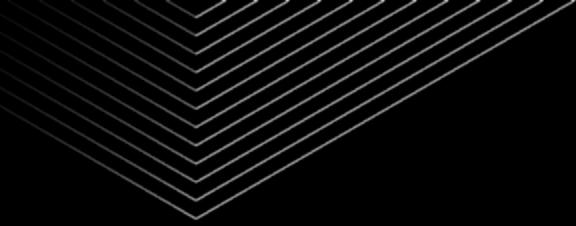




# Einführung in Nomad mit einem Turingpi



# Lance Haig

Regional Manager Solutions Engineering,  
HashiCorp



# Agenda

## Our Journey today

- Why Orchestration?
- Why Nomad
- Technical Overview
- Demonstration
- OpenSource and Enterprise



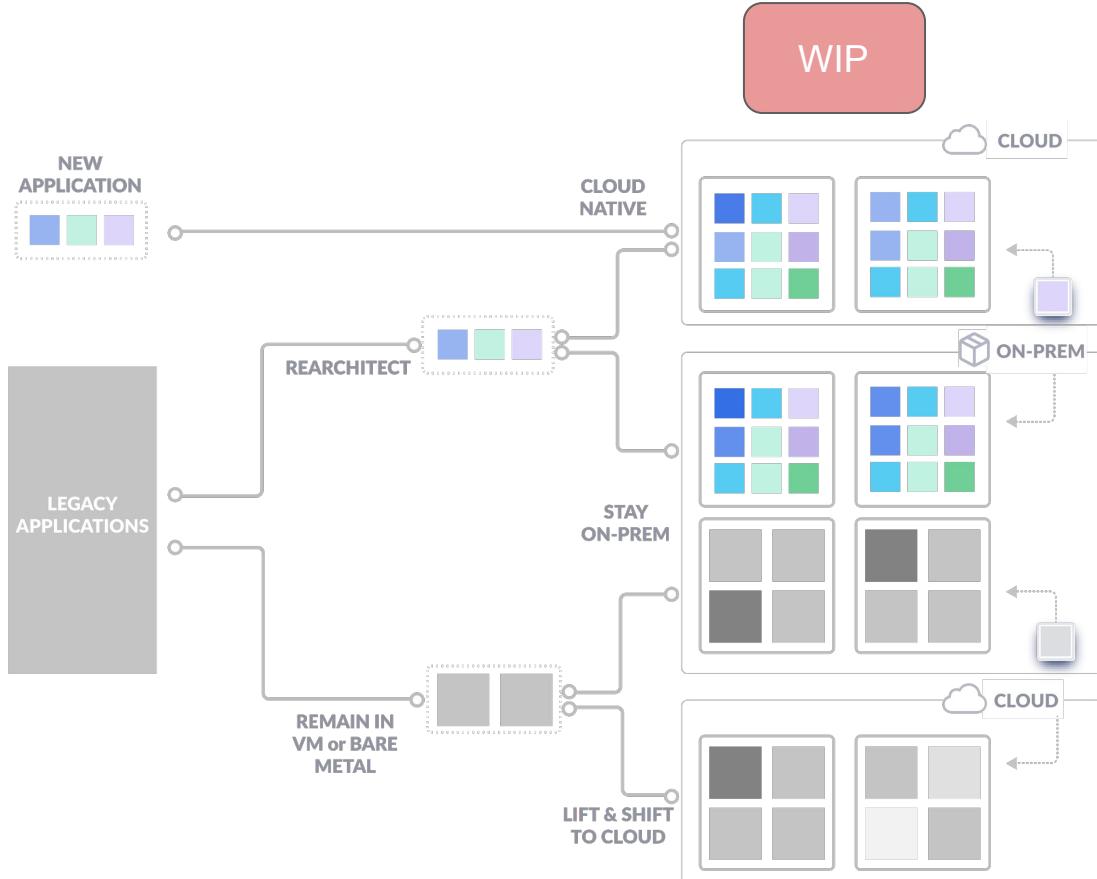
—

# Why Orchestration?



# 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

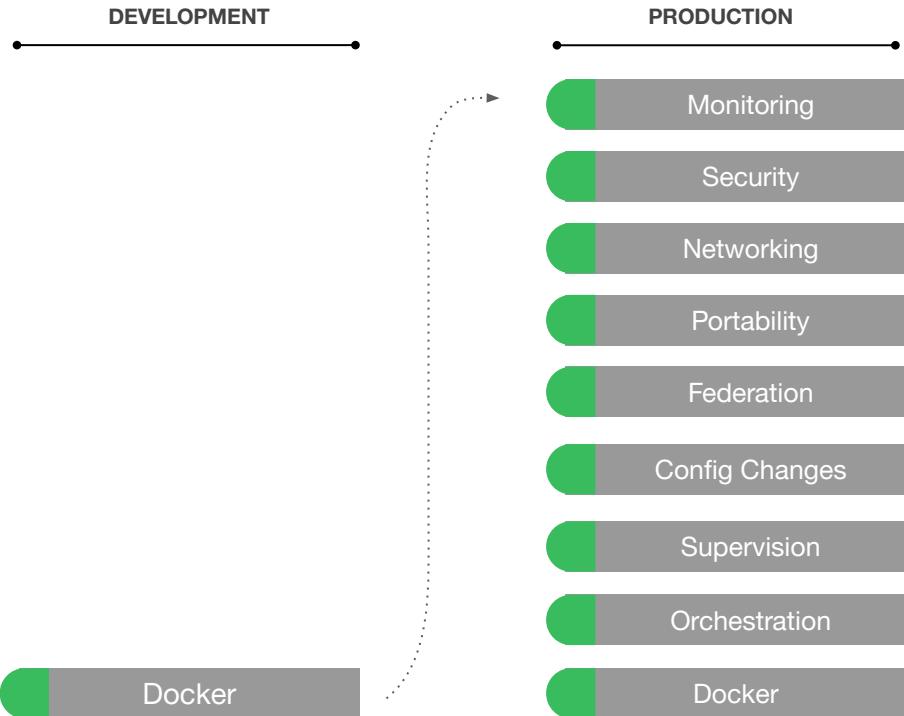




# Complexity is a top challenge in using and deploying containers

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

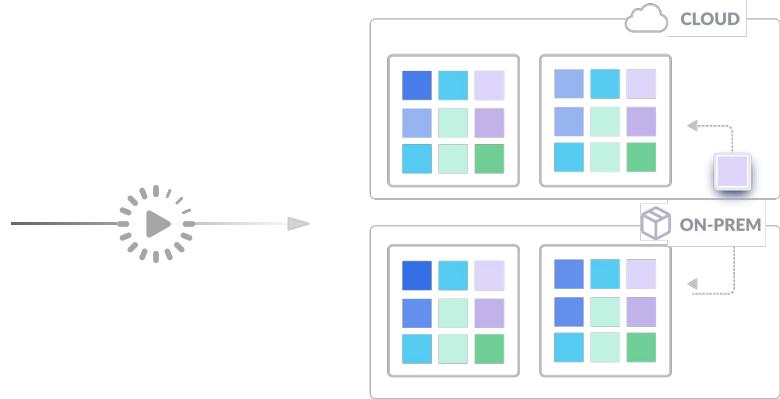
<https://www.cncf.io/blog/2018/08/29/cncf-survey-use-of-cloud-native-technologies-in-production-has-grown-over-200-percent/>





# 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





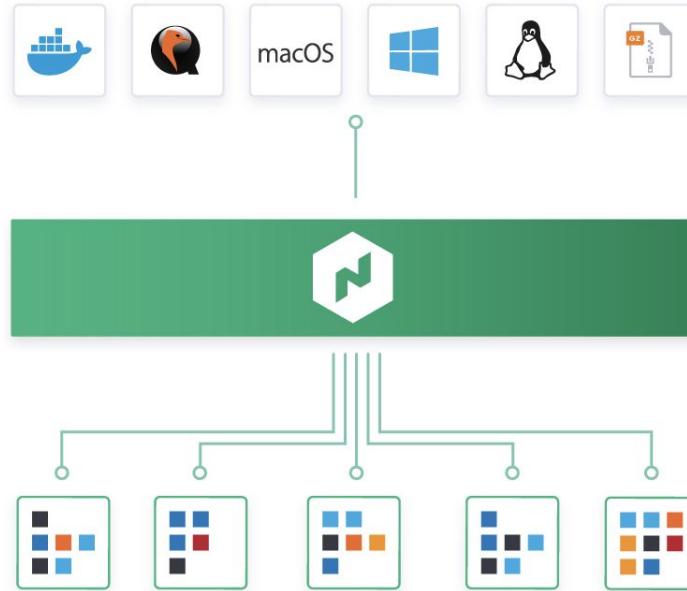
# Why Nomad?



# Guiding Principle

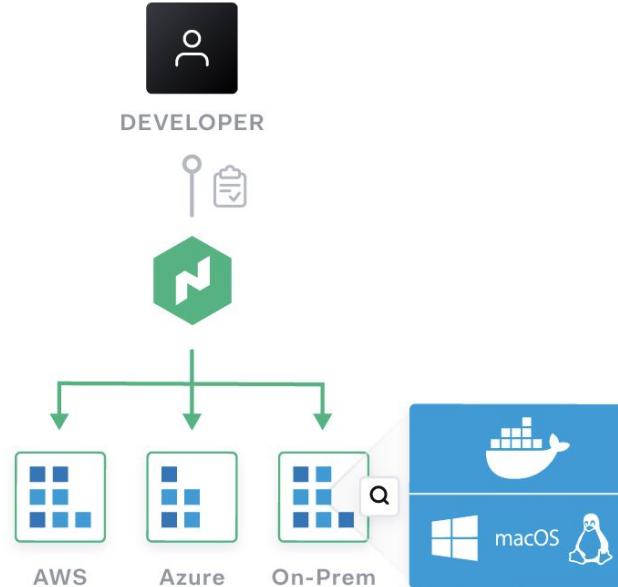
## Orchestrate Any Application

- 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



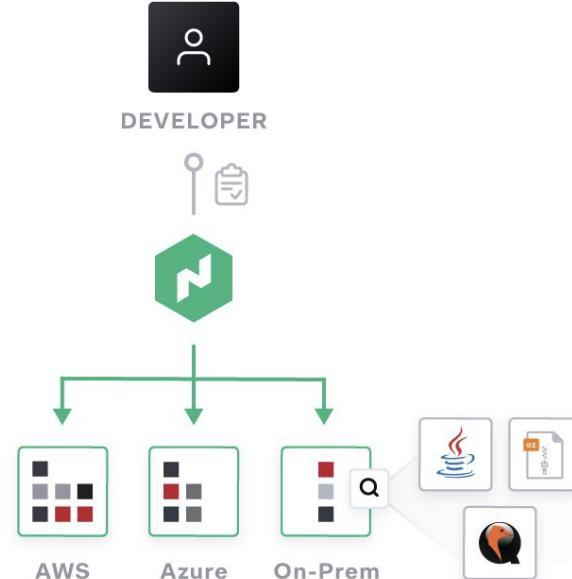
# Container Orchestration

- **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



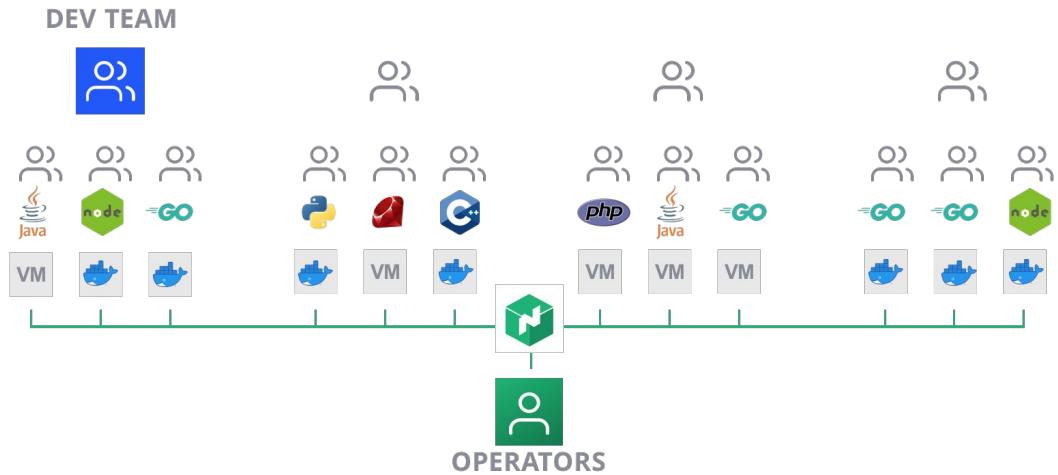
# Non-Containerized Application Orchestration

- **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



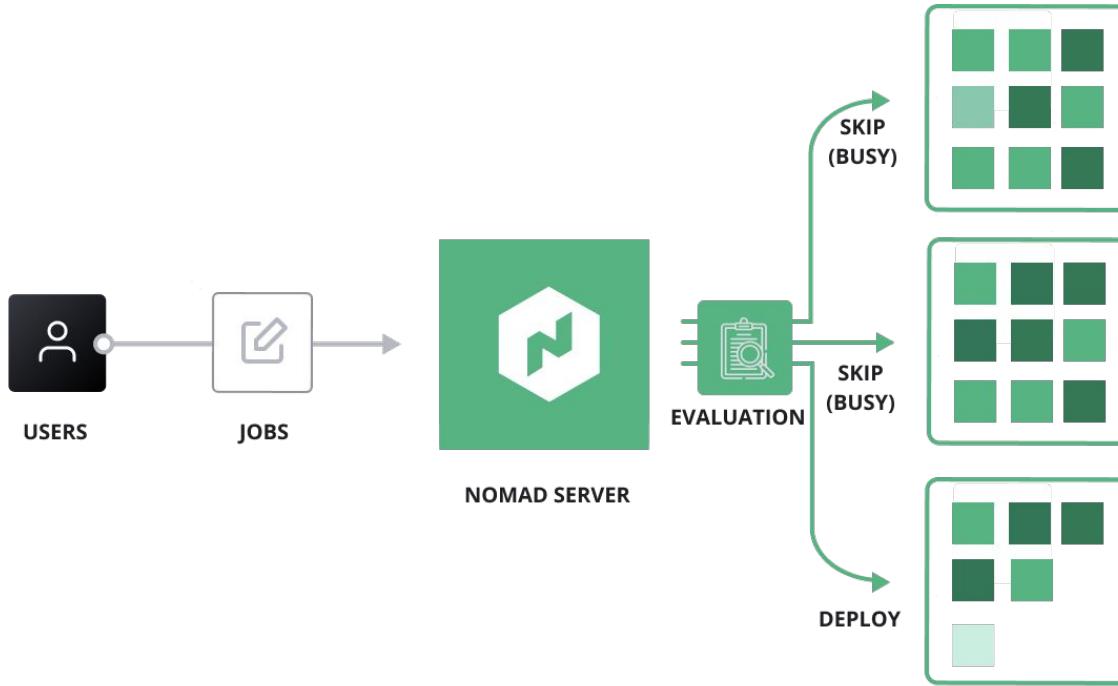
# Easy to deploy and maintain by system operators

- **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.



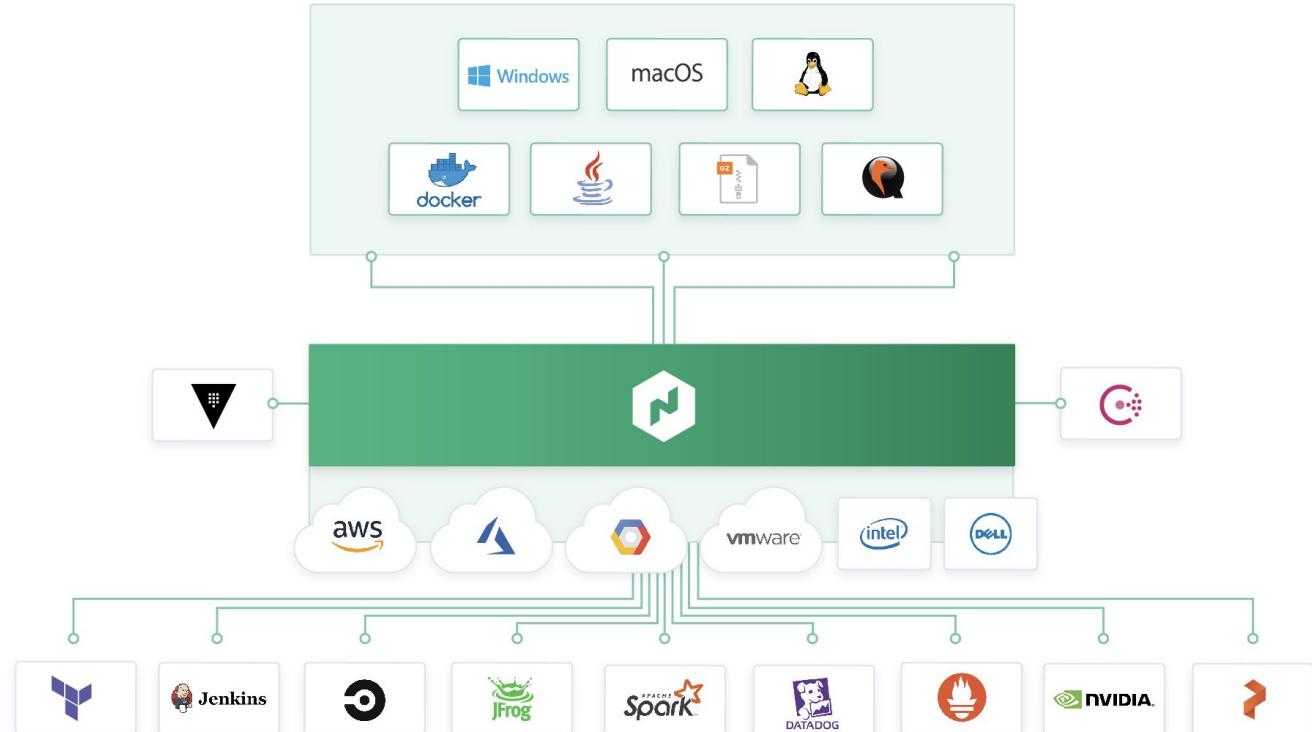


# Automated Deployment Workflow





# Broad Ecosystem Integration





# Public Users of Nomad



SAMSUNG

trivago

AUTODESK

pagerduty

pandora

ROBLOX

ebay

jet

Q2

circleci

deluxe



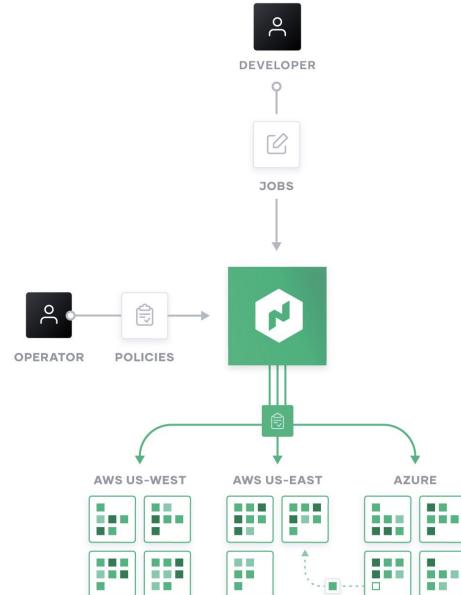
CLOUDFLARE

betterhelp

# 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



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

03

## 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.)

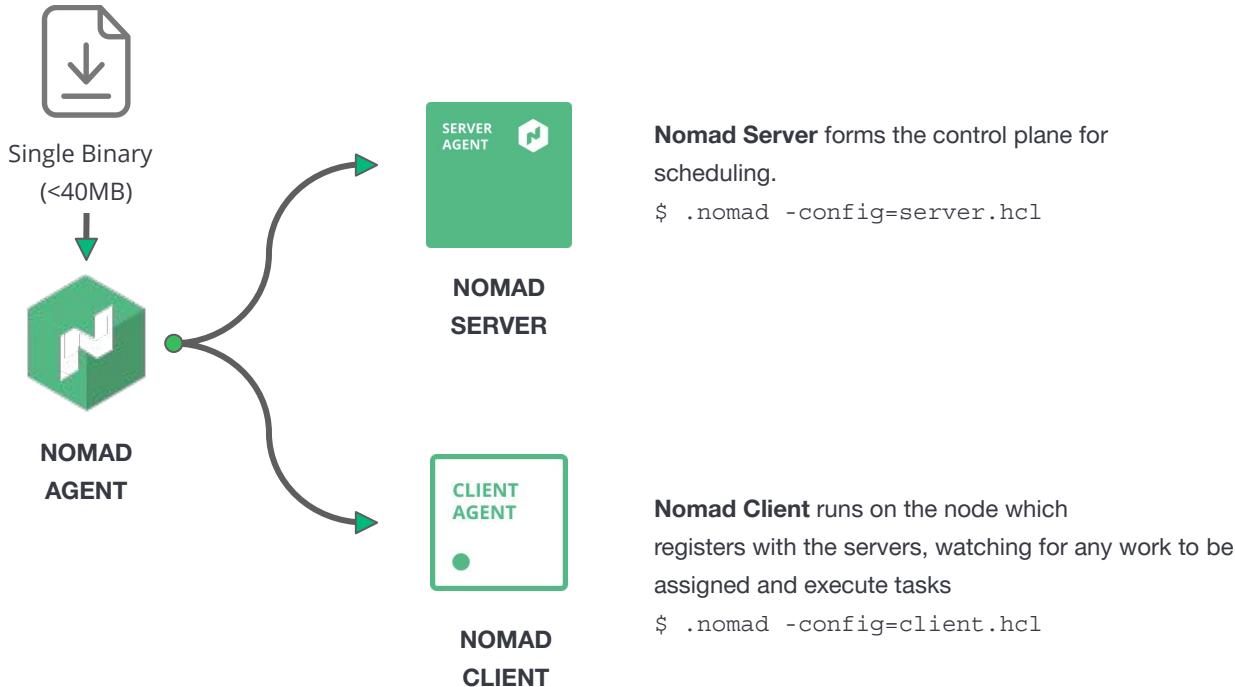


—

# Technical Overview

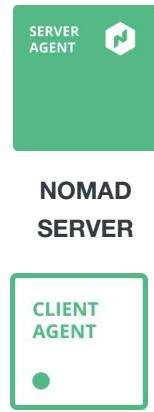
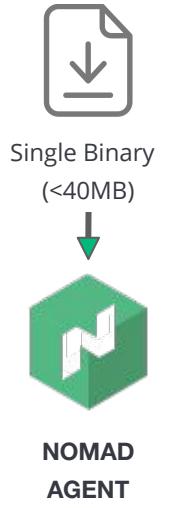


# Nomad Architecture - Single Process





# Nomad Architecture - Laptop

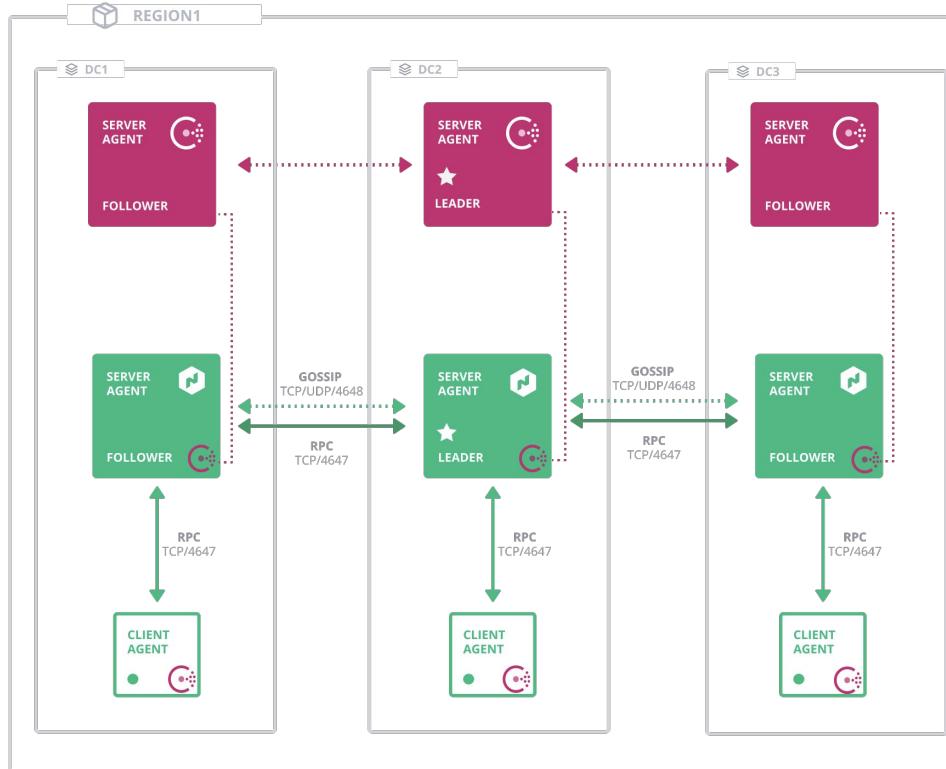


**Nomad Agent runs as a Server and a Client** on your local laptop.

```
$ nomad agent -dev
```

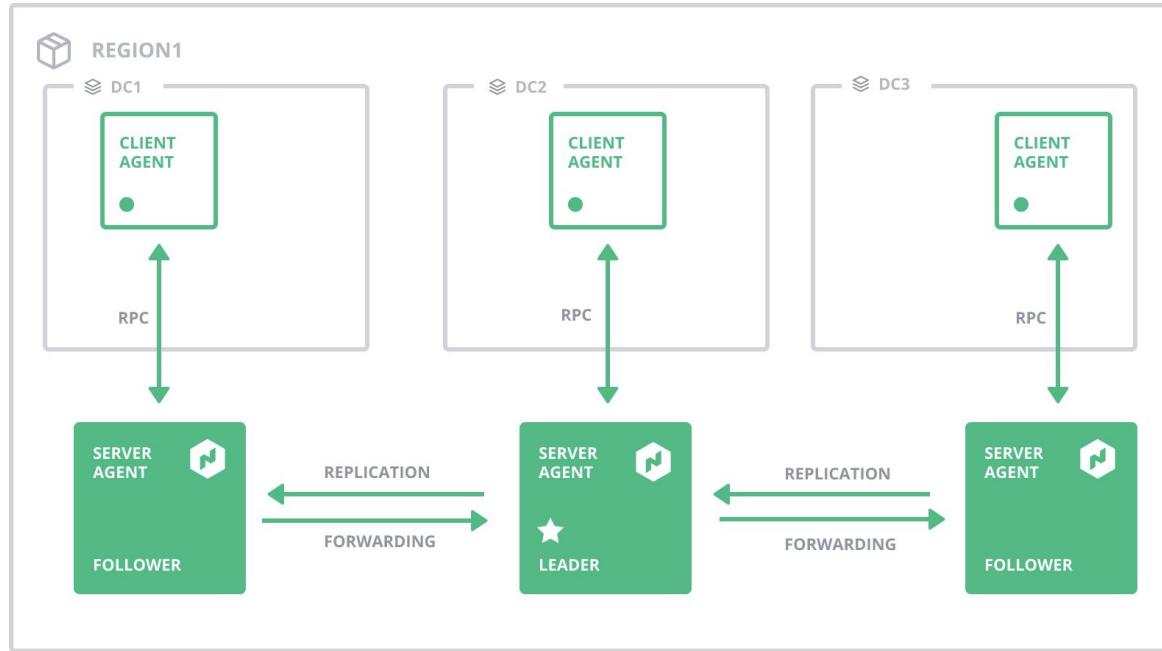


# Nomad Architecture - Single Region



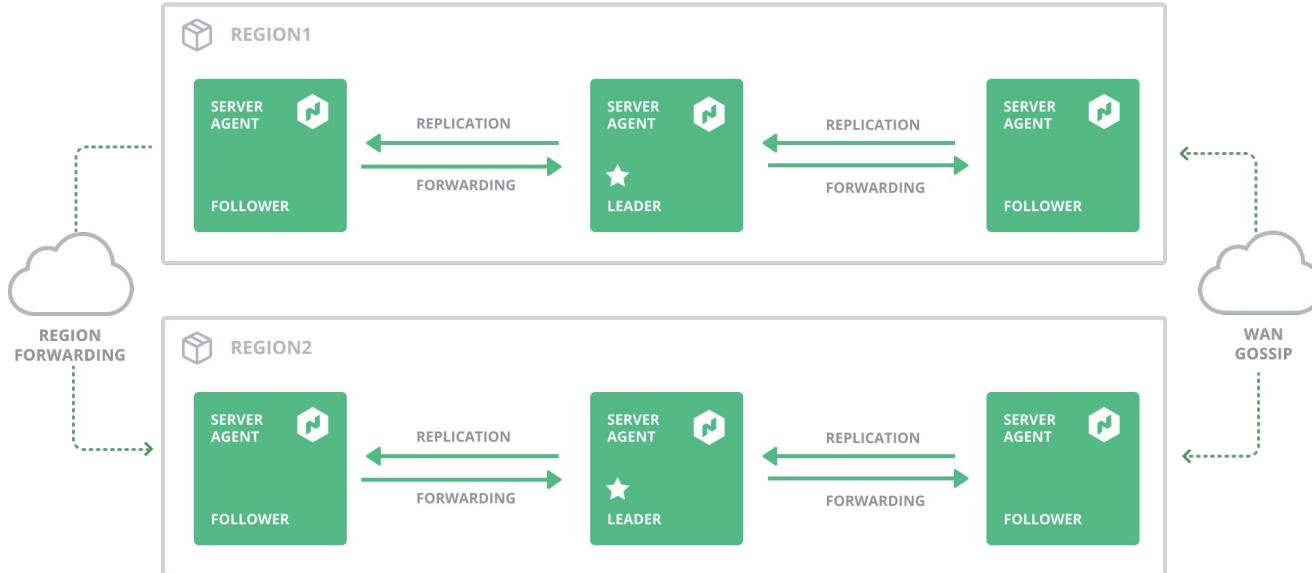


# Nomad Architecture - Single Region





# Nomad Architecture - Multi-Region





# Simple Cluster Formation



## Manual

Bootstrap a Nomad cluster manually without relying on any additional tooling.



## Automatic clustering with Consul

Nomad Servers and Clients can automatically discover each other by leveraging an existing Consul cluster



## Cloud auto-join

Automatic cluster joining using cloud metadata on AWS, Azure and GCP



# Demonstration



—

/

/



# OSS - Enterprise



## OSS

# Features

- Service & Batch Scheduling
- Task Drivers
- Device Plugins
- Multi-Upgrade Strategies
- Federation
- Autoscaling
- Container Storage Interface Plugin
- Container Network Interface Plugin
- Access Control System
- Web UI
- Consul Integration
- Vault Integration
- Namespaces



## Enterprise

- Automated Upgrades
- Automated Backup
- Enhanced Read Scalability
- Redundancy Zones
- Multi-Vault Namespaces
- Multi-Cluster Deployment
- Dynamic Application Sizing
- Resource Quotas
- Cross-Namespace Queries
- Audit Logging
- Sentinel Policies





—



# Thank you

—

[hello@hashicorp.com](mailto:hello@hashicorp.com)

[www.hashicorp.com](http://www.hashicorp.com)



---

## Links:

- Nomad Project <https://nomadproject.io>
- Learn Nomad <https://learn.hashicorp.com/nomad>
- Nomad Job Spec <https://www.nomadproject.io/docs/job-specification>
- TuringPi <https://turingpi.com/>



-

# BACKUP