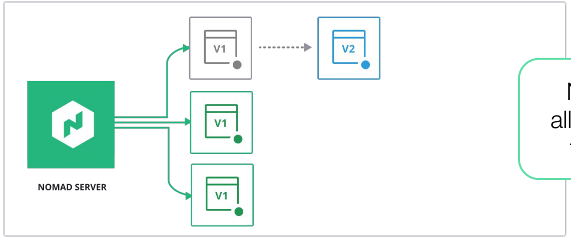
```
job “my_job” {
  update {
    max_parallel     = 3
    health_check     = "checks"
    min_healthy_time = "10s"
    healthy_deadline = "10m"
    auto_revert      = true
    canary           = 1
    stagger          = "30s"
  }
}
```

LINBIT SDS × HashiCorp Nomad

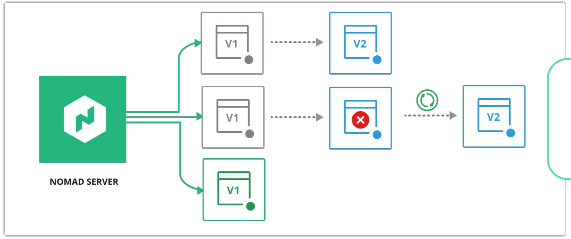
STORAGE DAY

Zero downtime deployment

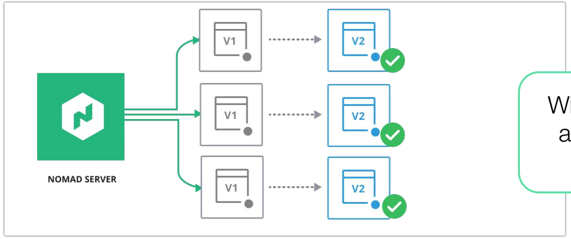
- **Reduce downtime** disruption by leveraging modern deployment strategies, such as rolling updates, blue/green, and canary deployments
- **Example:** Rolling Deployment



Nomad destroys the first allocation of V1 and creates the first allocation of V2.



If any V2 fails, Nomad will restart and reschedule the allocation.



When all V2 are created and are healthy, deployment is marked as successful.

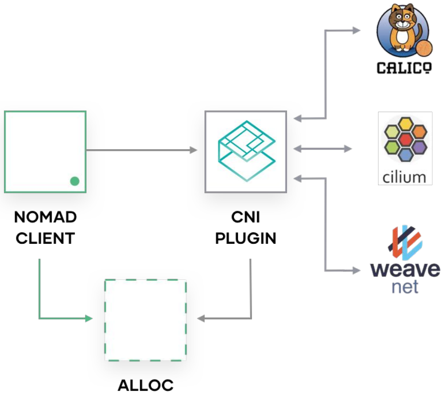
LINBIT SDS × HashiCorp Nomad

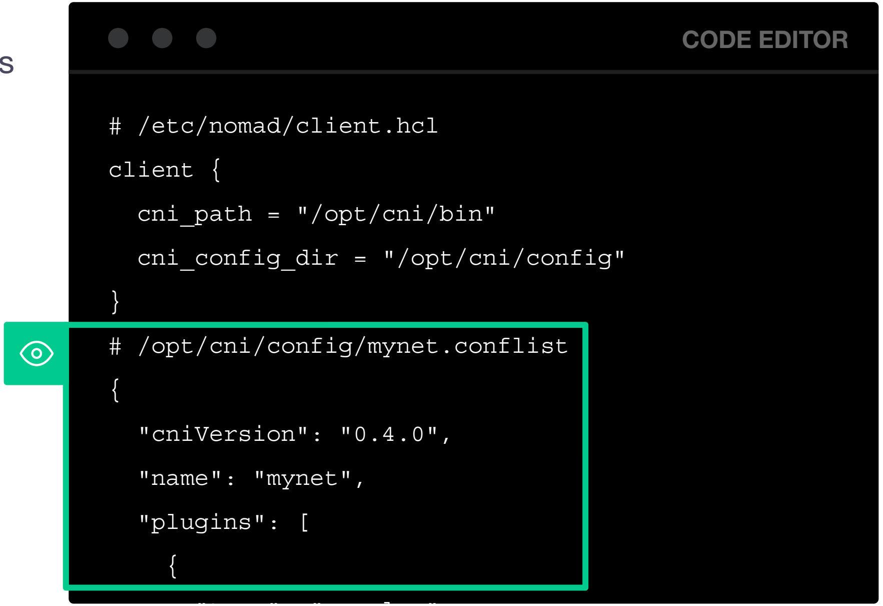
STORAGE DAY

Modern networking with simplicity

Deploy containerized applications on Nomad with their desired network configurations from any third-party vendors via the Container Networking Interface (CNI) plugin system.

- Operators set directories of CNI configurations,
- Nomad fingerprints and validate CNI networks on the host.
- Developers define network mode as “`CNI/xxxx`” in the job spec to use the CNI plugins.



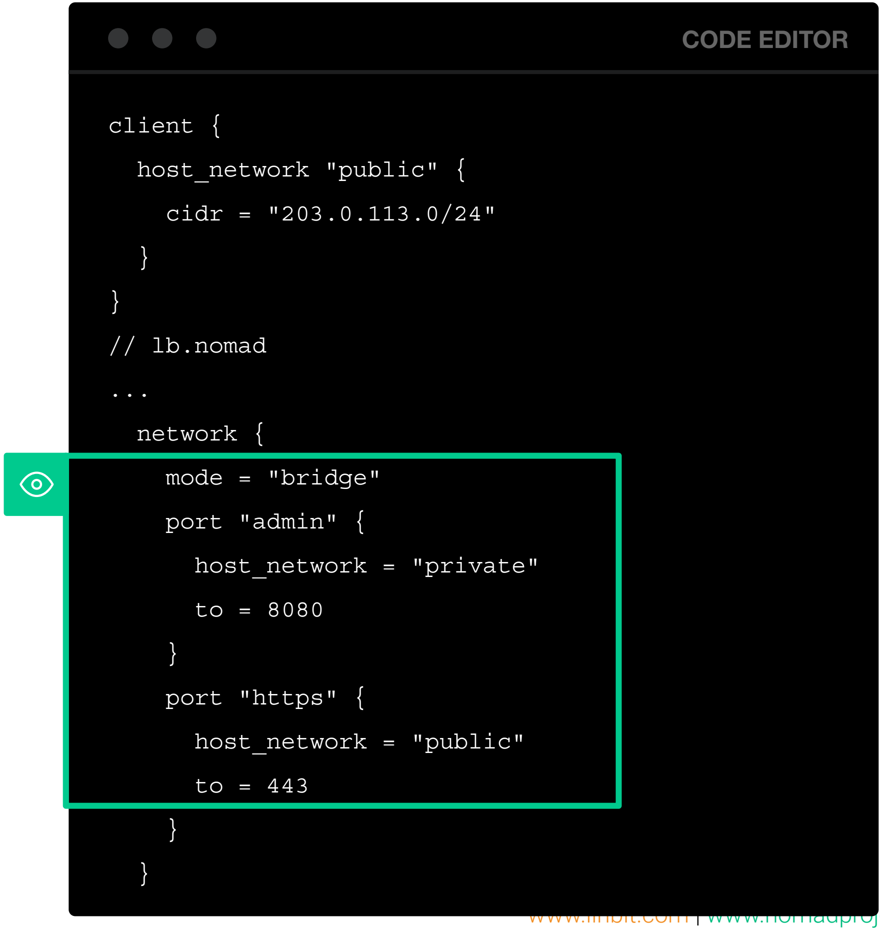


```
# /etc/nomad/client.hcl
client {
  cni_path = "/opt/cni/bin"
  cni_config_dir = "/opt/cni/config"
}
# /opt/cni/config/mynet.conflist
{
  "cniVersion": "0.4.0",
  "name": "mynet",
  "plugins": [
    {
```

Multi-interface networking

Allow applications to bind to multiple network interfaces to support communications across multiple networks.

- Define multiple host networks in Nomad Client configuration.
- Define `host_network` to the port block to select a named host network.
- New network env variables to support multiple interfaces.

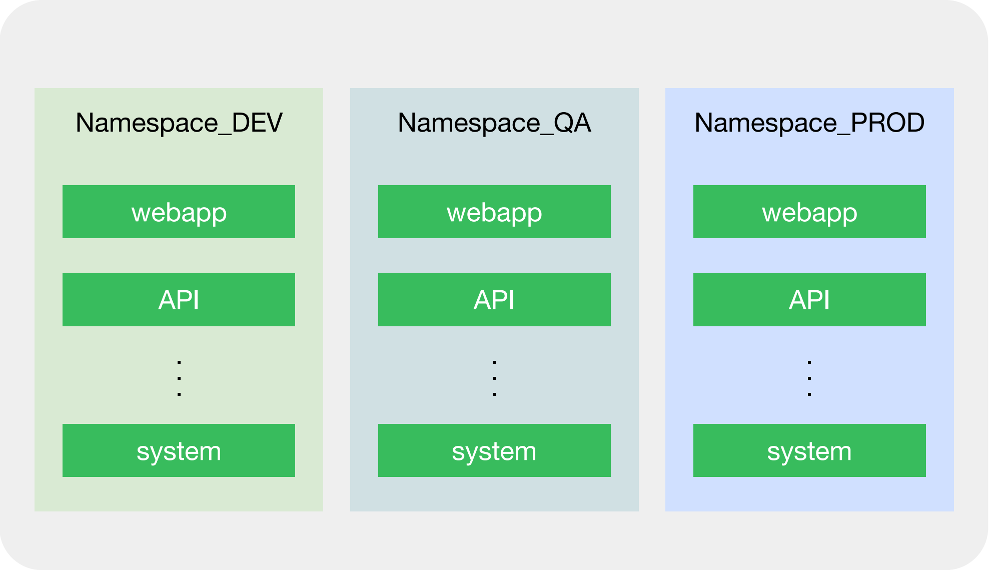


```
client {
  host_network "public" {
    cidr = "203.0.113.0/24"
  }
}
// lb.nomad
...
  network {
    mode = "bridge"
    port "admin" {
      host_network = "private"
      to = 8080
    }
    port "https" {
      host_network = "public"
      to = 443
    }
  }
```

Namespaces

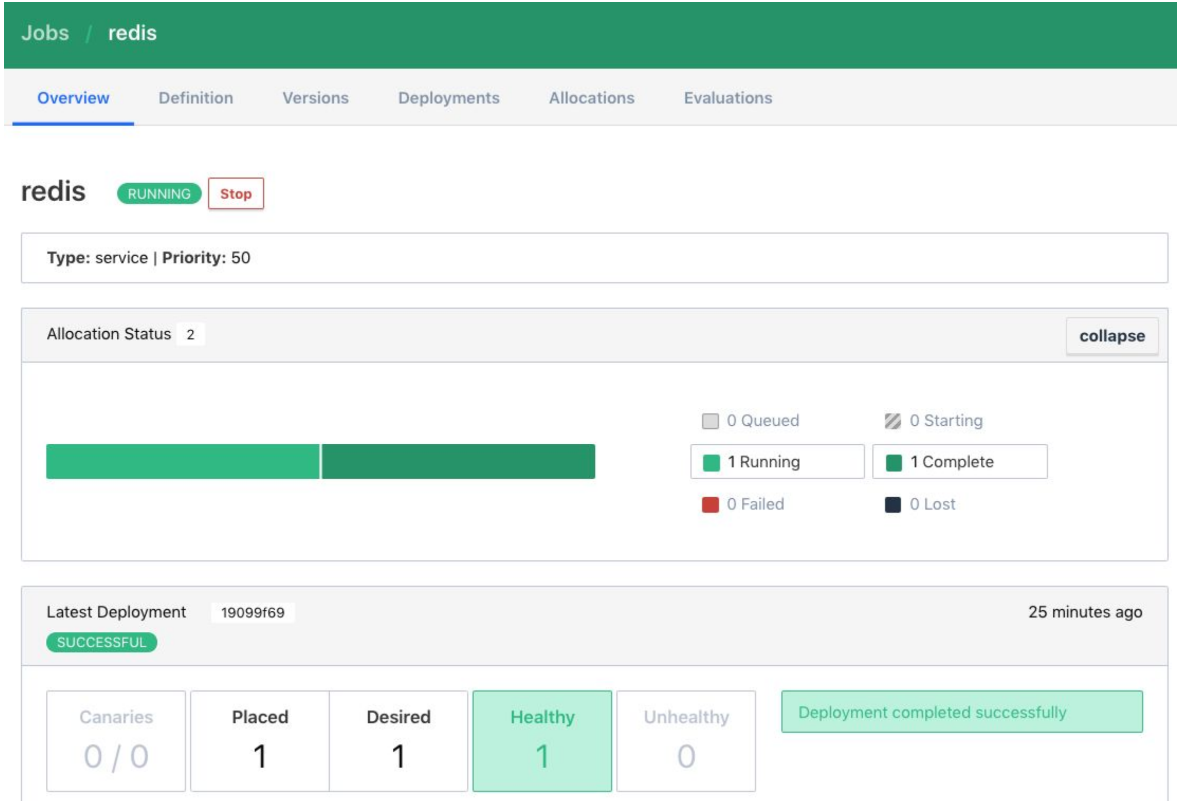
Allow many teams and projects to share a single multi-region Nomad deployment without conflict.

- ACL policies provide enforcement of namespaces.
- Job IDs are required to be unique within namespaces but not across namespaces.
- Namespaces are automatically replicated across regions for easy, centralized administration at global scale.



Namespace_DEV	Namespace_QA	Namespace_PROD
webapp	webapp	webapp
API	API	API
⋮	⋮	⋮
system	system	system

View job with deployment status



Jobs / redis

Overview | Definition | Versions | Deployments | Allocations | Evaluations

redis

RUNNING Stop

Type: service | **Priority:** 50

Allocation Status 2 collapse

□ 0 Queued ▨ 0 Starting
🟩 1 Running 🟩 1 Complete
🟥 0 Failed ■ 0 Lost

Latest Deployment 19099f69 25 minutes ago

SUCCESSFUL

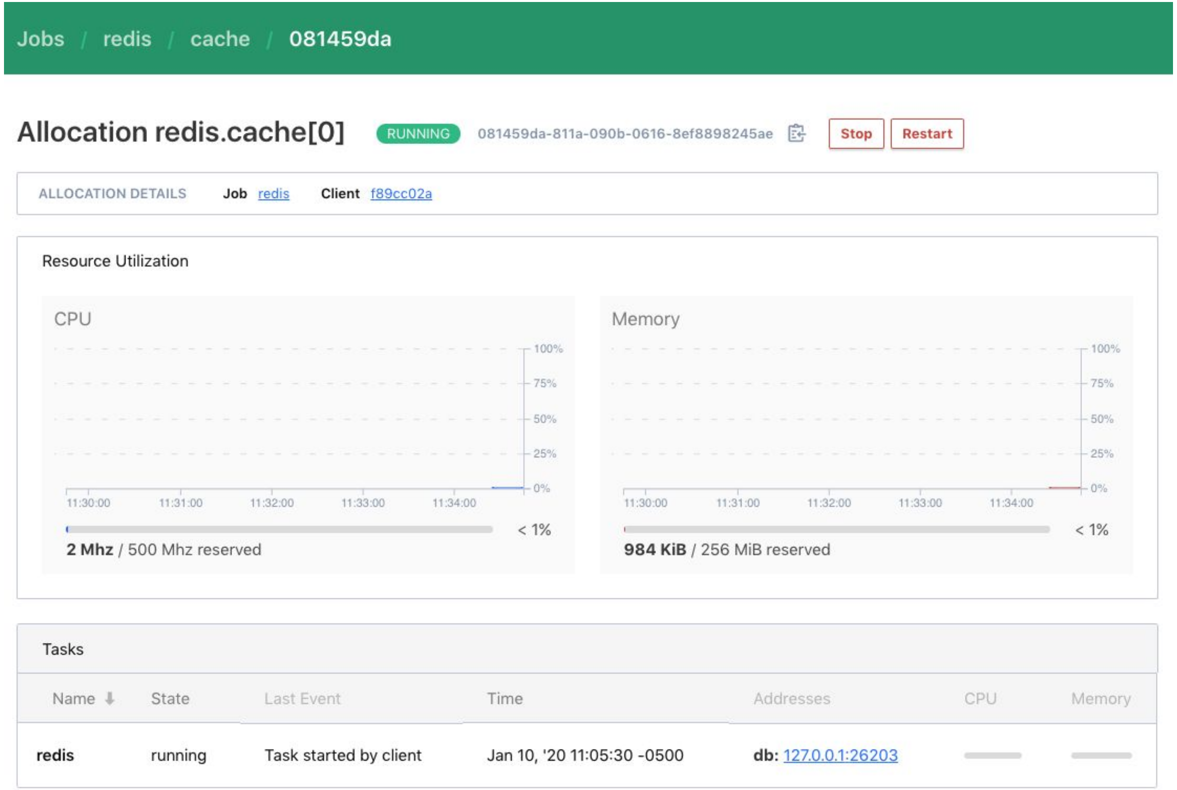
Canaries	Placed	Desired	Healthy	Unhealthy
0 / 0	1	1	1	0

Deployment completed successfully

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

View task status



Jobs / redis / cache / 081459da

Allocation redis.cache[0]

RUNNING 081459da-811a-090b-0616-8ef8898245ae

ALLOCATION DETAILS Job redis Client f89cc02a

Resource Utilization

2 Mhz / 500 Mhz reserved

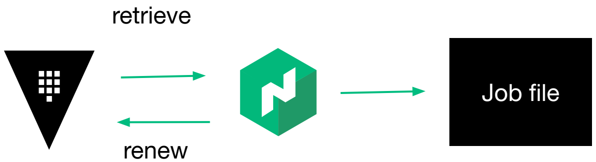
984 KiB / 256 MiB reserved

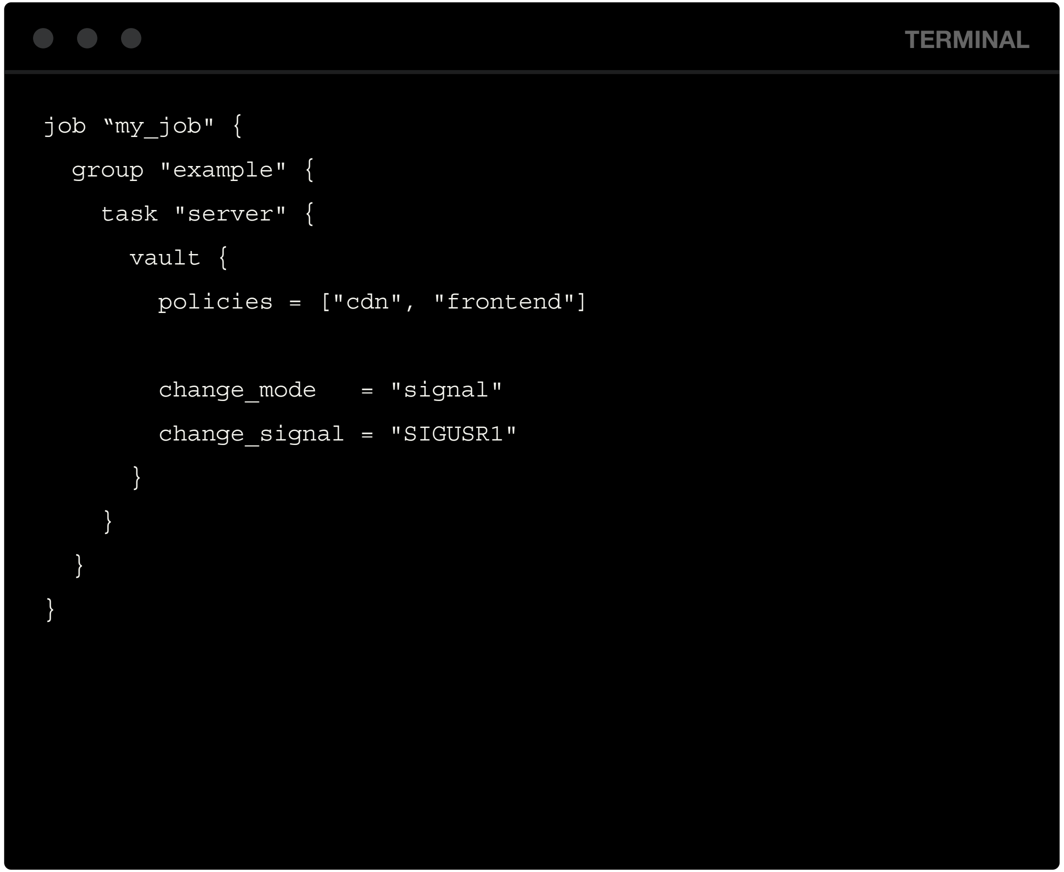
Tasks

Name ⬇	State	Last Event	Time	Addresses	CPU	Memory
redis	running	Task started by client	Jan 10, '20 11:05:30 -0500	**db:** 127.0.0.1:26203		

Secrets management with Vault

- Automatic Vault token retrieval.
- Automatic Vault token renewal.
- Automatic secret retrieval and renewal via template stanza.



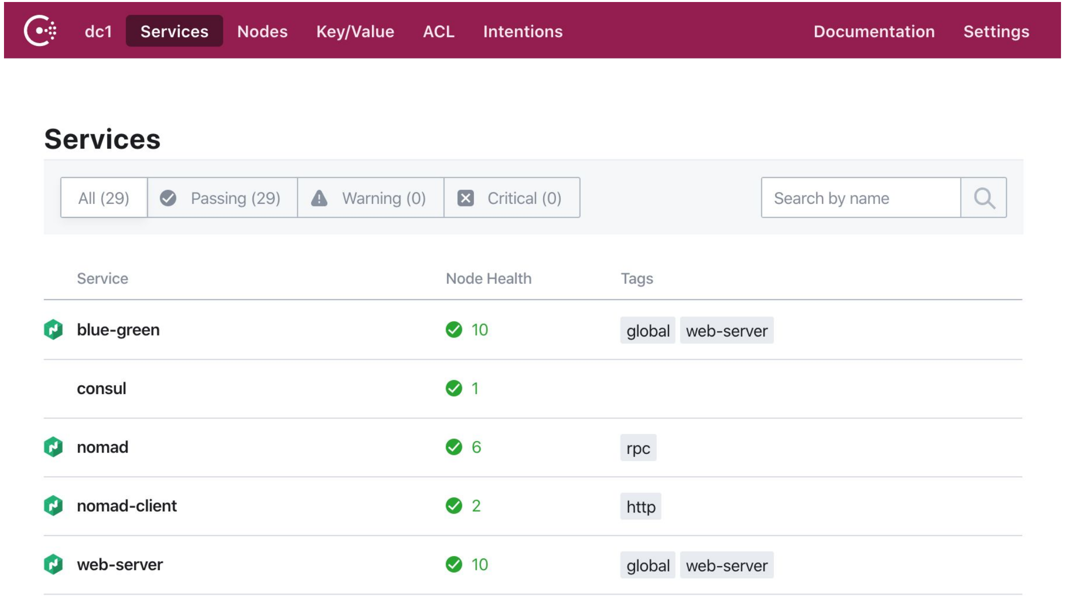


```
job “my_job" {
  group "example" {
    task "server" {
      vault {
        policies = ["cdn", "frontend"]

        change_mode   = "signal"
        change_signal = "SIGUSR1"
      }
    }
  }
}
```

LINBIT SDS × HashiCorp Nomad
STORAGE DAY

Service discovery with Consul

Built-in service discovery, registration, and health check monitoring for all applications deployed under Nomad.



Services

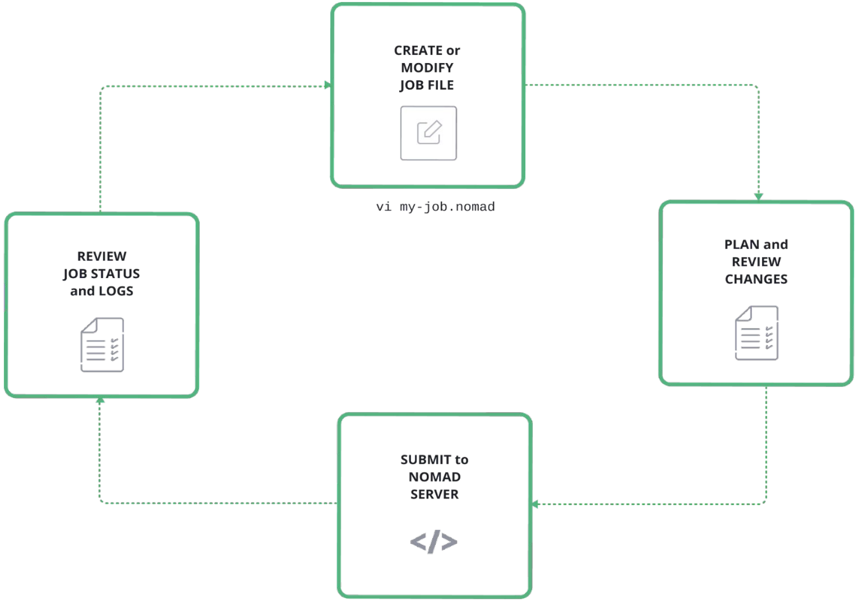
Service	Node Health	Tags
blue-green	✓ 10	global web-server
consul	✓ 1	
nomad	✓ 6	rpc
nomad-client	✓ 2	http
web-server	✓ 10	global web-server

www.linbit.com | www.nomadproject.io

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

Consistent workflow for job lifecycle management



```
vi my-job.nomad
```

```
nomad job plan my-job.nomad
```

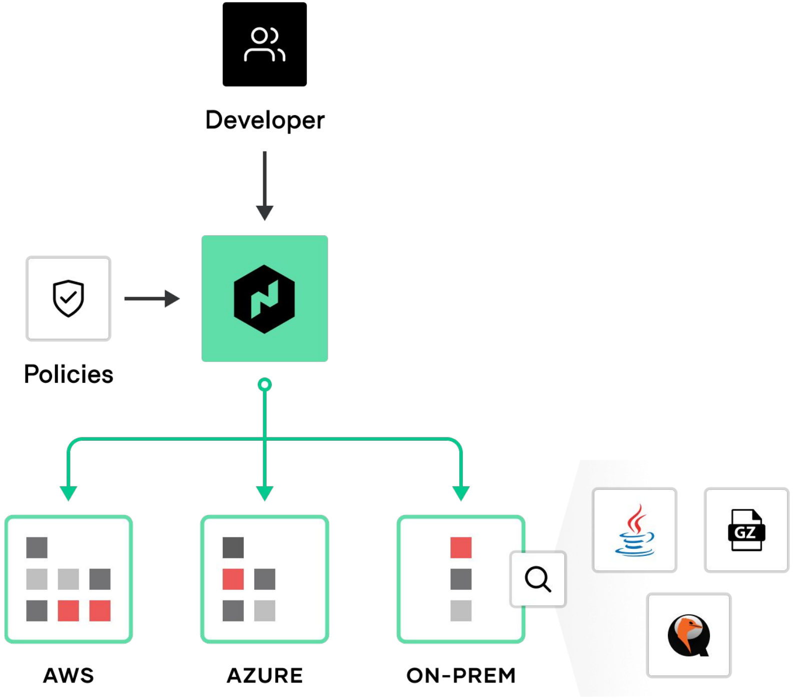
```
nomad job run my-job.nomad
```

```
nomad job status
nomad alloc status
nomad alloc jobs
```

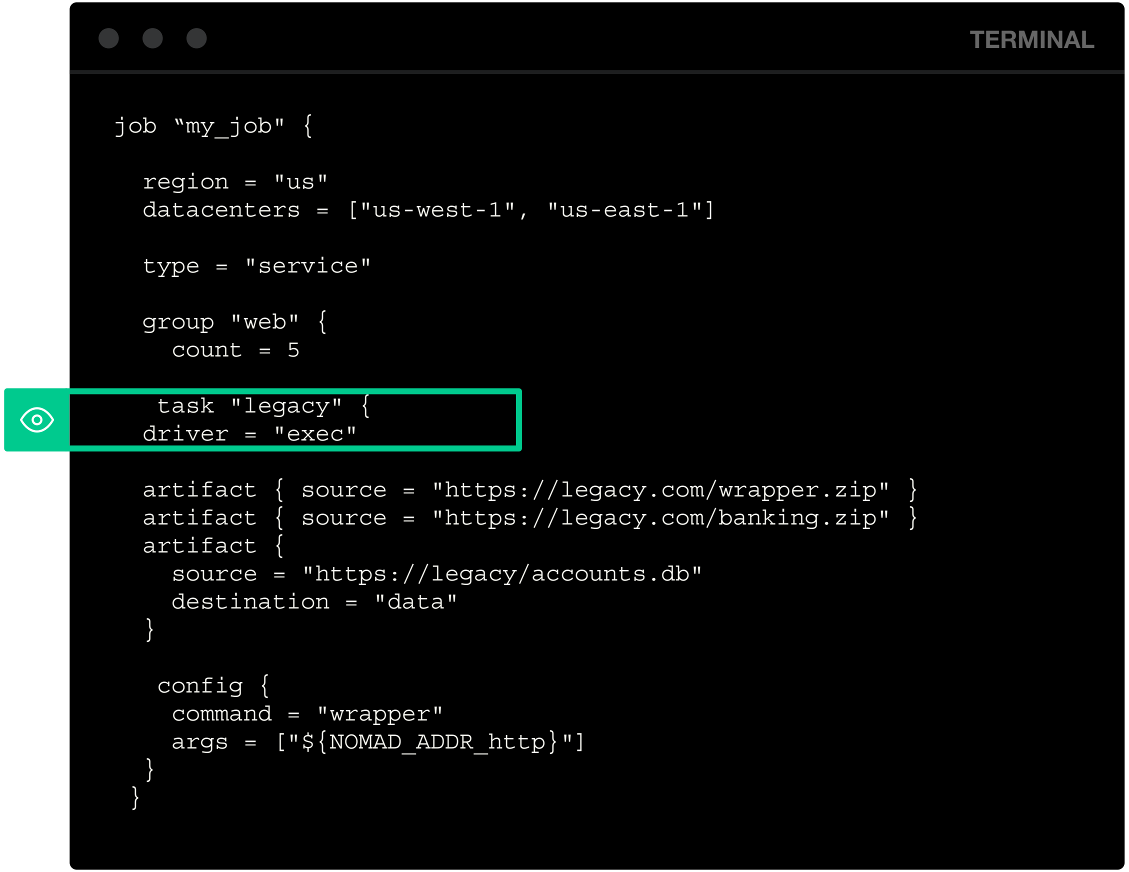
Orchestrate non-containers as containers

Non-Containerized Application Orchestration

- **Achieve immediate results** by integrating Nomad with existing infrastructure.
- **Reduce cost** by densely scheduling applications to underutilized resources.
- **Zero downtime deployment** by bringing the benefits of modern deployment strategies to legacy applications.



Single workflow for any application



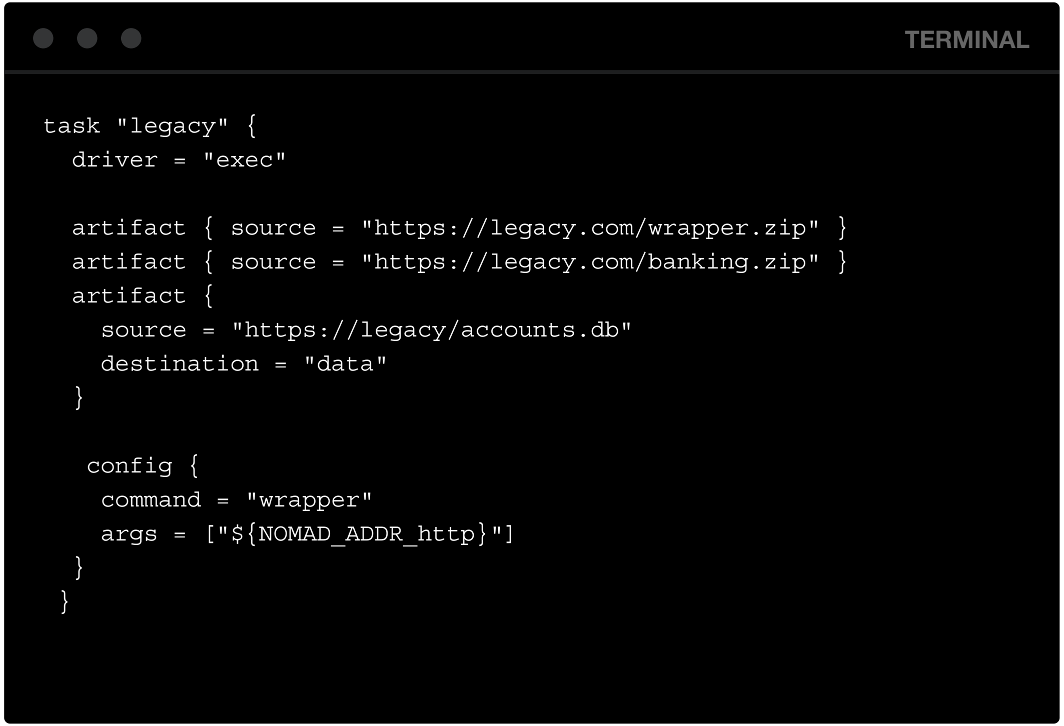
```
job “my_job" {

  region = "us"
  datacenters = ["us-west-1", "us-east-1"]

  type = "service"

  group "web" {
    count = 5

   task "legacy" {
  driver = "exec"

  artifact { source = "https://legacy.com/wrapper.zip" }
  artifact { source = "https://legacy.com/banking.zip" }
  artifact {
    source = "https://legacy/accounts.db"
    destination = "data"
  }

   config {
    command = "wrapper"
    args = ["${NOMAD_ADDR_http}"]
  }
 }
```

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

Enable flexibility with extensible driver

- **Task Drivers** execute tasks on the Nomad Client and provide resource isolation.
- First-class support of a broad set of workloads across all major operating systems.



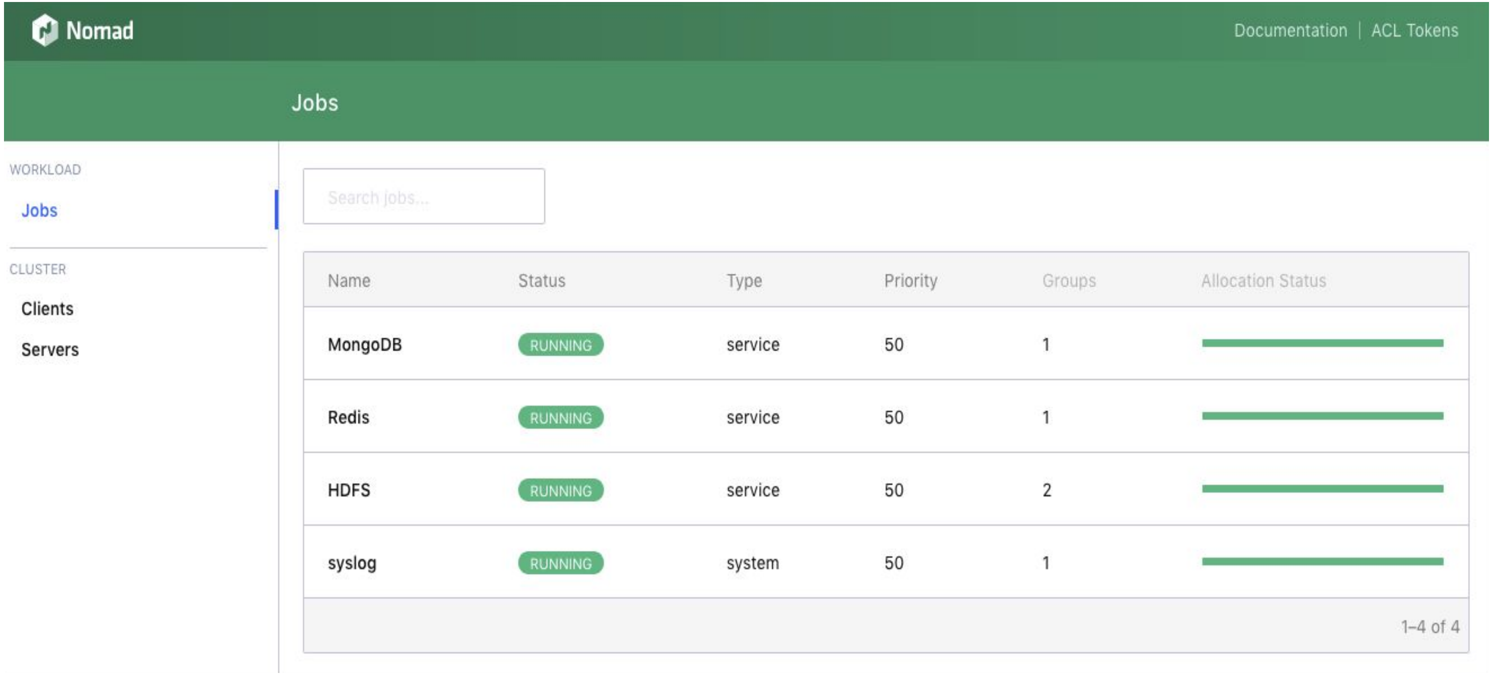
```
task "legacy" {
  driver = "exec"

  artifact { source = "https://legacy.com/wrapper.zip" }
  artifact { source = "https://legacy.com/banking.zip" }
  artifact {
    source = "https://legacy/accounts.db"
    destination = "data"
  }

   config {
    command = "wrapper"
    args = ["${NOMAD_ADDR_http}"]
  }
 }
```

www.linbit.com | www.nomadproject.io

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

Improved visibility - Consolidated control plan



Name	Status	Type	Priority	Groups	Allocation Status
MongoDB	RUNNING	service	50	1	🟩
Redis	RUNNING	service	50	1	🟩
HDFS	RUNNING	service	50	2	🟩
syslog	RUNNING	system	50	1	🟩

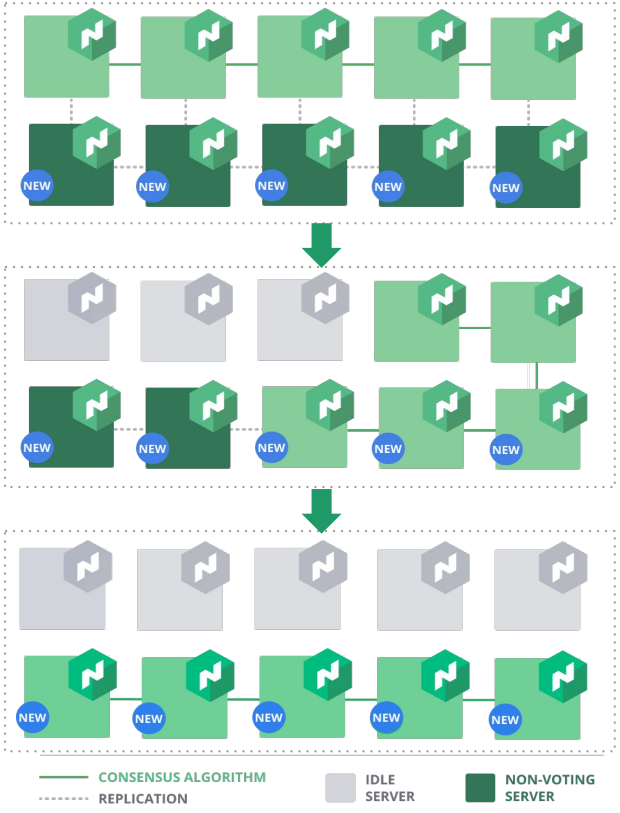
1–4 of 4

Nomad Enterprise

Automated upgrade

Prevent downtime with automated server upgrade.

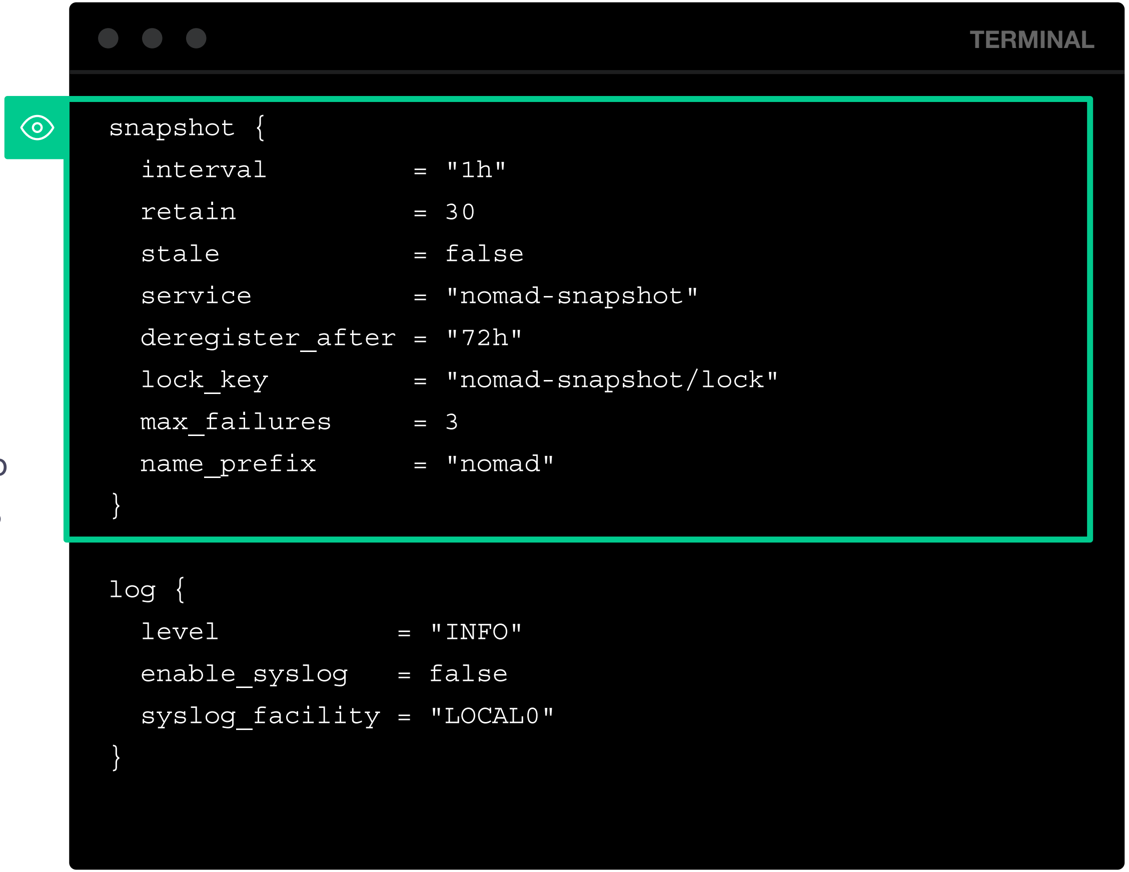
- New servers with higher version auto-join as non-voting servers.
- Autopilot waits until enough new servers have joined, and then starts to demote the old servers and promotes the new versions until the transition is complete.



Automated backups

Enables automatic snapshotting and storing of Nomad cluster state locally or to a cloud storage of choice, like Amazon S3 or Azure Blob Storage.

The `snapshot agent` subcommand starts a process that takes snapshots of the state of the Nomad servers and saves them locally, or pushes them to an optional remote storage service.

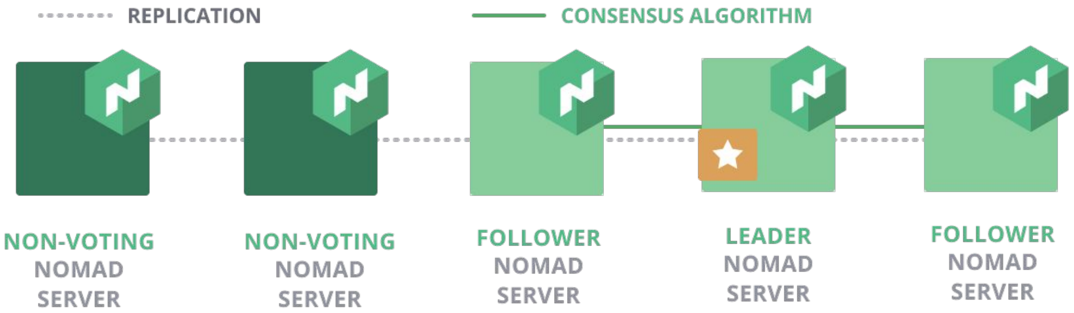


```
snapshot {
  interval         = "1h"
  retain           = 30
  stale            = false
  service          = "nomad-snapshot"
  deregister_after = "72h"
  lock_key         = "nomad-snapshot/lock"
  max_failures     = 3
  name_prefix      = "nomad"
}

log {
  level           = "INFO"
  enable_syslog   = false
  syslog_facility = "LOCAL0"
}
```

Enhanced read scalability

Increase the scale that a Nomad server group can handle for read intensive operations, without impacting performance.

- Non-voting servers receive the replication streams but do not vote in the quorum.
- Non voting servers can act as scheduling workers to improve scheduling throughput.



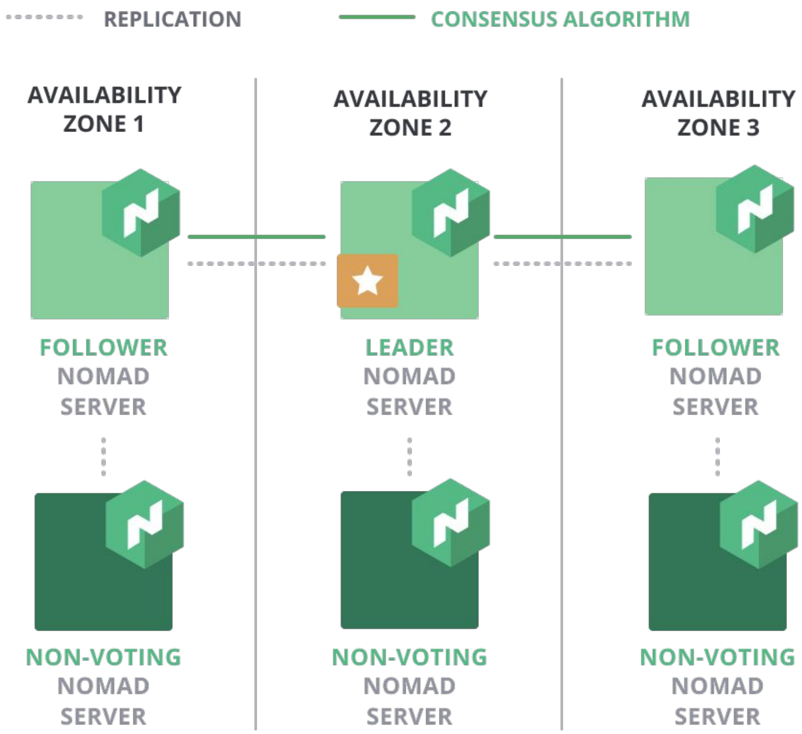
LINBIT SDS × HashiCorp Nomad

STORAGE DAY

Redundancy zones

Enable redundancy deployment in each availability zone, without impacting performance.

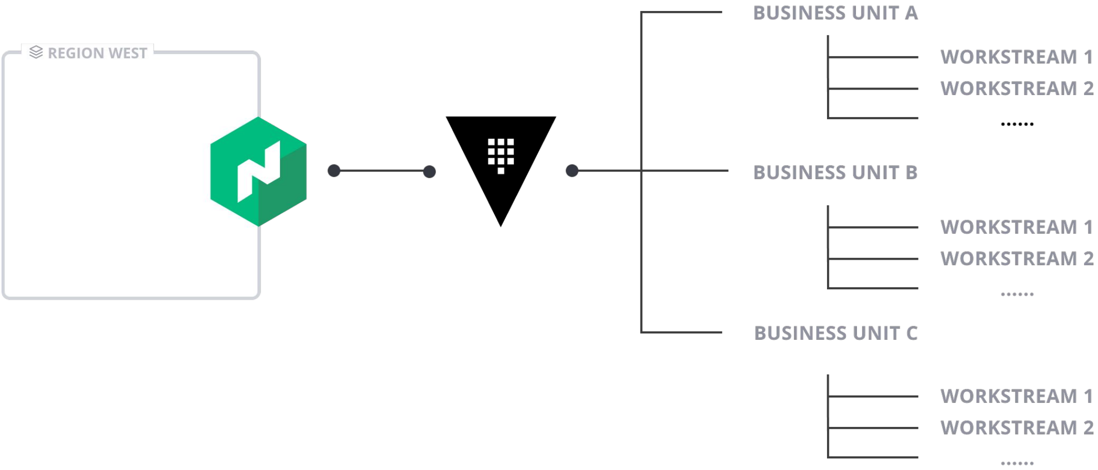
- One voting, one non-voting server per Zone.
- Auto-promote the non-voting server to a voting server in the cluster, when a voting server is lost.



Multi-Vault Namespaces

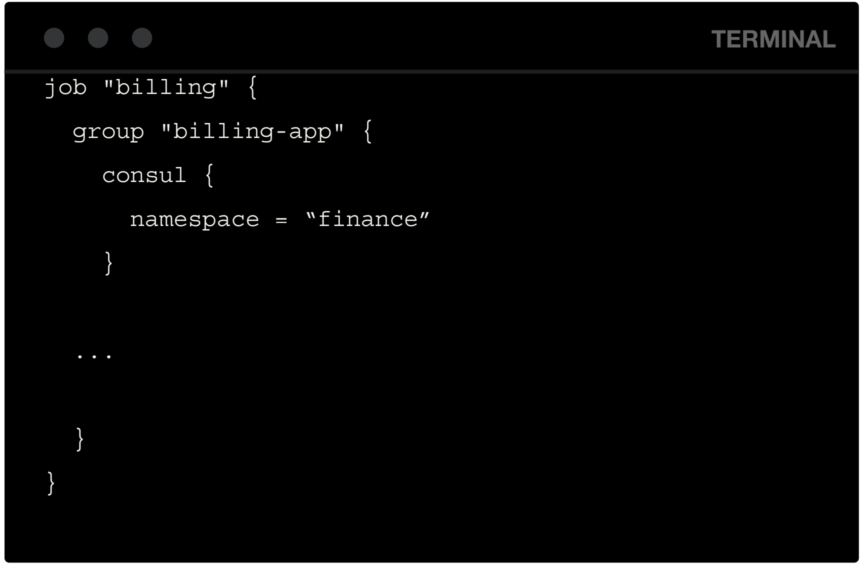
Allow one Nomad cluster to span multiple Vault namespaces for simplified topology and fleet consolidation

- Users can specify a Vault namespace in their job file



Consul Namespace Support

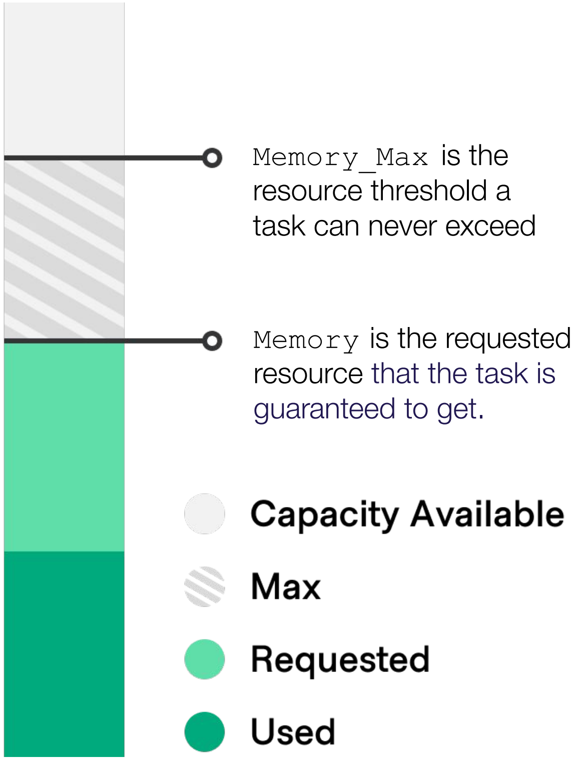
- Enable applications running in Nomad to register their Consul services in specific Consul Namespaces via Nomad job files.
- Improve the interoperability between Nomad and Consul while simplifying the adoption of hierarchical networks powered by Consul Enterprise.



```
job "billing" {
  group "billing-app" {
    consul {
      namespace = “finance”
    }


  ...


  }
}
```

Memory Oversubscription

- Allow tasks to use more than their scheduled amount of memory without OOM errors
- Improve cluster efficiency and bin packing
- Add additional guards to protect against client-level OOM errors



`Memory_Max` is the resource threshold a task can never exceed

`Memory` is the requested resource that the task is guaranteed to get.

○ Capacity Available

▨ Max

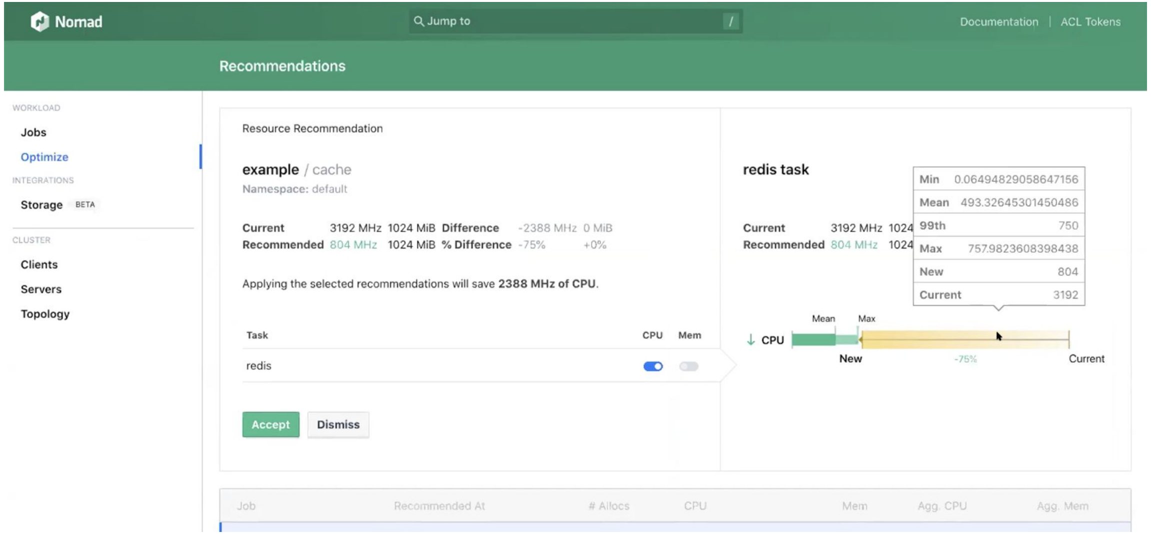
● Requested

● Used

Dynamic Application Sizing

Enables Nomad to right-size applications to their most efficient levels of resource consumption.

Reduce spend and drive cost savings intelligently and non-disruptively at scale.



Resource Recommendation

example / cache

Namespace: default

Current	3192 MHz	1024 MiB	Difference	-2388 MHz	0 MiB
Recommended	804 MHz	1024 MiB	% Difference	-75%	+0%

Applying the selected recommendations will save **2388 MHz of CPU.**

Task	CPU	Mem
redis	●	○

redis task

Current	3192 MHz	1024
Recommended	804 MHz	1024

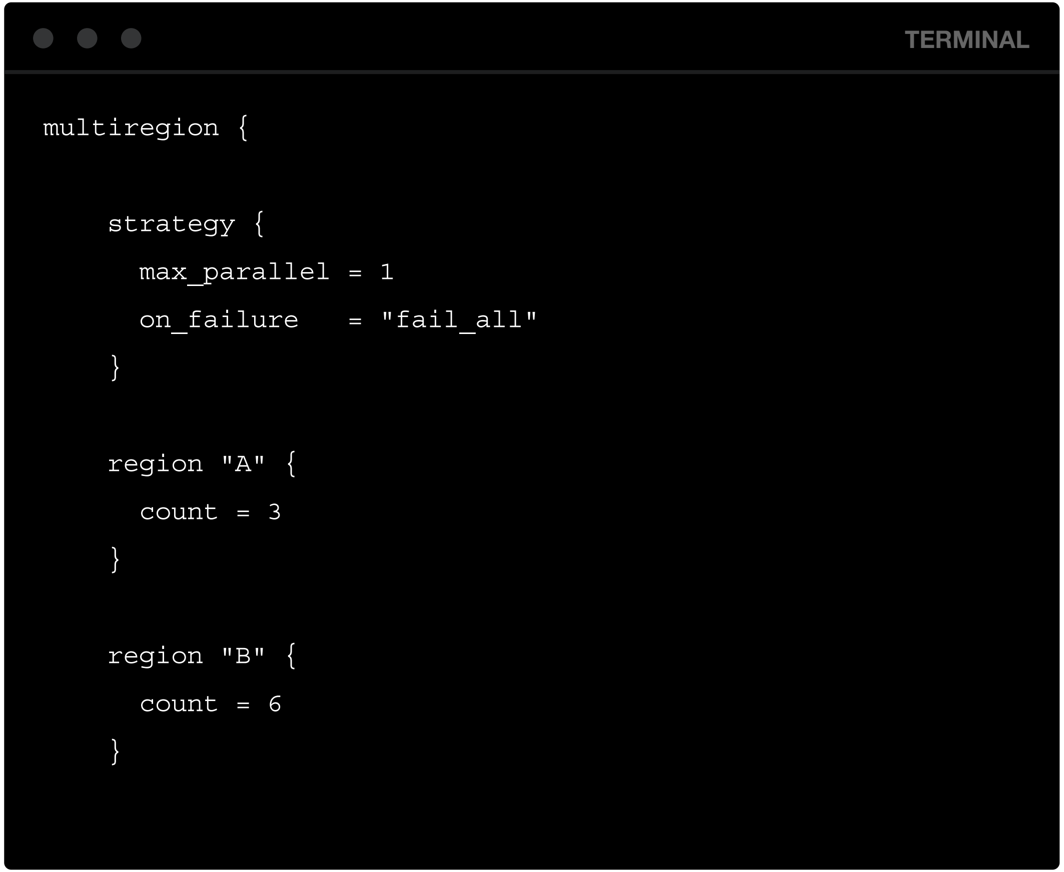
Min	0.06494829058647156
Mean	493.32645301450486
99th	750
Max	757.9823608398438
New	804
Current	3192

Job	Recommended At	# Allocs	CPU	Mem	Agg. CPU	Agg. Mem

Multi-cluster deployment

Example:

- Simple tweaks to existing single cluster job spec.
- Configurable rollout and rollback strategies.

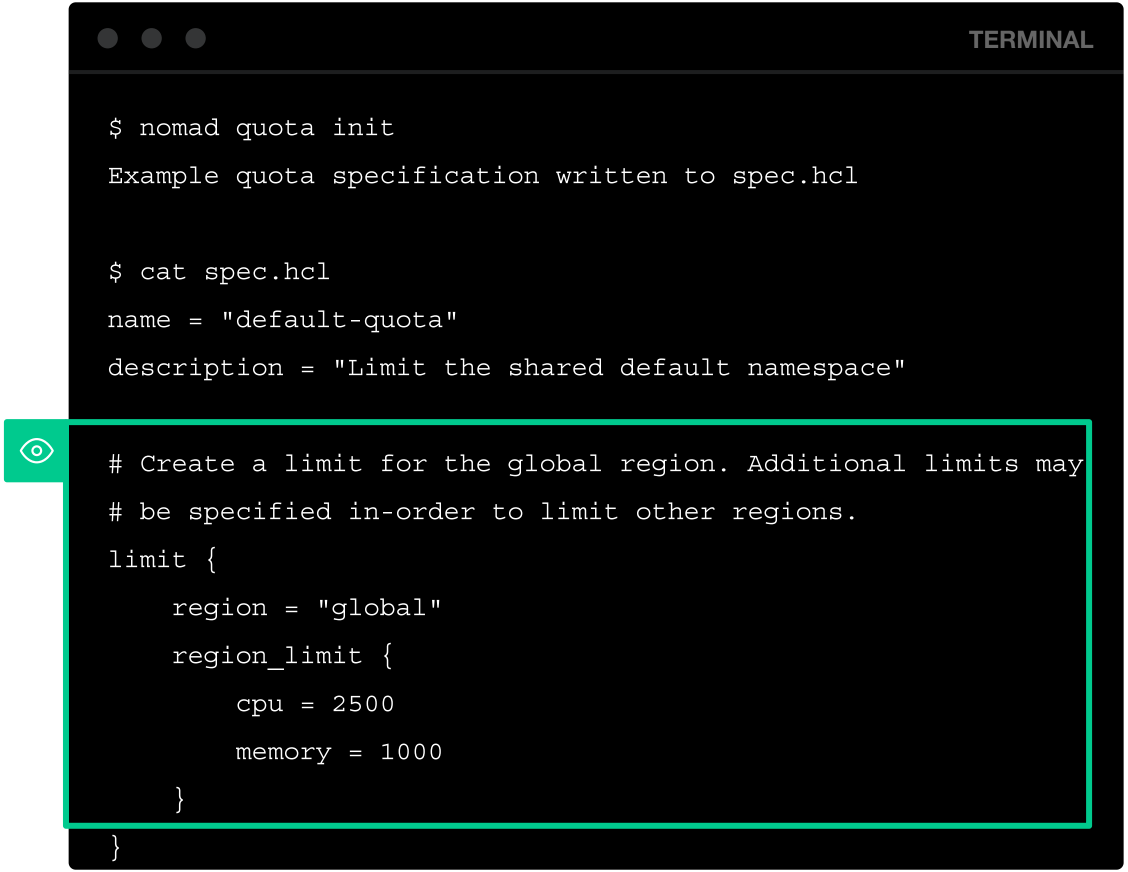


```
multiregion {

    strategy {
      max_parallel = 1
      on_failure   = "fail_all"
    }

    region "A" {
      count = 3
    }

    region "B" {
      count = 6
    }
```

Resource quotas

Enable governance across tenants by limiting resource consumption on a per namespace basis.

- Resources limits are defined independently per region.
- Single resource quota an be applied to multiple namespaces.
- Automatic replication across regions for easy, centralized administration at global scale.



```
$ nomad quota init
Example quota specification written to spec.hcl

$ cat spec.hcl
name = "default-quota"
description = "Limit the shared default namespace"

# Create a limit for the global region. Additional limits may
# be specified in-order to limit other regions.
limit {
    region = "global"
    region_limit {
        cpu = 2500
        memory = 1000
    }
}
```

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

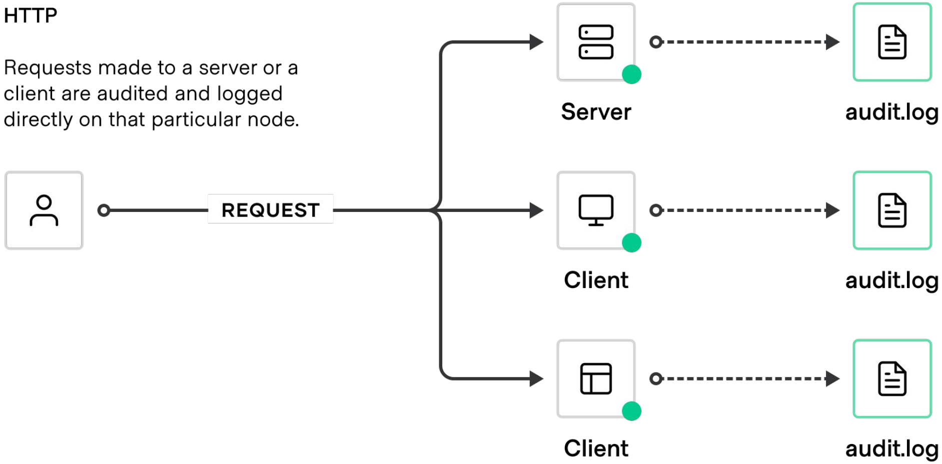
Audit logging

Secure clusters by providing administrators with a complete set of records for all user-issued actions.

- HTTP API-based logging system.
- Event broker that runs on all Nomad agents (clients & servers).
- Writes events to an audit log file local to each node.
- JSON response and request objects of each API transaction.
- Identical mental model as API for operator ease in digestibility.

HTTP

Requests made to a server or a client are audited and logged directly on that particular node.

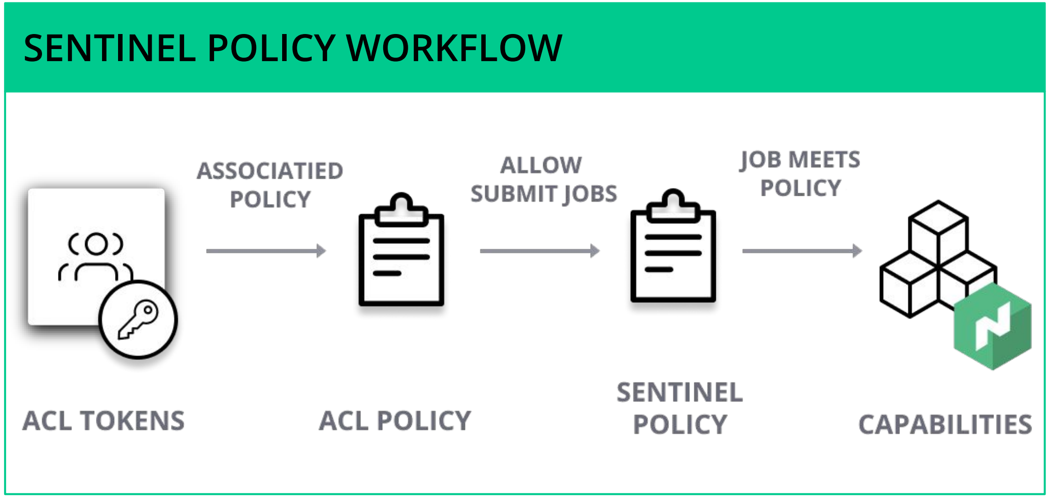


www.linbit.com | www.nomadproject.io

Sentinel policies

Enforce fine-grained policy as code with Sentinel integration.

- Integrate with the Nomad ACL system.
- Policies are enforced on job submission and job update.
- Sentinel policies are automatically replicated across regions for easy, centralized administration at global scale.



SENTINEL POLICY WORKFLOW

ACL TOKENS

ASSOCIATIED POLICY

ACL POLICY

ALLOW SUBMIT JOBS

SENTINEL POLICY

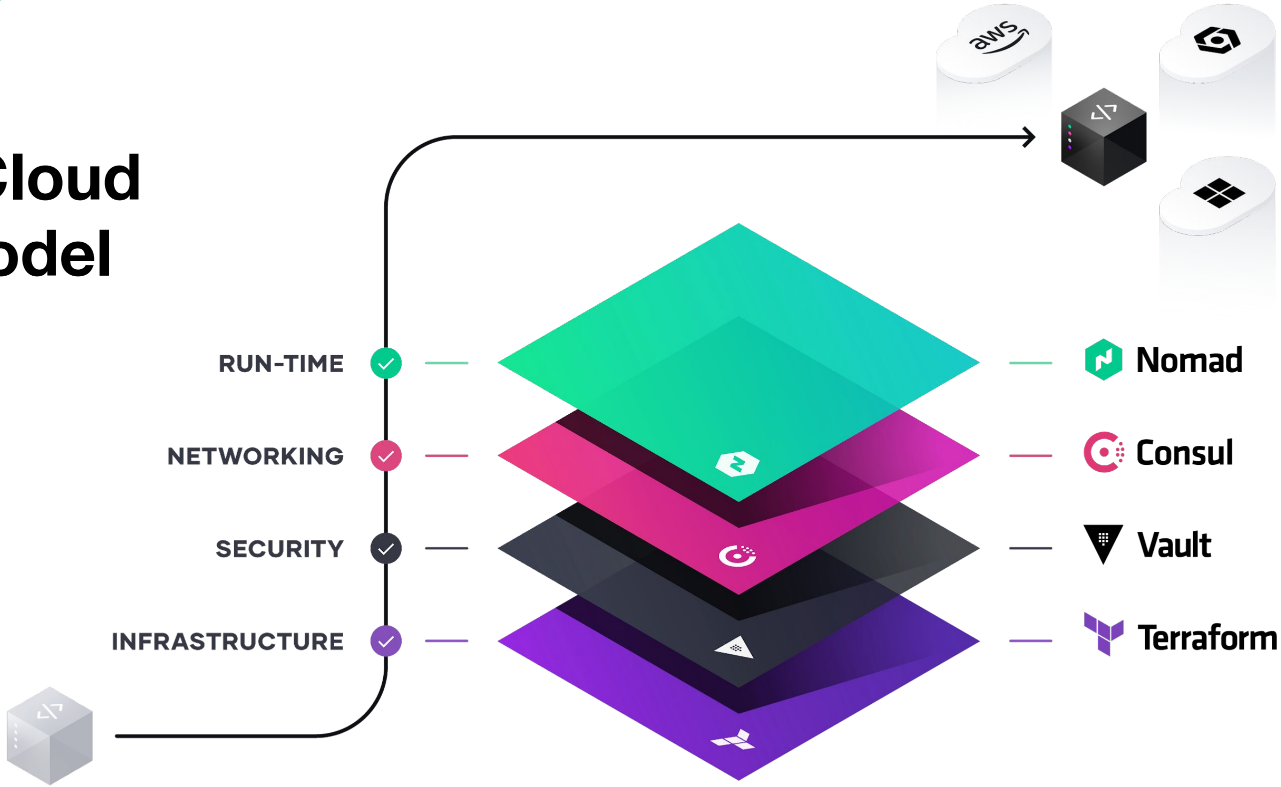
JOB MEETS POLICY

CAPABILITIES

LINBIT SDS × HashiCorp Nomad

STORAGE DAY

A Common Cloud Operating Model



RUN-TIME ✓ — Nomad

NETWORKING ✓ — Consul

SECURITY ✓ — Vault

INFRASTRUCTURE ✓ — Terraform

Q&A

CONTACT

LINBIT HA-Solutions Gmbh
Vivenotgasse 48
1120 Vienna, Austria
sales@linbit.com

HashiCorp Germany GmbH
team-emea-dach@hashicorp.com