



Distributing Releases at Scale

Roy Ben Shoushan, Solutions Engineering Manager

Training Team



Carlos Moya Moradas
Senior Solutions Engineer



Roy Ben Shoushan
Solution Engineering Manager

Agenda

- Introduction to the JFrog Platform
- Software Distribution
- Distribution Use Cases
- Distribution Patterns
 - Classic Distribution
 - Hybrid Distribution
 - Download Center
 - Content Delivery Network (CDN)
 - Private Delivery Network (PDN)
 - Air Gap Distribution
- Quiz, [Raffle](#), Feedback, Resources

CLASS LOGISTICS

- Coffee Break
- Course Materials
- Hands-on labs on your computer
- Class survey
- Quiz

CLASS PREREQUISITES

- Since a significant part of the training involves hands-on exercises, a laptop with the following configuration is required:
 - WiFi Connection
 - A browser with access to the internet.
 - An appropriate terminal client needs to be installed on the machine, for example, terminal or iTerm2 on MacOS.

LAB SETUP INFORMATION

- All the labs are in the below github page
 - <https://github.com/jfrog/SwampUp2022/tree/main/sup002-distribution>

JFROG AT A GLANCE



1,200+
EMPLOYEES



12
OFFICES WORLDWIDE



~7,000
CUSTOMERS
GLOBALLY



>300
COMMUNITY
EVENTS ANNUALLY



\$207M
ANNUAL
REVENUE



THE JFROG APPROACH

END-TO-END DEVOPS PLATFORM

Unify, Accelerate & Secure Your Software Delivery,
From Development To Distribution



ULTIMATE SCALABILITY

Flexible, Open And Expandable By Design, That
Grows With Your Scaling Needs



UNIVERSAL TECHNOLOGY

Supports 30+ Package Types and Any Tool in
Your DevOps Ecosystem



CONTINUOUS SECURITY

Integrated Security Across Your Pipeline to
Achieve Security and Compliance at DevOps
Speed



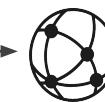
CLOUD, MULTI-CLOUD & HYBRID

Deployment Flexibility Supports Any
Development Topology Including Cloud Native



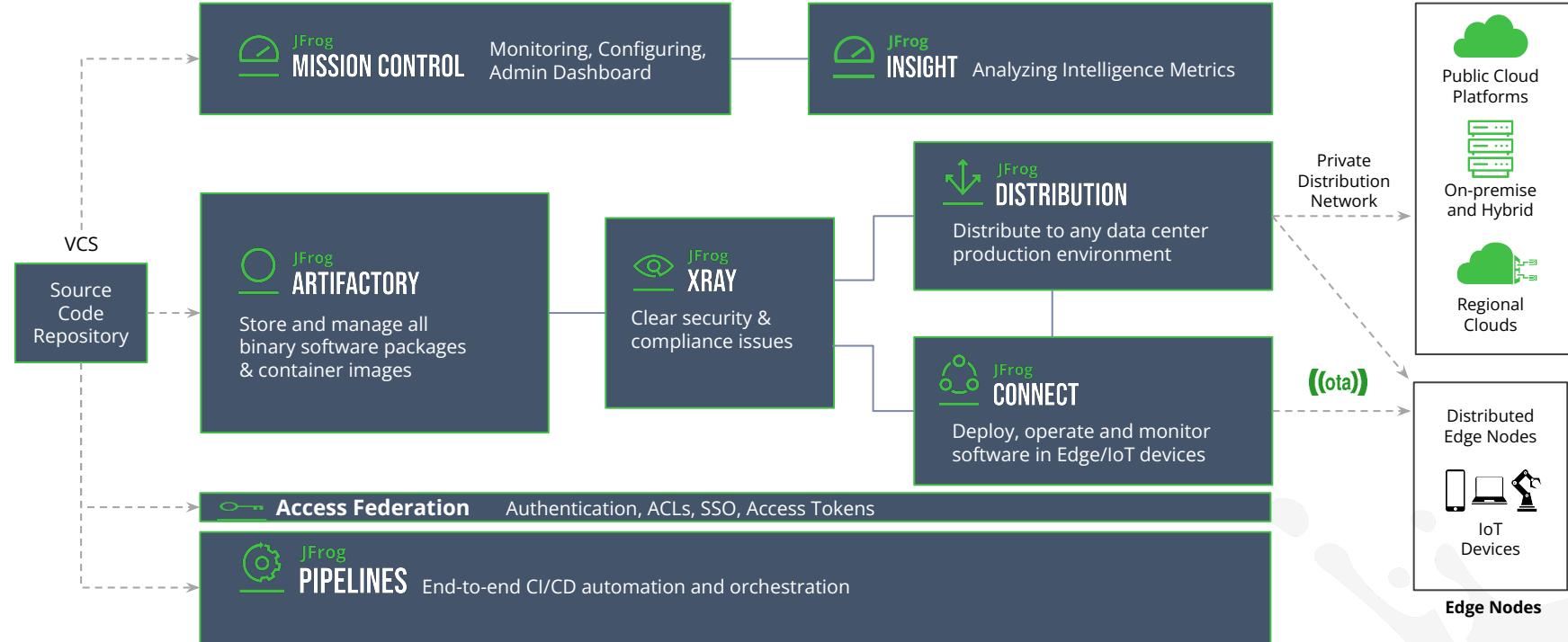
INTEGRATED ECOSYSTEM

Extensive Technology, ISV and Cloud Partner
Integrations, Out-of-the-Box



THE JFROG DEVOPS PLATFORM

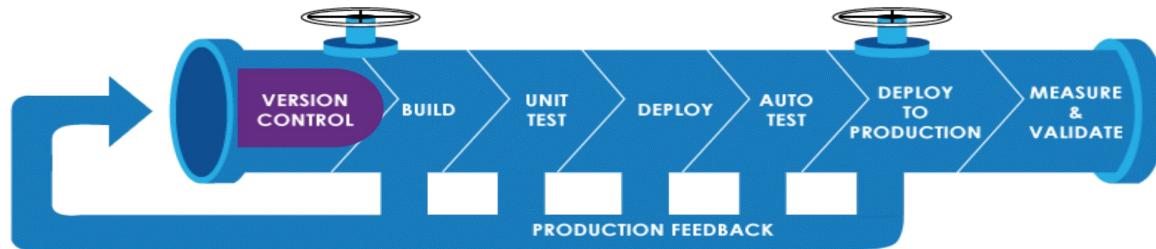
Unify, accelerate & secure your software delivery, from development to distribution



Demo: JFrog Platform (UI)

Continuous Pipeline

A traceable, approved and secure process that is driving the artifacts' transition across lifecycle stages



- A Continuous Pipeline is an automated representation of the **value stream** of an organisation
- The Product Increment (Component, Package, Software Bundle, etc.) is transitioned (via the pipeline) to different life cycle stages based on policies

What is CD

Continuous Delivery

- Build software in such a way that the software can be released to the production at any time
- Automation of steps to securely get changes into production

Continuous Deployment

- Focuses on the actual deployment, Continuous Delivery focuses on the release and release strategy
- Every change that you make, goes through the pipeline, and if it passes all the tests, it automatically gets deployed into production



Software Distribution



The Liquid Software Company

What Is Software Distribution

Software distribution is a **critical part of your delivery/deployment process.**

Once applications are developed, **software binaries/artifacts** and **Software Bill of Materials (SBOM)** need to be **securely and reliably distributed** to the **next stage** in your **SDLC**.

For example:

- Testing/integration stages in your pipeline
- Deployment/release stages- for last-mile for production

Distribution speed impacts Delivery speed.



Modern Delivery Trends Impacting Distribution



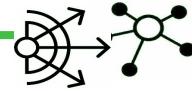
Over
520M
new updates
to apps/services
by
2024

X



>60%
More
frequently
daily or faster
by
2025

X



Across
huge, complex,
hybrid datacenters
Edge by 2023
footprint
7M Edges
thick-to-thin
38B 'Things'

X



80%
Weightier
edge
Artifacts
applications
are K8s

THESE MULTIPLIERS AT THIS SCALE ARE NEW

Distribution Bottlenecks



DEV TEAMS



OPERATIONS

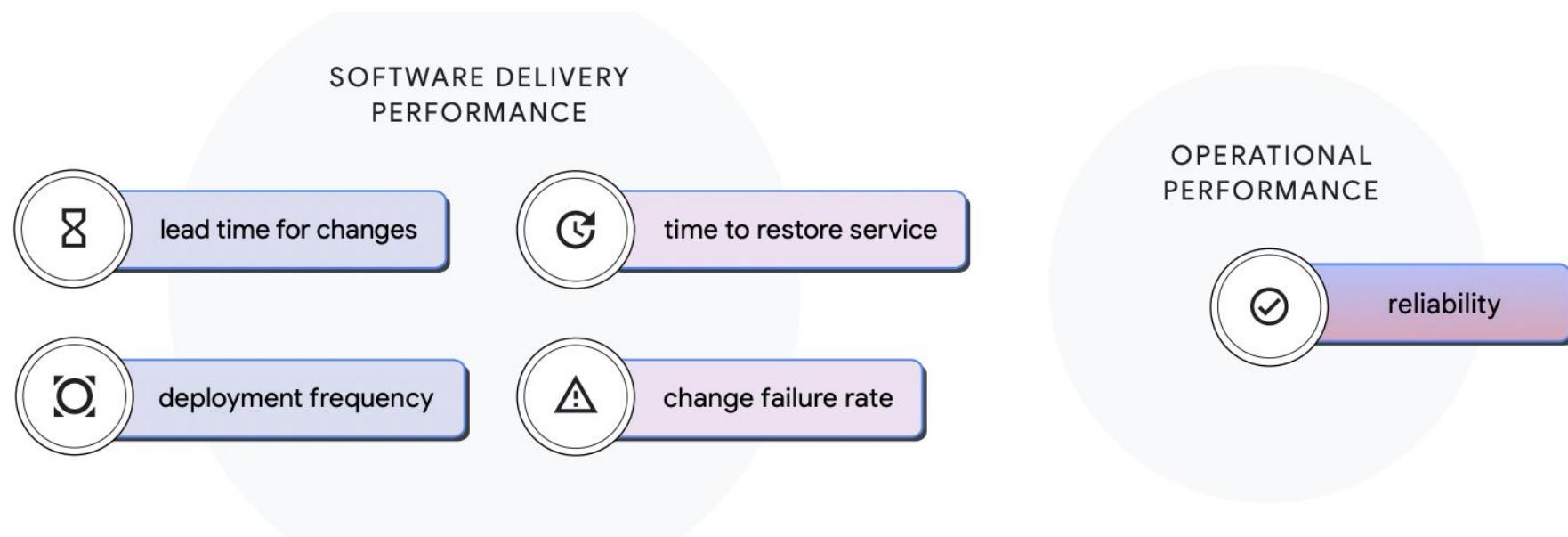


RUNTIME ENV.



END USERS

Metrics of Delivery Performance



Source: [The Accelerate State of DevOps Report](#)

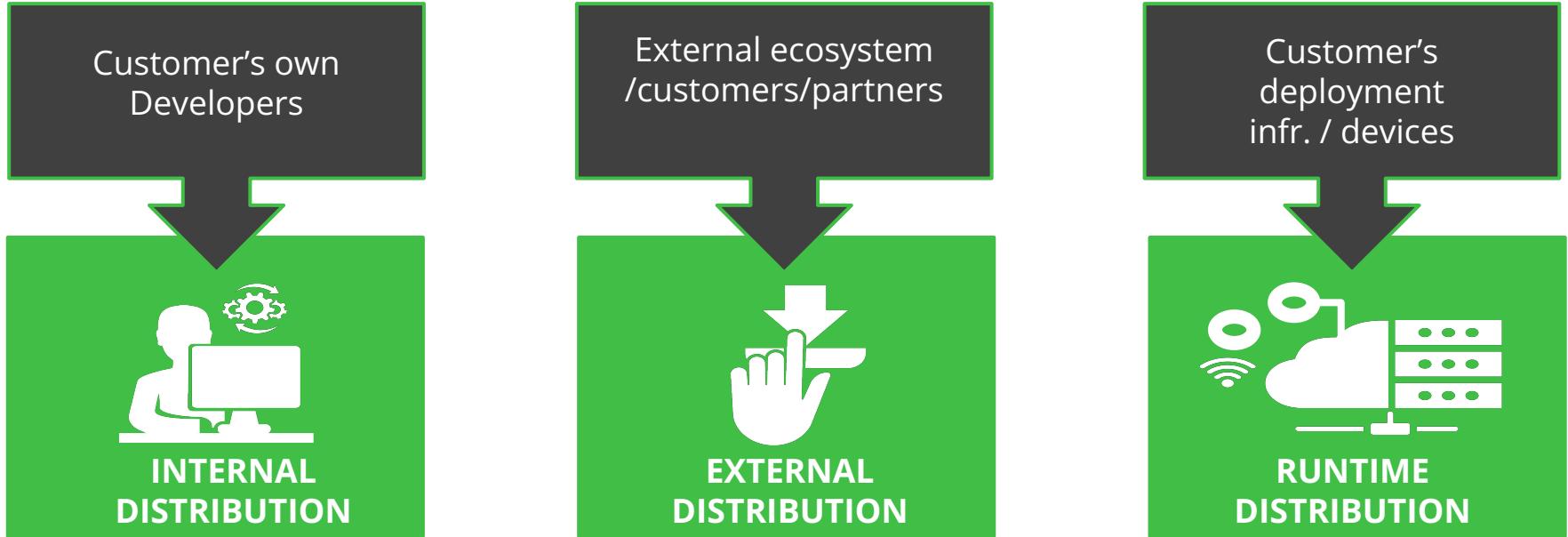


Distribution - Use Cases



The Liquid Software Company

Who is consuming the Distributed Software?



Internal Distribution

Customer shares a set of binaries with remote developers/offices in the org for continued development/next stage in the pipeline. Needs a **fast, secure, governed** solution.

1. **Remote offices & GEOs**
2. **Getting past the Restricted Networks**
3. **Separation of duties** requirements when handing off between Dev to the next stage in Ops.

External Distribution

Customer shares packages with users outside of the organization in a “**download center**” - can be open to the entire ecosystem, or with certain restrictions

1. **Developer ecosystem:** SDK/ plugins/ drivers/ OSS / Freemium/ base images / toolkits / utilities (i.e. catalog)
2. **Specific partners or customers:** (Authenticated)
3. **Repo-to-repo** in complex supply chain

Runtime & Infrastructure EDGE Distribution

Customer delivers a set of binaries to a computing infrastructure for runtime deployment. The binaries need to be transferred as close as possible to the machines/last-mile in order to **accelerate deployment sequence**.

1. **Distribution to production nodes/ 'Thick' Edges (hybrid)**
2. **IoT Devices** Binaries into non-server grade gateway/device
3. **Kubernetes** cluster deployments



Distribution Solutions

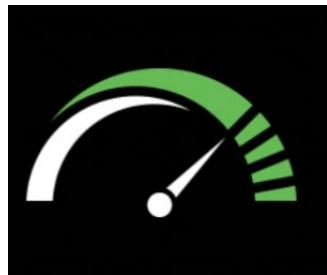


The Liquid Software Company

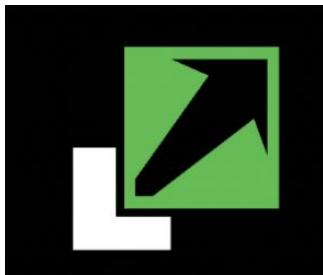
Possible Solutions

- Cloud Storage Hosted Solution
- CDN
- Download Centers
- Replication
- Federated Repositories
- JFrog Distribution

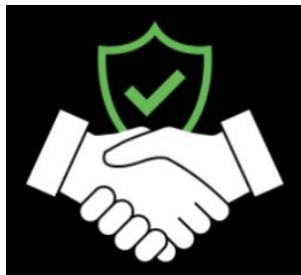
A Flexible Distribution Mechanism



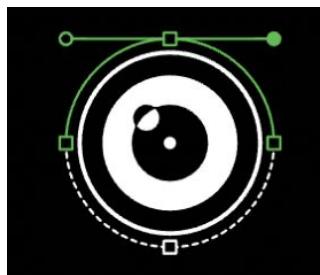
SPEED



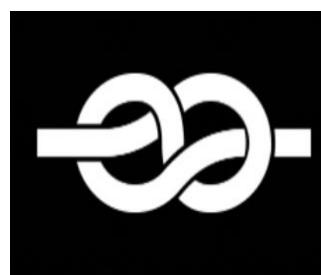
SCALE



VISIBILITY



TRUST



SIMPLICITY
through
Flexibility

TIGHTLY INTEGRATED ACROSS YOUR E2E DEVOPS DELIVERY

- ✓ Pipeline Automation
- ✓ Security & Compliance
- ✓ Central Management of Enterprise DevOps



Terminologies

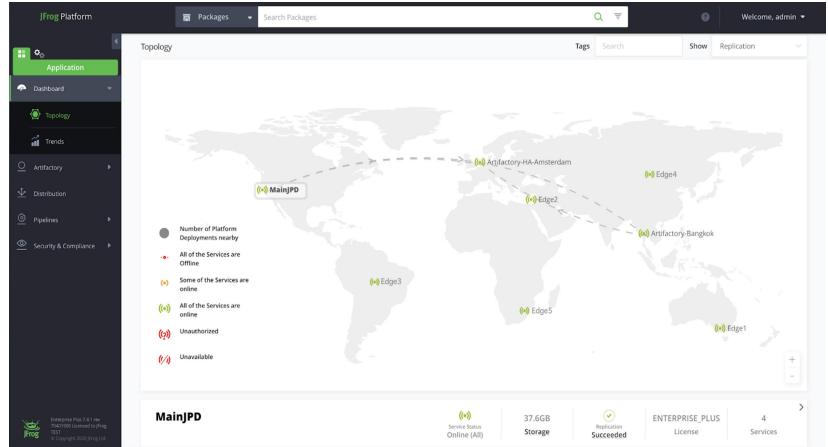


The Liquid Software Company

Distribution Edges

Lightweight READ ONLY version of Artifactory

- Read-only, secured (GPG-signed) distribution to remote locations
- Smart Remote Repositories - read-only OOTB repository mirroring
- Direct upload of software to an Edge node is not possible
- Getting past the Restricted Networks
- Build for scale: >500GB Release Bundles, >30K files in a BOM, >50GB large files, and 1000's of Edge Nodes



Release Bundle

A grouping of contents that are part of your **release**, providing the **bill of materials** for your software releases.

- **Secure and Immutable**, ensuring no manipulation by unauthorized users
- Verified distribution: Secure data in transit with verification at consumption point
- Fine-grained Role-Based Access Control (RBAC) - for publishing, managing and consuming binaries, across all internal/external targets



Distribution Patterns

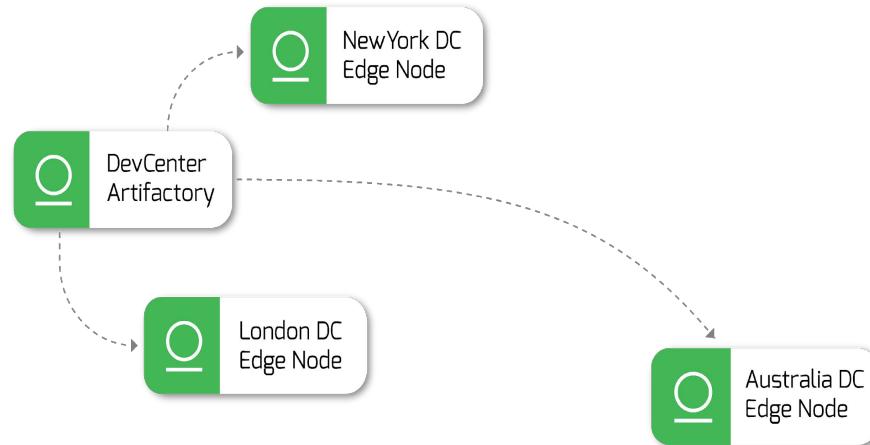


The Liquid Software Company

Classic Distribution

DISTRIBUTE TO YOUR DATACENTERS

- Provision software in private/public clouds
- Global distribution
- Can span multiple clouds
- Overcome high latency networks, low bandwidth





Distribution Flow

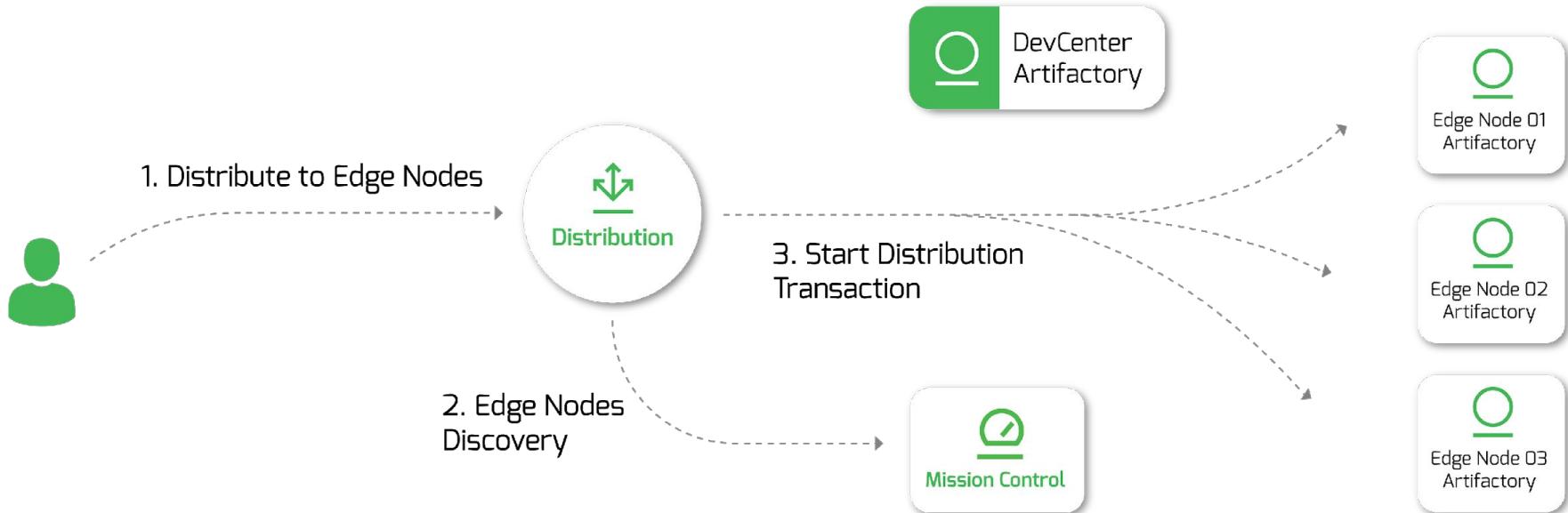


The Liquid Software Company

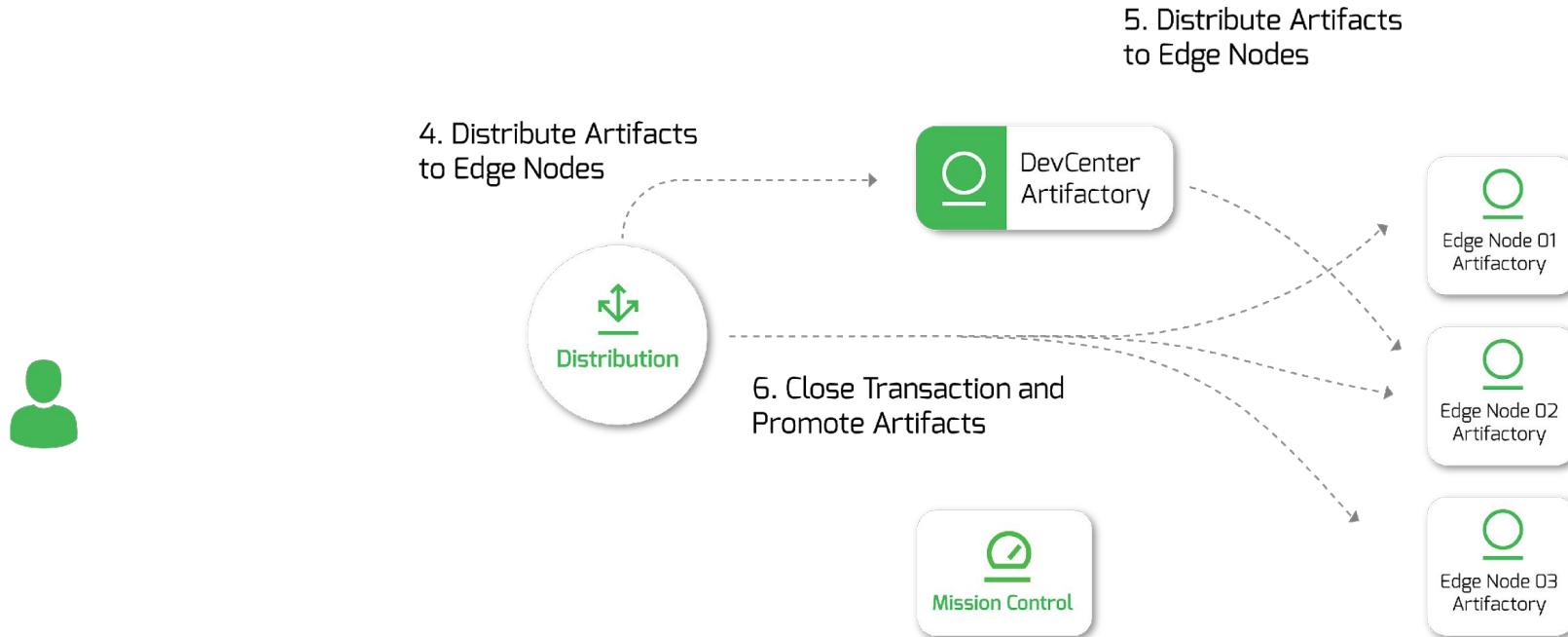
STEP 1 : CREATE RELEASE BUNDLE



STEP 2 : DISTRIBUTE RELEASE BUNDLE



STEP 3 : COMPLETE THE TRANSACTION



Demo: Classic Distribution (UI)

Lab 1: Classic Distribution (CLI)

Hybrid Distribution

Distribute from JFrog SaaS to Self Hosted

- Supports distributing of Release Bundles from SaaS JPD to Self-hosted Edge Nodes
- Balancing your distribution workloads in response to changing workloads, new challenges, and increasing security requirements.
- Distributing sensitive, highly regulated, and mission critical Release Bundles to Artifactory On-Prem Edges while using the JFrog Platform on the cloud for mainstream public distributions and thereby gaining significant cost savings.

Download Center

Distribute from JFrog Self-Hosted to SaaS

- Supports distributing of Release Bundles from Self-Hosted JPD to SaaS Edge Nodes
- Minimal Maintenance of Edge nodes
- Ease of maintenance
- Expand beyond supported regions
- Artifacts downloadable by External Entities : Partners, Vendors, Consumers



Dynamic Distribution



The Liquid Software Company

Dynamic Distribution

Single API call for the Distribution process

- The background flow is similar to the classic distribution process.
- This process is simpler
- You can distribute a dynamic Release Bundle, which is a Release Bundle you can create, sign, and distribute on the fly using the [Dynamic Release Bundle REST API](#).
- The Release Bundle version is created instantly and distributed to the selected Distribution Nodes.
- The distribution process for a Dynamic Release Bundle is the same as a regular Release Bundle.

Lab 2: Dynamic Distribution

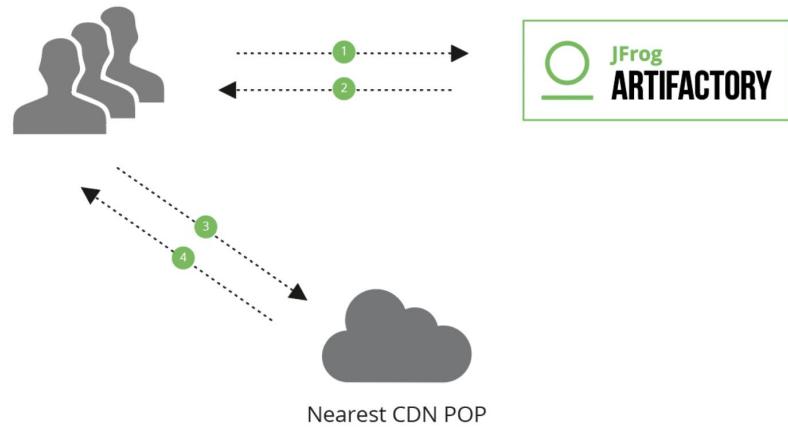
Content Distribution Network (CDN)

Simple management of distribution at scale

- Enable by repository
- Full RBAC
- Geo restrictions
- Signed URLs for non-authenticated downloads

CDN Flow

1. The user submits a download artifact request to Artifactory.
2. Artifactory sends back a redirected CDN URL to the user.
3. A redirected download request is automatically routed by the user to the nearest CDN.
4. The requested artifact is downloaded to the user.



Lab 3: CDN

Private Distribution Network (PDN)

Lightweight fast, secure, scalable and reliable network of nodes that act as an intermediate layer between Artifactory/Distribution Edge and the downloading clients

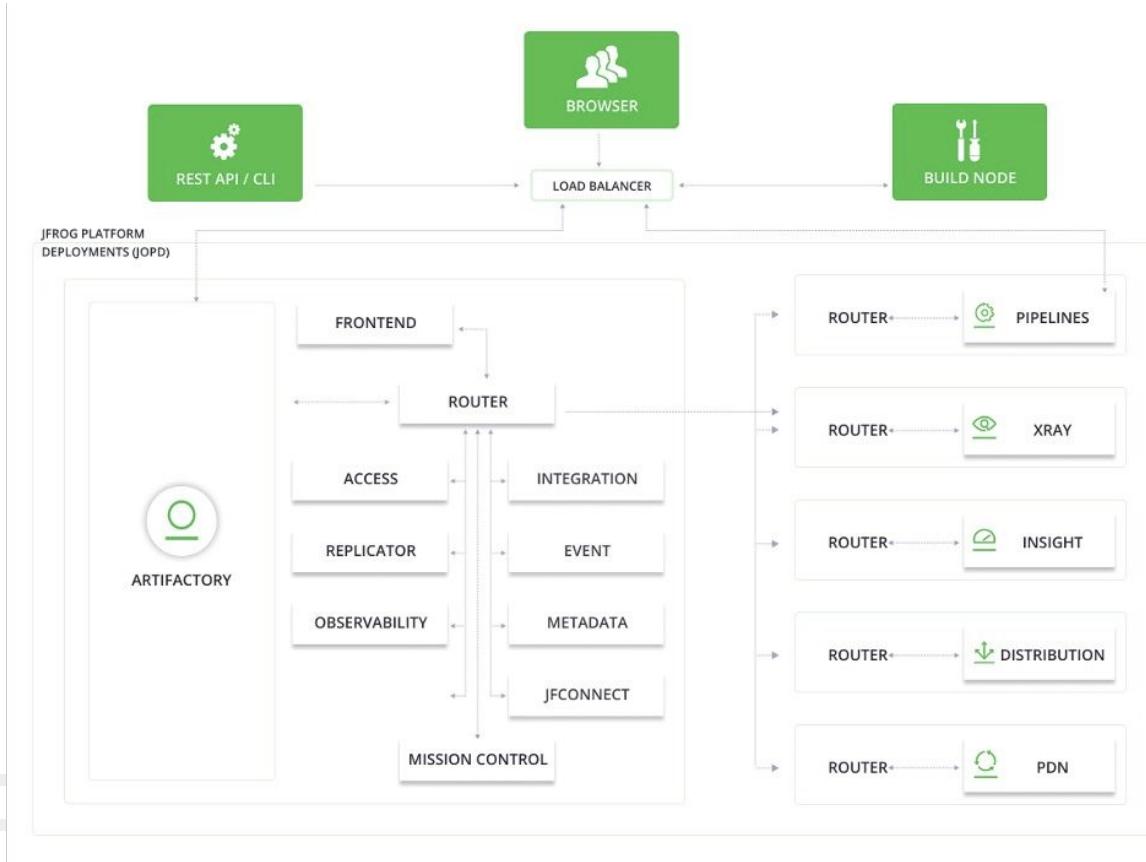
PDN Nodes: Lightweight instances which serve artifacts to clients.

- There may be thousands of nodes in a single PDN deployment, each node serving several clients
- Nodes can be grouped and have hierarchy

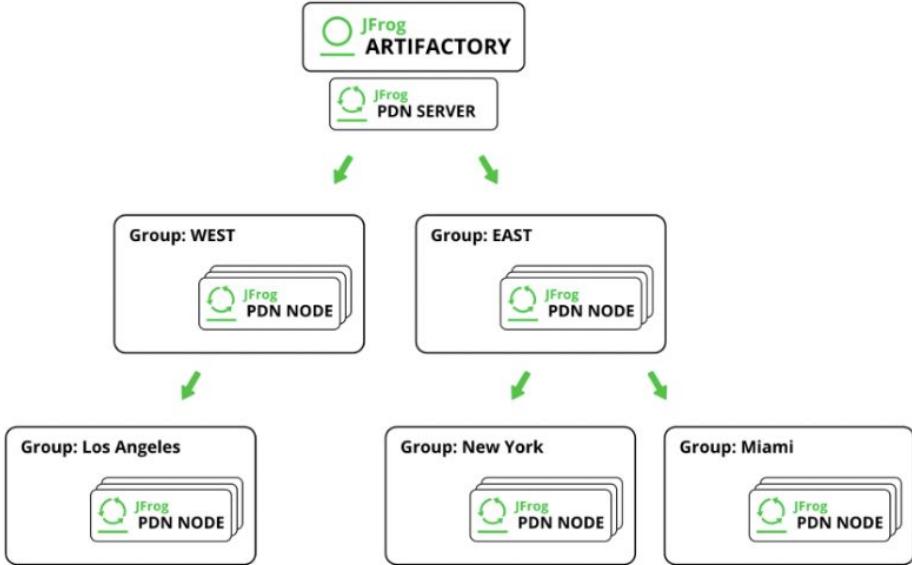
PDN Server: A single instance, responsible for managing and overseeing PDN Nodes.

- New service in the JFrog Platform Deployment (that can be deployed in HA)
- Responsible for registering PDN as a service on the JPD, and the "brain" of any

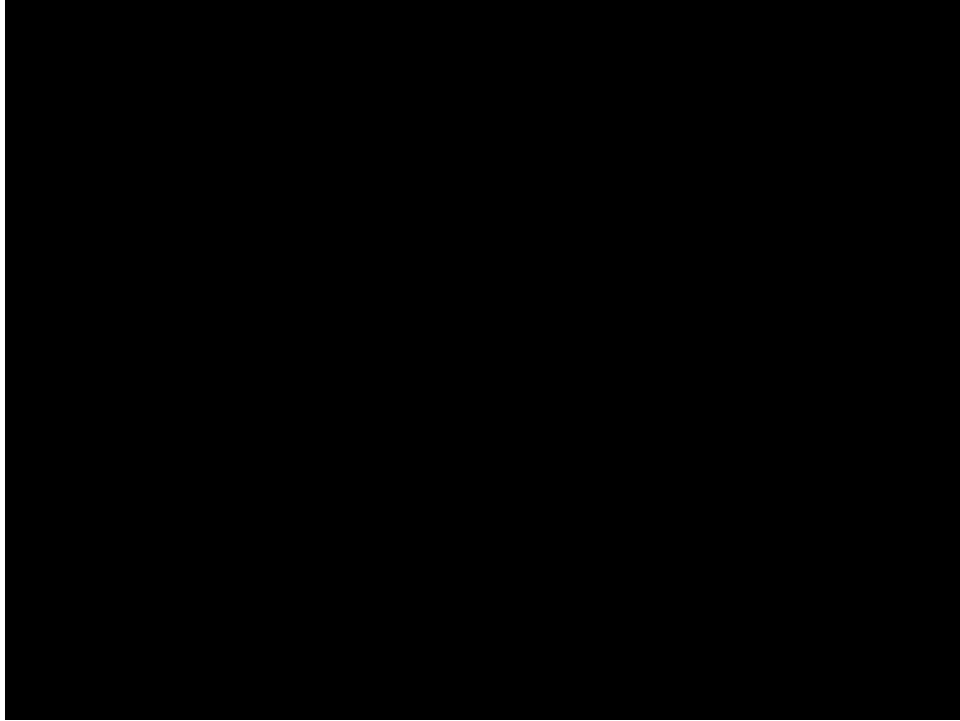
PDN Architecture



PDN Topology



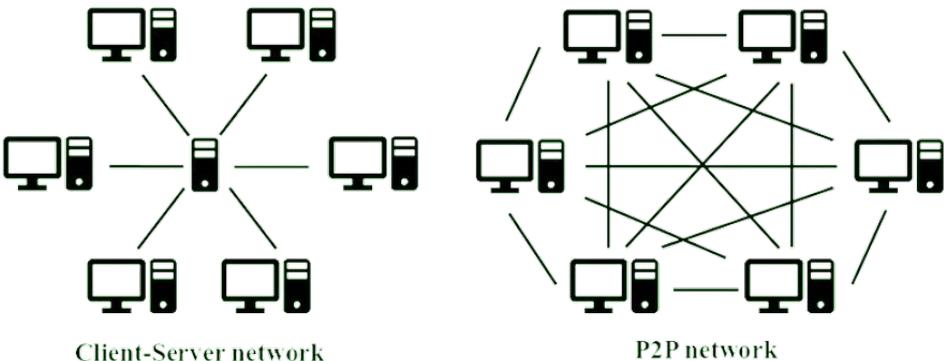
Demo: PDN



Copyright @ 2022 JFrog - All rights reserved.

Demo: PDN (Cont'd)

Artifactory vs PDN - 3Gb file



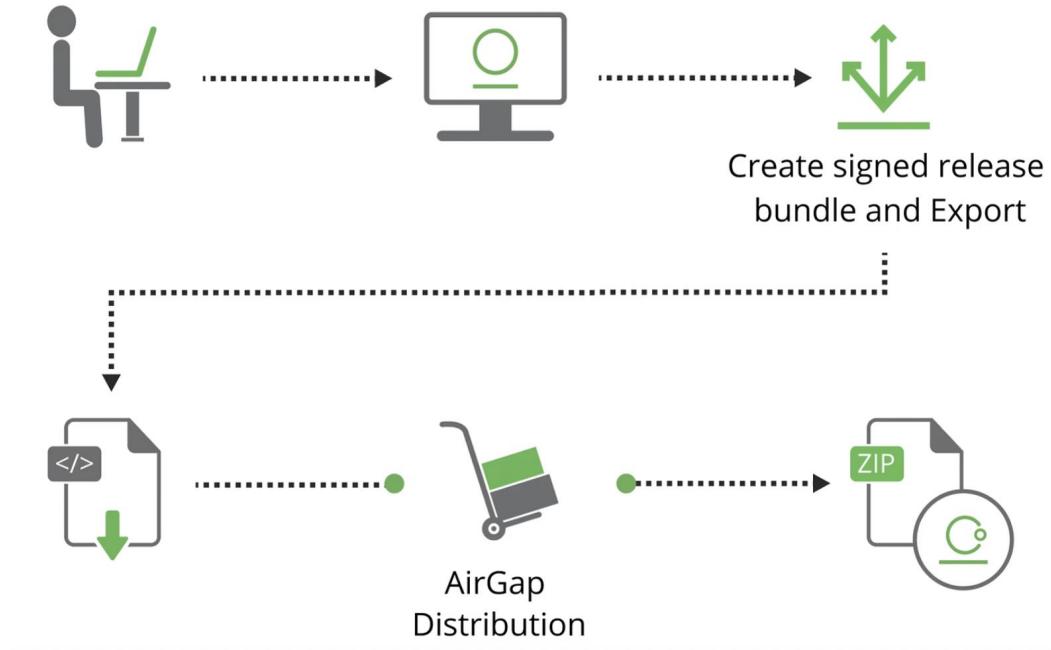
Air Gap Distribution

No communication between 2 networks

- Either by choice or by regulations, air gapped networks are used to avoid security breaches.
- An air-gapped network of devices can communicate with each other, but cannot communicate with any devices outside of the air gap.

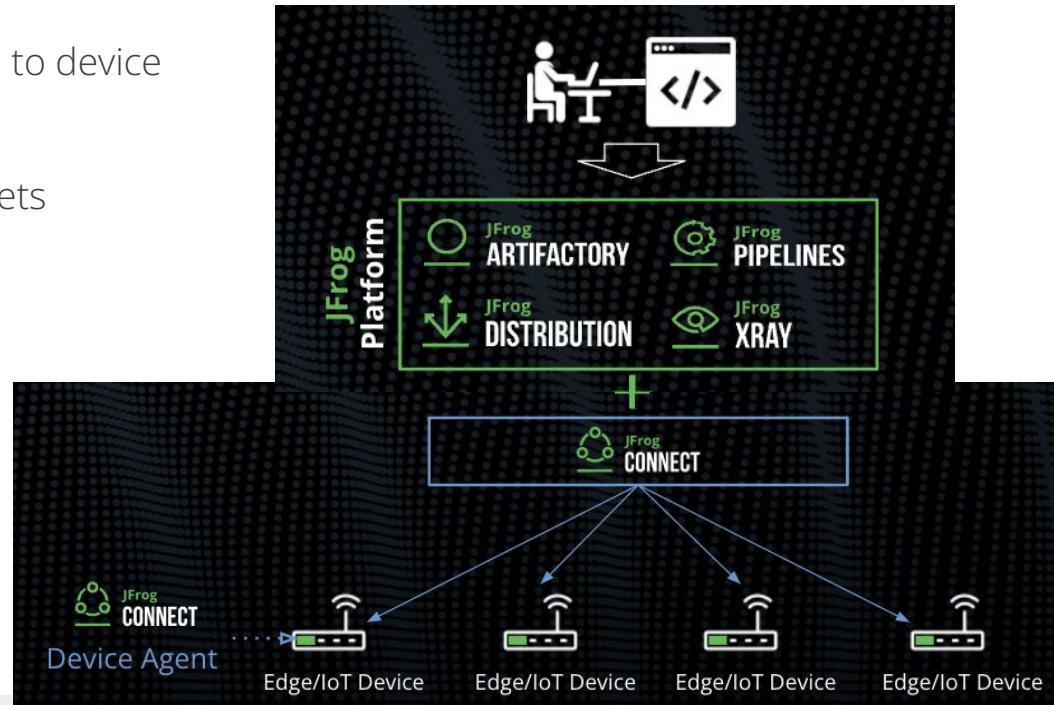
Air Gap Distribution Flow

1. Create and sign a release bundle.
2. Export the release bundle as an archive (ZIP file) from JFrog Distribution.
3. Download the release bundle.
4. Copy the archive to an external device, such as a portable hard drive or USB flash drive.
5. Import the archive to the Artifactory node on the air-gapped network.



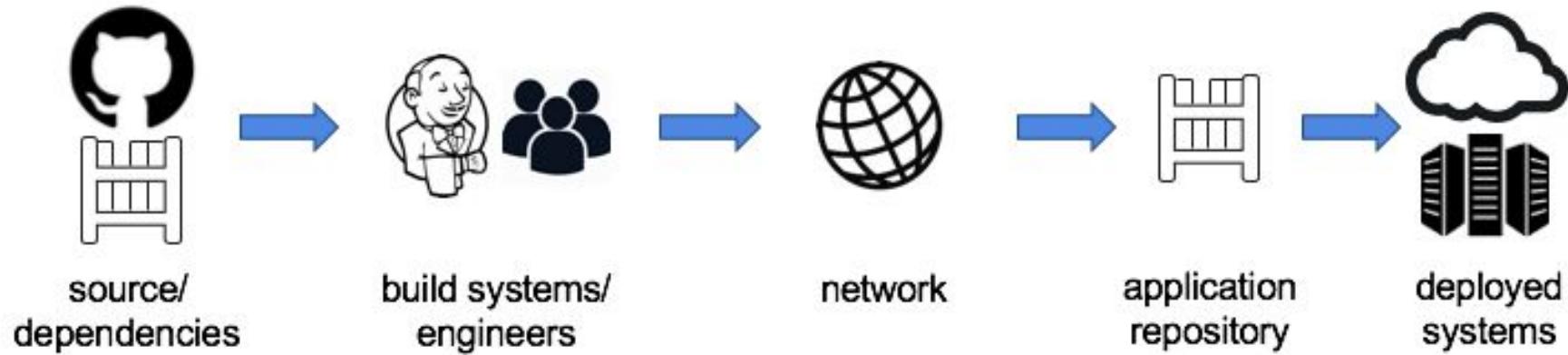
JFROG CONNECT: DEVOPS FOR CONNECTED DEVICES

- Automate software deployment from code to device
- Manage software supply-chain (SBOM)
- Keep track of software versions in large fleets
- Remediate failures remotely
- Monitor devices in the field



Secure Software Supply Chain

Everything you need for building your software: source code, 3rd party libs, OS packages, systems and infras, users and processes



JFROG RESOURCES

JFROG COURSE MATERIALS:

- JFrog Training Material - [GitHub](#)

JFROG REFERENCE DOCUMENTATION:

- JFrog Platform - [Whitepaper](#)

JFROG ACADEMY:

- JFrog Academy - [Online Training](#)

JFROG FREE TIER:

- JFrog Free Tier - [Start Free](#)



DISTRIBUTION RESOURCES

- [Documentation](#)
- [Solution Sheet](#)
- [Screencast](#)
- [Capital One Use Case](#)
- [PDN Blog](#)
- [Air Gap Distribution](#)
- [Advanced PDN Configuration](#)
- [Webinar](#)
- [IDC Infographic](#)
- [JFrog on AWS](#)
- [JFrog on AWS](#)
- [Webinar](#)
- [Air Gap Screencast](#)



THANK YOU!



The Liquid Software Company