TOOL CONSOLIDATION

Create a traceable path to production with a unified approach to container registries and deployment technologies

BACKGROUND

Container registries and technologies like Terraform and Ansible play a critical role in organizations deploying containerized applications to dynamic runtime environments. As organizations scale there are multiple considerations in deciding what container registry to use, where it is hosted, and how to manage infrastructure as code files.

If deploying to public clouds such as AWS, GCP, or Azure, it's oftentimes compelling to make use of their provided container registries (e.g.,ECR) for production-ready assets. This is because authentication can be easier when storing the containers in the cloud that they're deploying it on. Organizations will also feel compelled to adopt solutions like Ansible Tower and Terraform Cloud to allow for more reliable management and deployment of their infrastructure files.

Because of this approach organizations architecture themselves such that mid-level artifacts are in one repository management solution, but production assets are in a separate registry. This creates a situation where there's a gap in audit capability and impossible-to track issues between the development and deployment stages of the software lifecycle. The bifurcation of production and development assets also calls for maintaining multiple solutions, which involves managing:
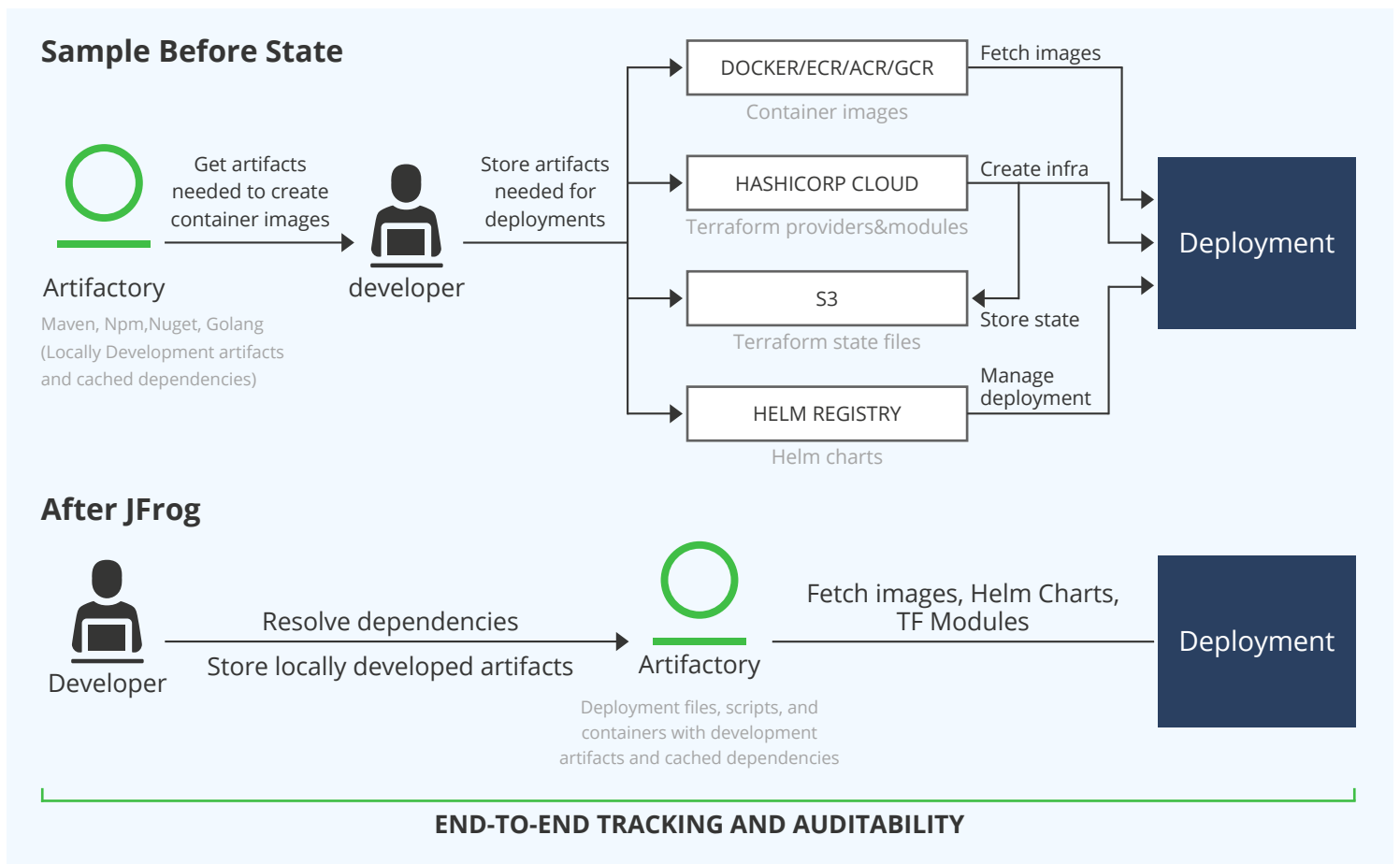• Authentication and access
• What assets are deployed where
• Who has performed various actions against a given asset
• Multiple infrastructure products with their own operational overhead

SOLUTION

JFrog allows organizations to maintain full visibility and control of applications from development through deployment by serving as their fully featured container and Kubernetes registry, along with being a registry for every other package used in development. Incorporating your IaC and config files with containers and build metadata in JFrog provides the most direct traceable path of what is being delivered into production.

Key capabilities supporting deployment tool consolidation with JFrog:
• OCI compliant container registry (Docker & OCI containers)
• Universal package support including Helm Charts
• Full featured Terraform Module, Provider, and Backend file management
• Automated deployment workflows for Terraform and Ansible
• Extensive build, user, and custom metadata in one place
• Self-hosted, hybrid, and multi-cloud support
• Integrated holistic security solution
• Robust RBACs

Sample Before State

Artifactory

Maven, Npm, Nuget, Golang
(Locally Development artifacts
and cached dependencies)

Get artifacts needed to create container images →

developer

Store artifacts needed for deployments →

DOCKER/ECR/ACR/GCR
Container images

Fetch images →

HASHICORP CLOUD
Terraform providers&modules

Create infra →

S3
Terraform state files

Store state

HELM REGISTRY
Helm charts

Manage deployment →

Deployment

After JFrog

Developer

Resolve dependencies →
Store locally developed artifacts →

Artifactory

Deployment files, scripts, and containers with development artifacts and cached dependencies

Fetch images, Helm Charts, TF Modules →

Deployment

END-TO-END TRACKING AND AUDITABILITY

With JFrog, every artifact and every build is managed and secured in one system. This allows organizations to be certain of every action taken against a component for the ultimate knowledge of what's running in production with robust auditing ability. Security gatekeeping is simplified as they only need to apply efforts on a single system.

RESULTS

Root Cause Analysis

Easily identify what's changed between production builds and config files

Cost Savings

Thousands of dollars across license, maintenance, and integration costs

Quality Control

Confidently attest to the quality of software released to production

Greater Visibility

Capture all build info, custom and user generated metadata

We've been using JFrog Artifactory on AWS at Monster.com, and it has been a winning combination for us. After evaluating our options, we went with Artifactory because of its breadth of coverage – we were able to avoid the operational challenges of using multiple tools. Artifactory on EKS Anywhere is an exciting next step from the two vendors because it combines a top-notch Kubernetes distribution with a top-notch DevOps solution."

Martin Eggenberger, chief architect, Monster.com

ABOUT JFROG

JFrog empowers thousands of DevOps organizations globally to build, secure, distribute, and connect any software artifact to any environment using the universal, hybrid, multi-cloud JFrog Platform.

Start for free at https://jfrog.com/register/

LEGAL STATEMENT

www.jfrog.com

www.twitter.com/jfrog

www.facebook.com/artifrog/

www.linkedin.com/company/jfrog-ltd