
[PRD] Nomad Deployment Recipe

Summary: Nomad operators struggle with administrative visibility, ACL enforcement, security compliance, and greater traceability without a set of records that logs all operations taken by users in the cluster.

Created: March 1, 2021

Owner: rtan2@uw.edu

Contributors: rtan2@uw.edu
ktan5@uw.edu
zihengl@uw.edu

Status: **WIP** | In Review | Approved | Obsolete

RFC: [NMD-043: Nomad Audit Logging](#)

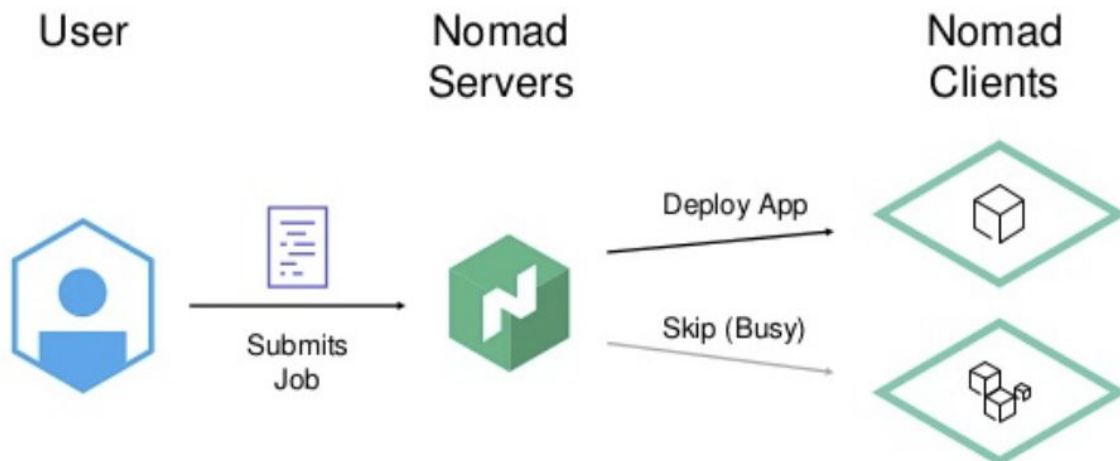
Background

Cloud deployment technologies are used to place workloads into a cloud computing environment. They provide the foundation to facilitate execution of applications and services. These technologies range from bare metal to serverless computing and enable organizations to realize the benefits of cloud computing. However, moving an existing application to that environment has its peculiarities. Pre-tests are required to ensure a migration, which comes out correctly and on time.

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. And Helm is the best way to find, share, and use software built for Kubernetes, which is a package and configuration manager that gives teams the tools they need to collaborate when creating, installing, and managing applications.

Inspired by Helm of K8s, we propose a project that brings this collaborative, single-command deployment experience to Nomad, HashiCorp's workload orchestration and scheduler. The objective would be to develop an open source command-line utility and API to support single-command application deployment and service catalogs (chart repositories) on Nomad, and help us simplify manual operations during deployment..

Nomad Workflow



Problem

This tool helps you manage nomad applications — It helps you define, install, and upgrade even the most complex nomad application. It is easy to create, version, share, and publish — so start using it and stop the copy-and-paste.

Impaired Developer Productivity

Developers can spend a lot of time deploying test environments to test code and replicate customer issues.

- With our tool, developers can focus on developing applications instead of deploying dev-test environments.
- Our Registry – such as WordPress and minio – allow developers to get a working configuration quickly for their application. Plus, developers can author their own registry, which automates deployment of their dev-test environment.

Steep Learning Curve

It can take someone new to nomad apps a long time to learn how to use it resulting in high lead times to deploy production-grade apps on Nomad.

- Our Registry provides “push button” deployment and deletion of apps, making adoption and development of Nomad apps easier for those with little container or microservices experience.

- Apps deployed from Our Registry can then be leveraged together to meet a business need, such as CI/CD or blogging platforms.

Deployment Complexity

Nomad applications can be complex to deploy. Developers can use incorrect inputs for configuration files or not have the expertise to rollout these apps from HCL/Nomad templates.

- Our Registries allow software vendors and developers to preconfigure their applications with sensible defaults. Charts also allow users/deployers to change parameters (e.g., resource limits for CPU and memory) of the application/chart using a consistent interface.
- Developers leveraging our Registries can incorporate production-ready packages while building applications in a Nomad environment. This can eliminate deployment errors due to incorrect configuration file entries or mangled deployment recipes.

Production Readiness

Deploying and maintaining Nomad applications can be tedious and error prone.

Avoiding Duplication of Efforts

By sharing our registries within an organization or across organizations, duplicate efforts are avoided leading to higher efficiency and reduced errors.

Personas(based on user research) (TBD)

- **Affected Persona 1** has troubles developing with this problem
- **Affected Persona 2** has troubles operating with this problem
- **Affected Persona 3** has troubles securing with this problem

Requirements and Phases

	Requirements
Phase 1: Nomad practitioners install the required “package” to local file (potentially on running nomad cluster)	Required packages can be downloaded from a git repo
	Validate downloaded file authenticity
	“install” the package to local file/Nomad cluster (have to define meanings of “package” and “install”)
Phase 2: Support CSI plugins	Hide the manual setups when a “package” requires persistent storage volume to be set up
Phase 3: Validation, Documentation, and Release	Initial code drop
	Demo for reference application
	README; Announcement for tool’s features

Nomad practitioners install the required “package” to local file (potentially on running nomad cluster)

Nomad practitioners in this case stand for people who have experience in pulling docker images and deploying their own applications on docker containers and who have bare knowledge of writing docker/nomad files.

Considerations

- downloaded files have to be validate through comparing the checksum value
- downloaded files need to store in a cache file to avoid duplicated download

Acceptance Criteria

- commands like “install”, “uninstall”, “info” should be implemented
- “install”: pull the package from given URLs and install locally
- “uninstall”: remove the downloaded package and its dependencies
- “info”: show info of the downloaded files

User Research (TBD)