DevOpsSchool
Lets Learn, Share & Practice DevOps

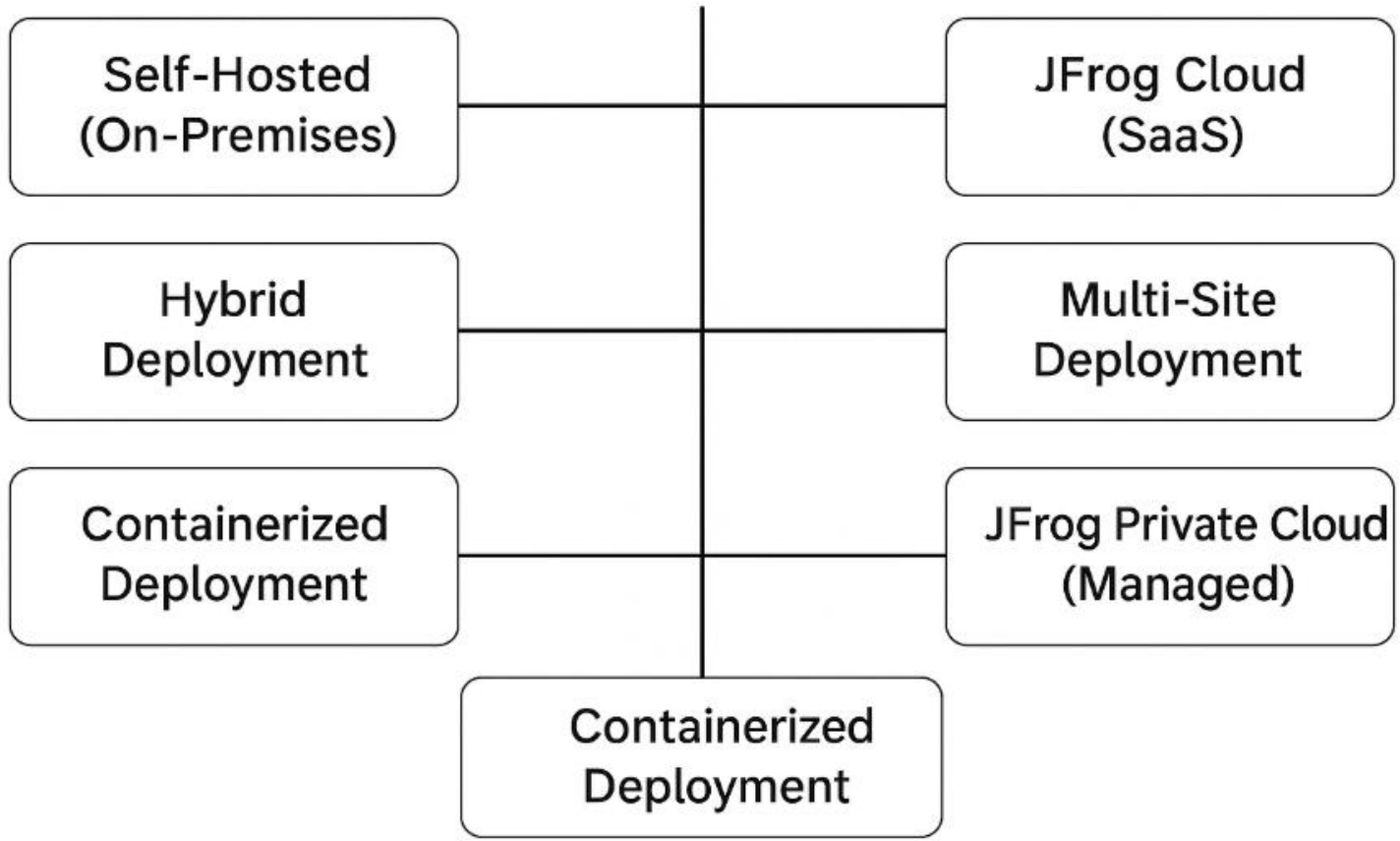
A Jfrog Artifactory

Installation & Configurations

DevOpsSchool
Lets Learn, Share & Practice DevOps

Rajesh Kumar
www.RajeshKumar.xyz
DevOps@RajeshKumar.xyz

JFrog Artifactory Deployment Models



Artifactory Deployment Models

- **Self-Hosted:** Full control, on-prem infrastructure.
- **JFrog Cloud (SaaS):** Fully managed service.
- **Hybrid:** Combination of on-prem and cloud.
- **Multi-Site:** Global, HA with federated repositories.
- **Containerized:** Docker/Kubernetes deployment.
- **Private Cloud:** Dedicated JFrog-managed VPC instance.

Artifactory Deployment Models

Deployment Model	Description	Typical Use Cases/Notes
Self-hosted (On-Premises)	Deploy Artifactory on your own infrastructure (bare metal, VM, on-prem datacenter, or private cloud).	Organizations needing maximum control over data, network, and compliance.
Self-hosted in Cloud	Run Artifactory on IaaS (e.g., AWS EC2, Azure VM, GCP Compute Engine), fully managed by user.	For full control with cloud flexibility; integrates with enterprise cloud tools.
JFrog SaaS (Cloud Hosted)	Fully managed by JFrog. Available as SaaS on AWS, Azure, GCP.	Minimal management overhead, rapid onboarding; scales elastically.
Hybrid	Combines self-hosted and SaaS for distributed or multi-site needs, often connecting global workflows.	Enterprise replication, DR, or specific compliance/disaster requirements.
Edge/On-Prem Hybrid (Edge nodes)	Artifactory Edge servers act as remote caches/distribution nodes, deployed near remote teams/locations.	For geographically distributed teams needing fast

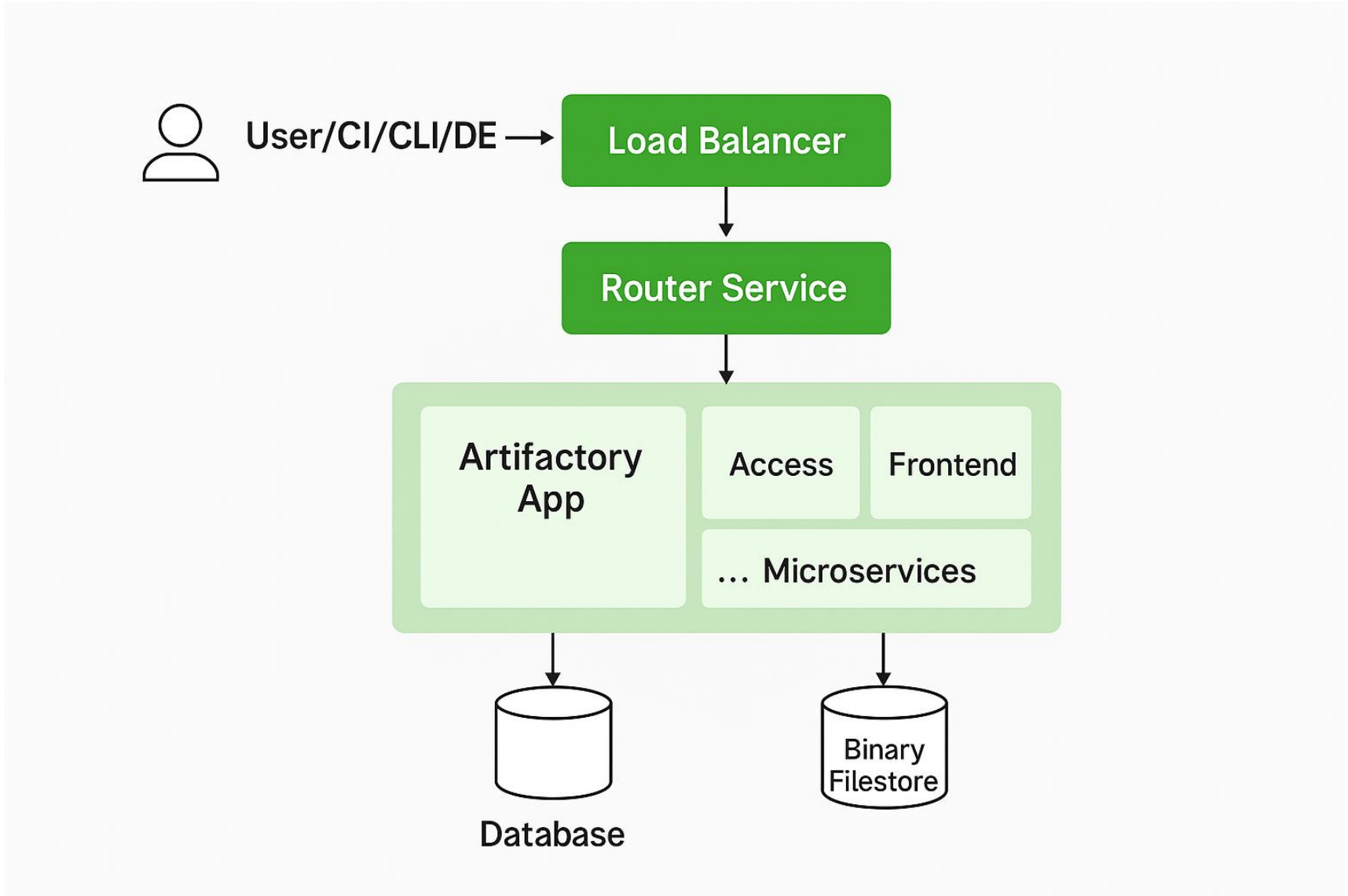
DevOpsSchool
Lets Learn, Share & Practice DevOps

Jfrog Artifactory Architecture

JFrog Artifactory acts as the core of the JFrog Platform, providing robust, universal artifact management across the software supply chain.

Its architecture is modular, scalable, and designed to fit both standalone and multi-site enterprise deployments

Antifactory Architecture: Core Components



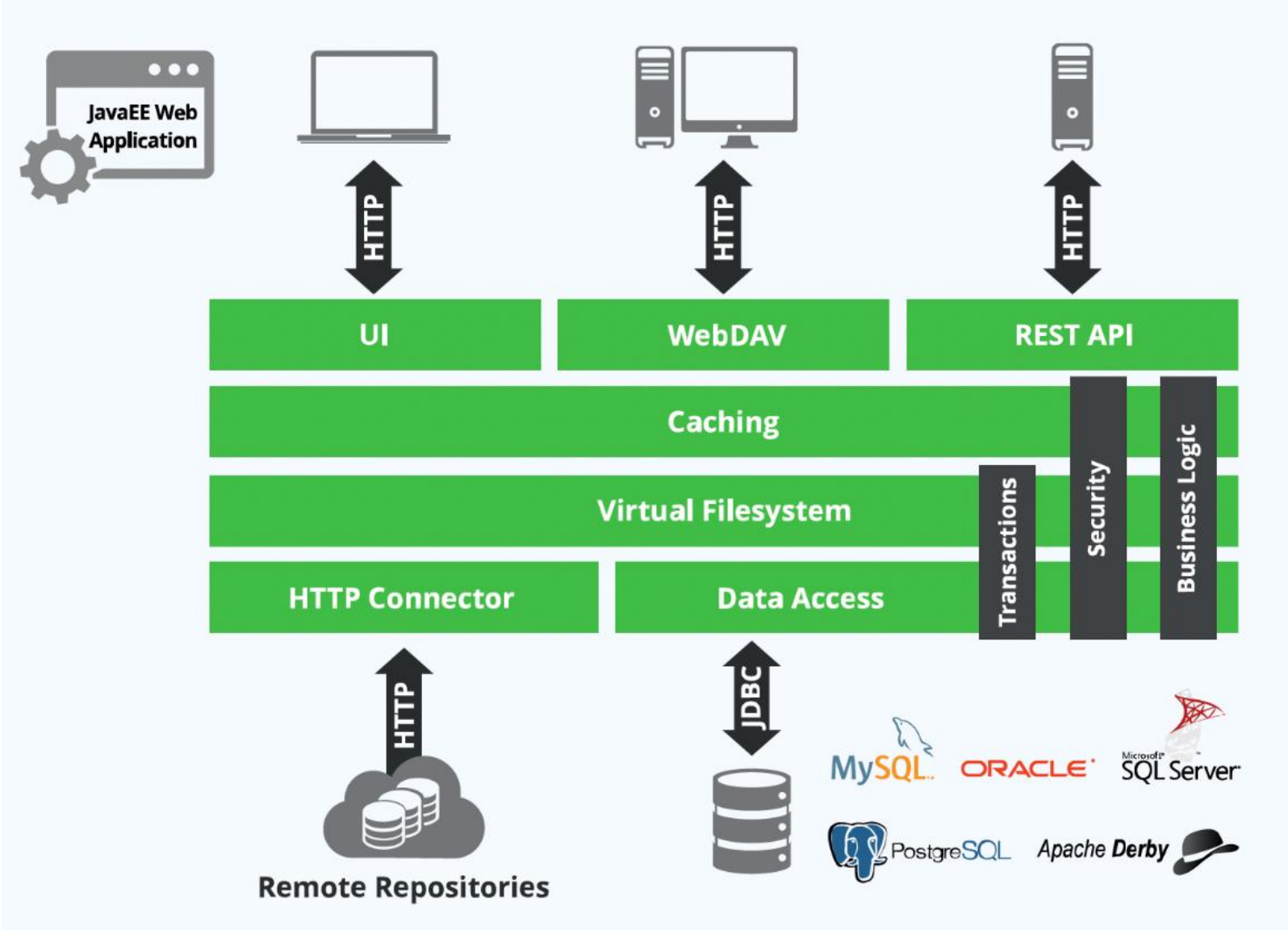
Artifactory Architecture: Core Components

Component	Description
Artifactory Application	The main server application, handling REST/API requests, artifact management, replication, and all core functions. Requires a DB and filestore to operate.
Database	Stores all metadata: user info, permissions, repo configs, artifact paths/names, and platform configuration. Recommended to use an external SQL DB (PostgreSQL, MySQL, etc.) for production.
Filestore (Binary Store)	Stores only the actual binaries (artifacts) themselves, organized by checksums (e.g., SHA-1). May use local storage, NFS, or cloud object storage (AWS S3, GCS, Azure Blob).

Artifactory Architecture: Microservices (JFrog Platform, v7+)

Service	Functionality
Router	Service discovery and internal API gateway, handling all traffic routing across microservices and products.
Access	Authentication, access control, group/user management, and access token issuance for all platform services.
Frontend	The user interface serving the web UI for all JFrog products in the deployment.
Topology	Manages platform topology, service registry, and node discovery within clustered deployments (from v7.104).
One Model	Central API model for all JFrog entities, providing consistency and a unified GraphQL API.
JFConnect	Manages subscription, licensing entitlements, and cloud checks.
Event	Handles platform-wide webhook and event distribution for automation and integrations.

Artifactory Architecture: Microservices (JFrog Platform, v7+)



Artifactory Architecture: Microservices (JFrog Platform, v7+)

DevOpsSchool
Lets Learn, Share & Practice DevOps

Top Layer – Interfaces

1.JavaEE Web Application

1. The Artifactory server application built on JavaEE standards.
2. Provides the web-based admin and user interface.
3. Acts as the core platform to serve all user and API requests.

2.UI

1. The **User Interface** (web-based dashboard).
2. Allows browsing repositories, uploading/downloading artifacts, and managing configurations via HTTP.

3.WebDAV

1. Web-based Distributed Authoring and Versioning.
2. Enables mounting Artifactory as a network drive for easy drag-and-drop artifact management via HTTP.

4.REST API

1. Provides programmatic access to Artifactory functionality.
2. Used by CI/CD pipelines, automation scripts, and integrations.

Artifactory Architecture: Microservices (JFrog Platform, v7+)

Middle Layer – Core Services

1.Caching

1. Temporarily stores frequently accessed artifacts.
2. Improves performance and reduces repeated requests to remote repositories.

2.Virtual Filesystem

1. Provides a unified, logical file structure to users.
2. Aggregates artifacts from local, remote, and virtual repositories under a single namespace.

3.Transactions

1. Ensures atomic operations (upload/download) are consistent.
2. Handles concurrent requests without data corruption.

4.Security

1. Manages authentication, authorization, and role-based access control (RBAC).
2. Integrates with LDAP, SAML, OAuth, and API tokens.

5.Business Logic

1. Implements Artifactory's core repository management rules.
2. Handles artifact promotion, replication, checksum validation, and metadata processing.

Artifactory Architecture: Microservices (JFrog Platform, v7+)

Bottom Layer – Connectors and Storage

1.HTTP Connector

1. Manages communication between Artifactory and external remote repositories via HTTP.
2. Responsible for downloading external dependencies and caching them.

2.Data Access

1. Handles interaction with the metadata database using JDBC.
2. Ensures artifact metadata, configuration, and permissions are stored reliably.

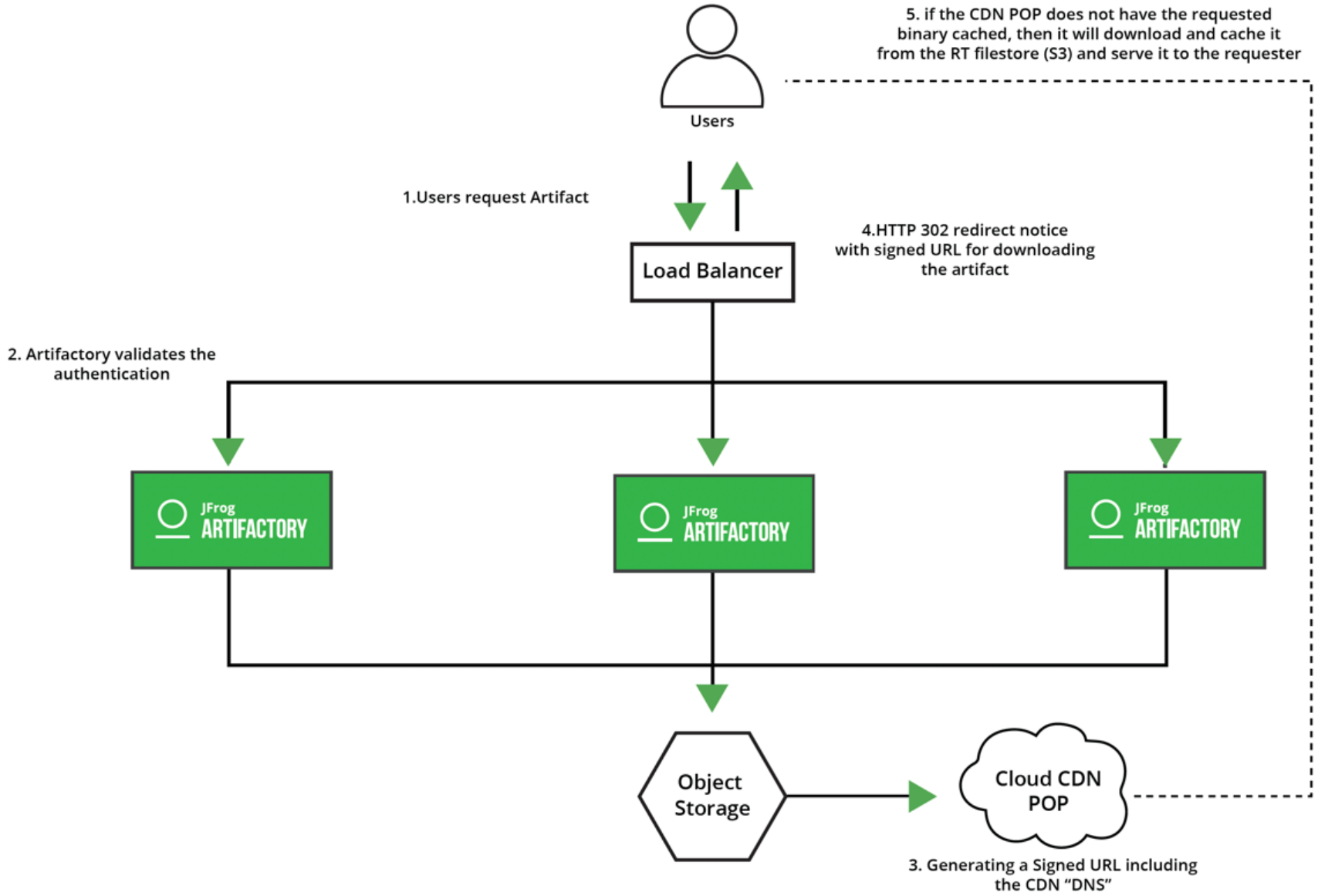
3.Remote Repositories

1. External artifact sources (e.g., Maven Central, npm registry, PyPI).
2. Artifactory proxies and caches these via remote repositories.

4.Databases

1. Stores metadata, configuration, and access control data.
2. Supported DBs:
    1. **MySQL**
    2. **Oracle**
    3. **Microsoft SQL Server**
    4. **PostgreSQL** (recommended)
    5. **Apache Derby** (for testing, not production)

Artifactory Architecture: Workflow



Antifactory Architecture: High Availability & Scalability

• HA Cluster: In enterprise mode, Artifactory nodes are grouped behind a load balancer. Nodes share the same external database and filestore, ensuring redundancy, failover, and horizontal scalability. Sessions are stateless, so users or jobs can connect to any node.

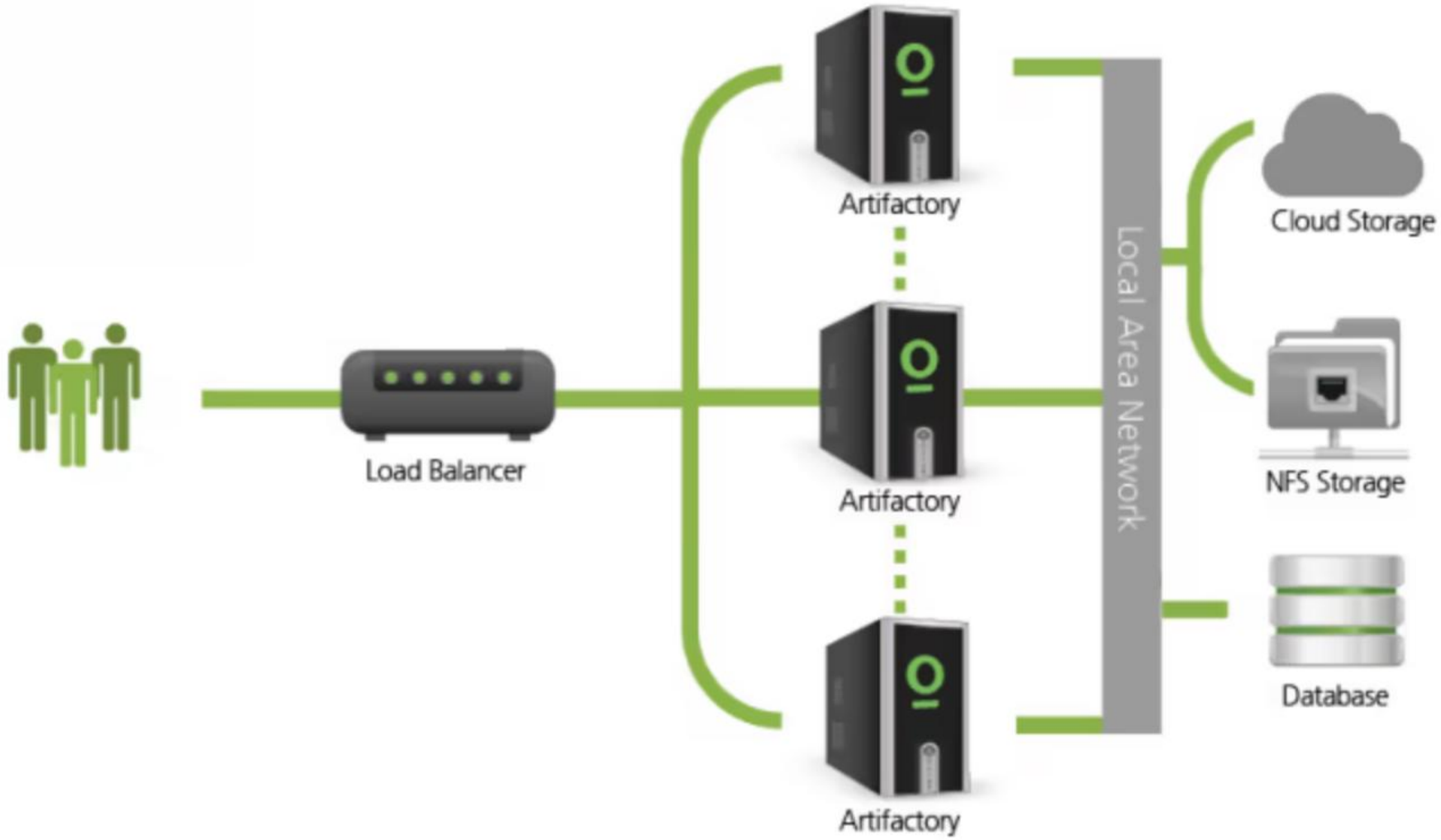
• Replication: Both unidirectional and bidirectional (federated) repository replication supported for DR, multi-site collaboration, and distributed pipelines.

• Edge Nodes: JFrog Edge servers enable caching and distribution closer to remote teams; critical for global or hybrid SaaS setups

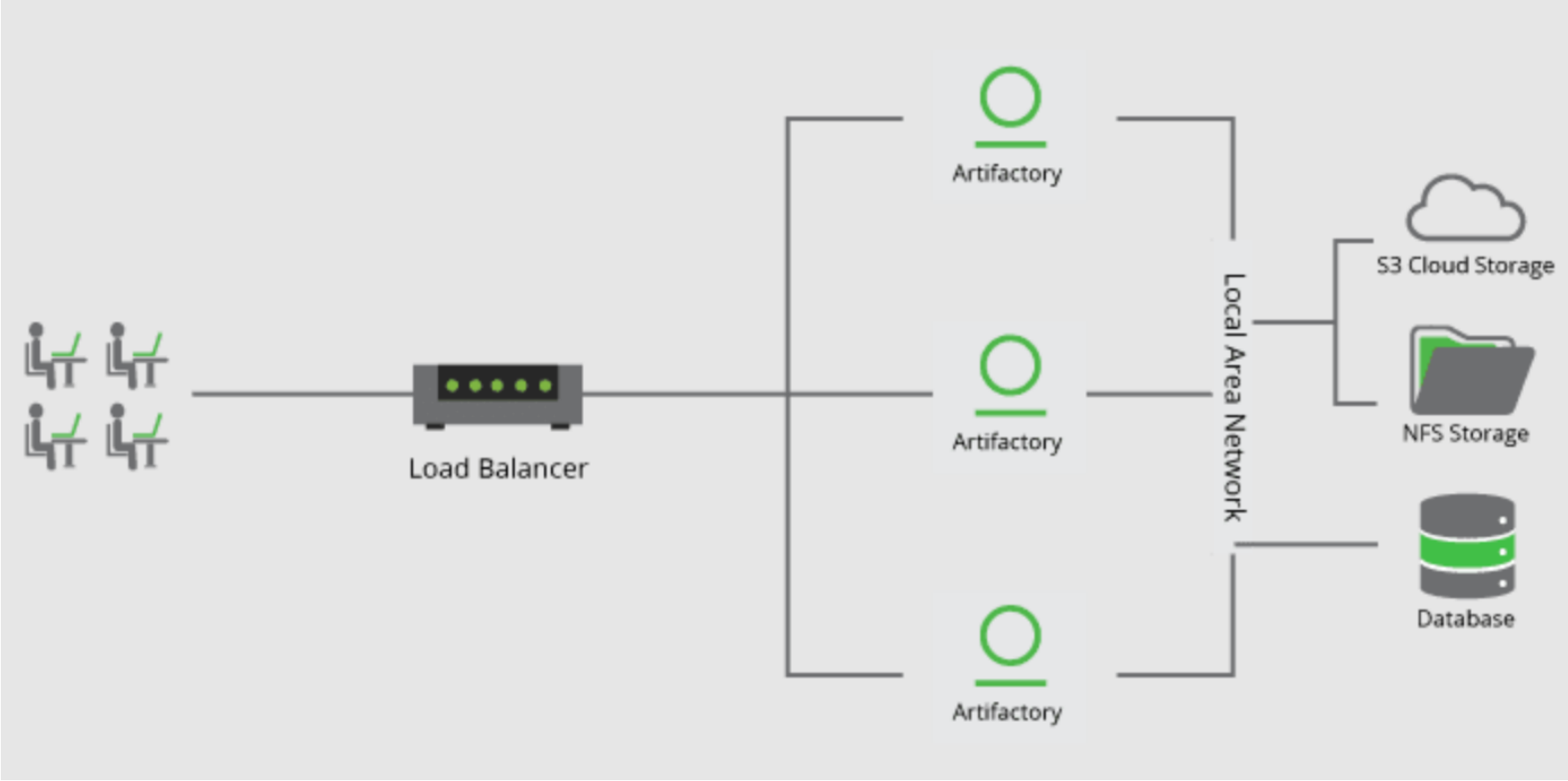
Artifactory Architecture: High Availability & Scalability

- Single Node: All services deployed on one VM/container, suitable for evaluation, small teams, or edge.
- Multi-Node / HA: Active-active clusters with externalized DB and filestore for production.
- Hybrid/SaaS: Combines JFrog-managed SaaS core with on-prem or edge nodes for copliance or performance.
- Cloud-Native: Supports Kubernetes, scalable object storage, integrations with CSPs.

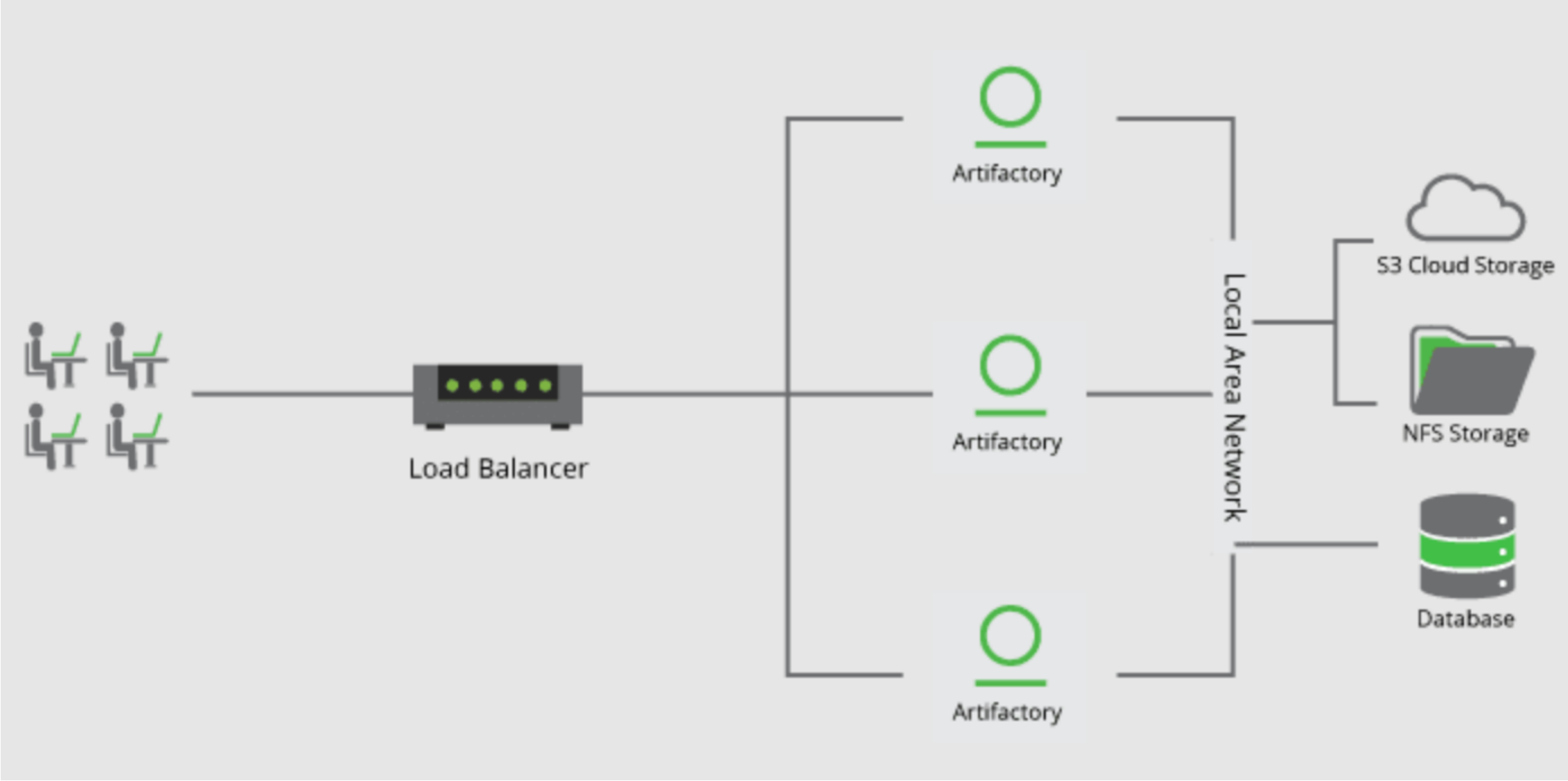
Artifactory Architecture: High Availability



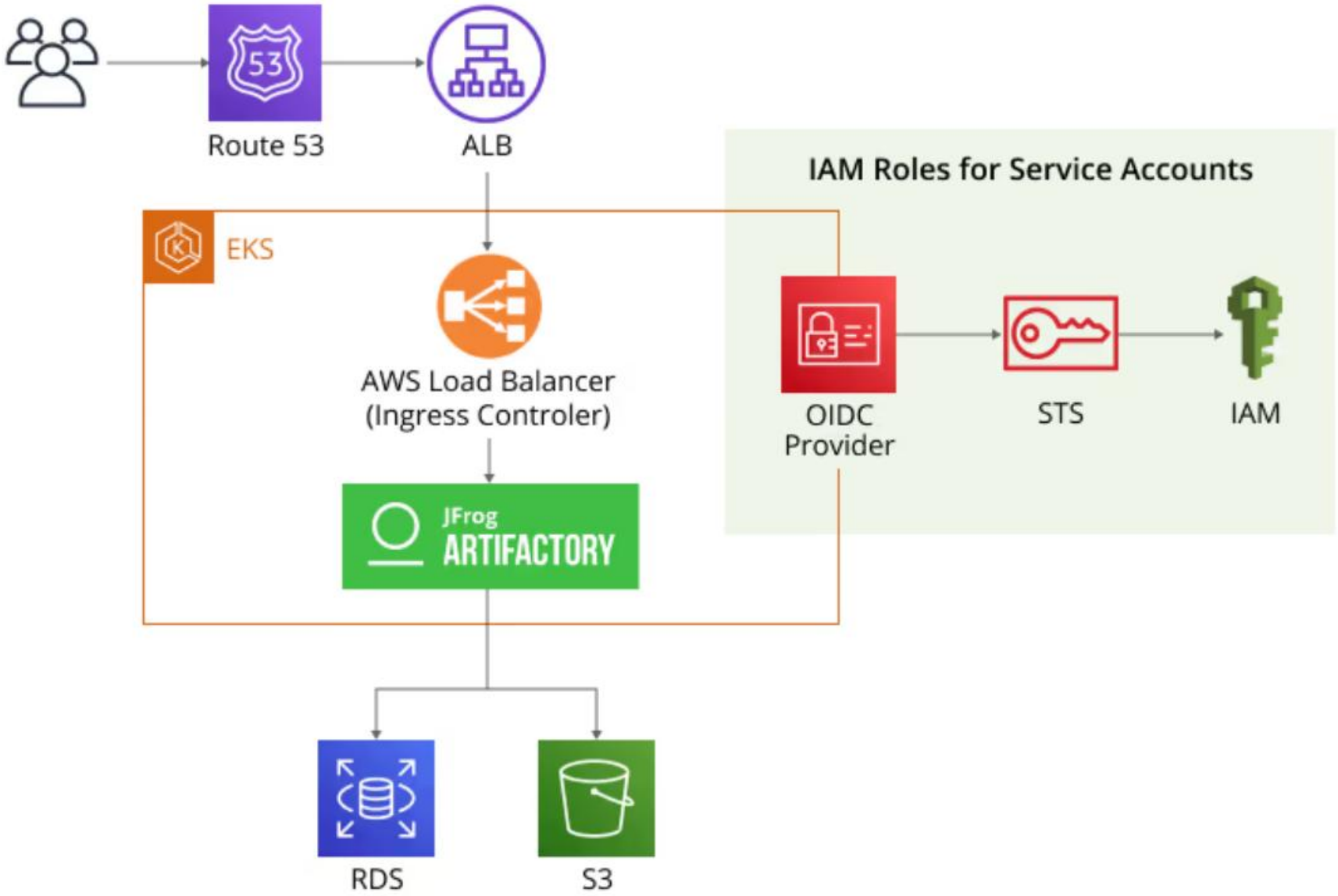
Artifactory Architecture: High Availability

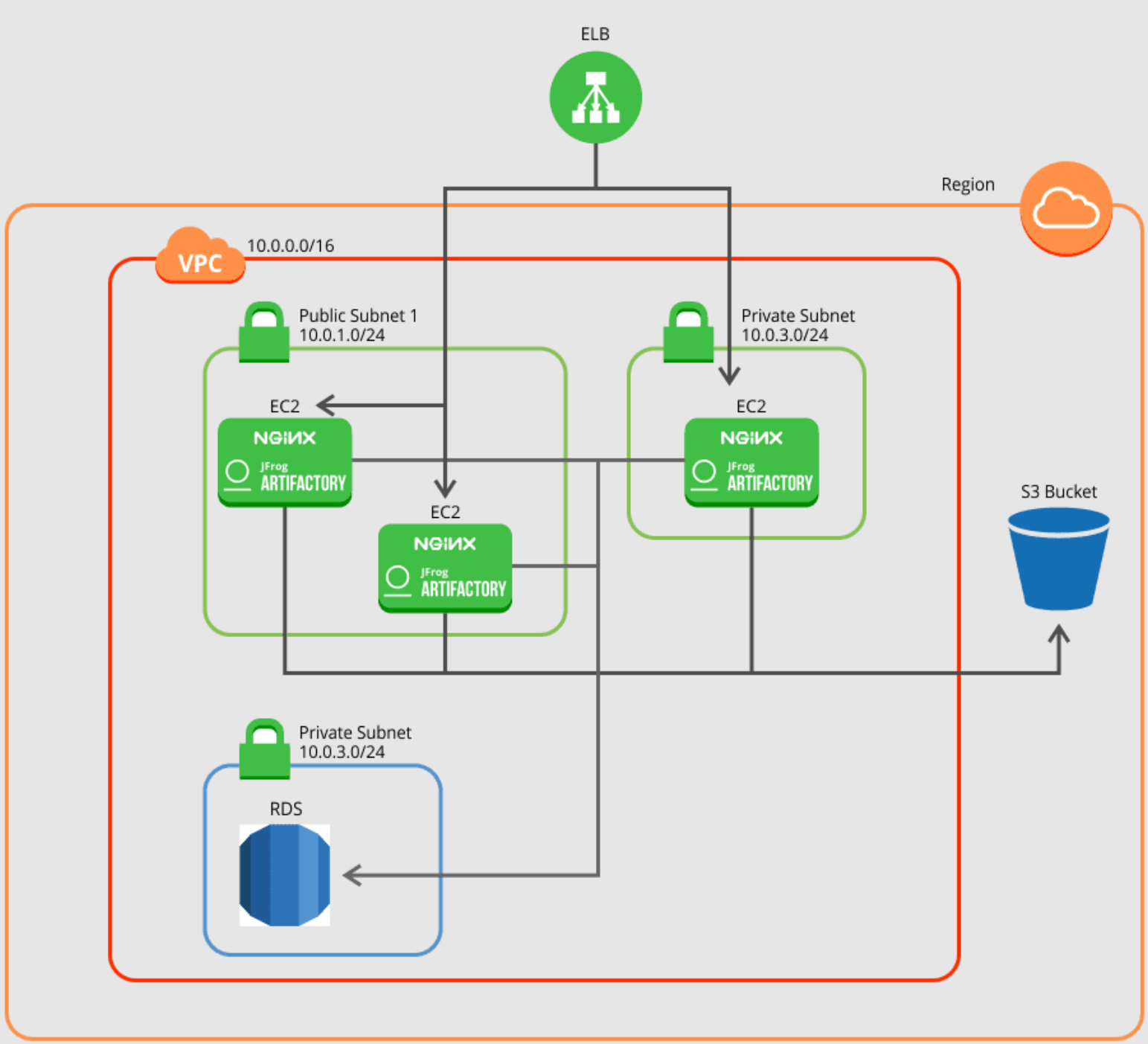


Artifactory Architecture: High Availability



Artifactory Architecture: High Availability





Artifactory Architecture: High Availability

Antifactory Architecture: Summary

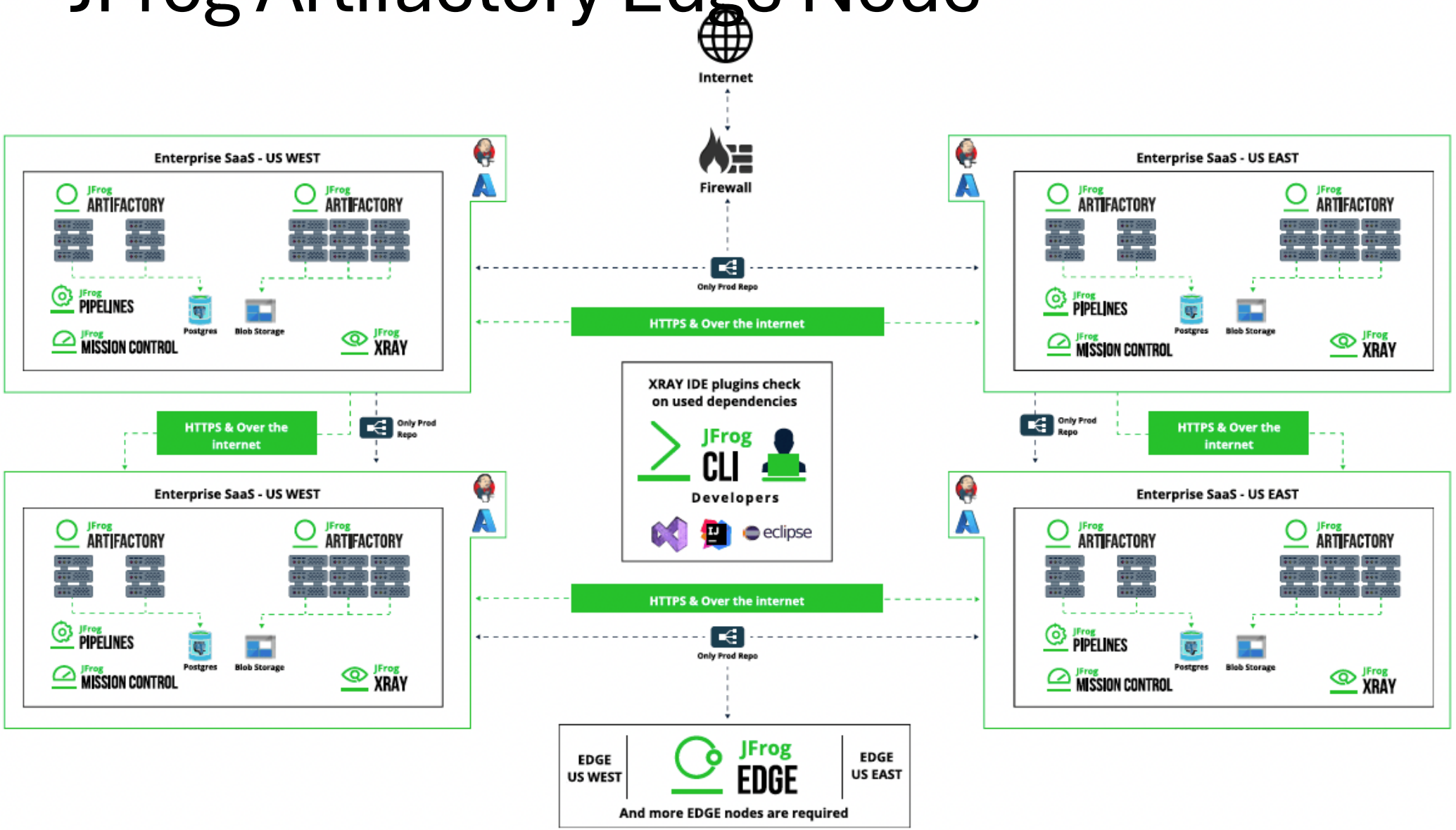
Capability	Description
Artifact Storage	Highly scalable & pluggable filestore; all binary assets managed via checksum
Metadata Management	Relational DB for all config, repo, and artifact metadata
Multi-tenancy & RBAC	Fine-grained role/access control throughout
API/Automation	Full-featured REST API, scripted automation, and webhook/event platform
Scalability & HA	Active/active cluster mode with shared DB/file store; seamless failover, rolling upgrades
Multi-site/Edge	Support for federated and replicated topologies; edge node distribution for global organizations
Security & Compliance	Encrypted storage, access controls, audit logs, optional integrated scanning and policy engines (via JFrog Xray)
Observability	Unified logs, queryable event streams, and usage dashboards

JFrog Artifactory Edge Node

A **JFrog Artifactory Edge Node** is a lightweight Artifactory instance designed to be deployed in **remote sites, branch offices, or edge locations** to bring artifacts closer to distributed teams and environments. It is part of the **JFrog Enterprise+** and **multi-site distribution architecture**.

- It acts as a **read-only or limited-write replica** of a central Artifactory.
- Optimized for **low-latency access** to artifacts in geographically distributed environments.
- Uses **JFrog Distribution and Release Bundles** to securely deliver immutable artifacts to edge sites.

JFrog Artifactory Edge Node



JFrog Artifactory Edge Node: How It Works

Central Artifactory / Distribution Node

- Your main JFrog Enterprise instance stores master artifacts and release bundles.
- When a new version is ready for deployment, it is packaged as a Release Bundle.

Release Bundle Distribution

- Using JFrog Distribution, the release bundle is signed, versioned, and pushed to one or more Edge Nodes.
- This process uses secure channels and checksum validation to ensure integrity.

Edge Node Receives Artifacts

- The Edge Node stores the artifacts locally.
- Artifacts are immutable to prevent tampering and guarantee consistency across sites.

Local Access at the Edge

- Developers, CI/CD pipelines, or deployment scripts at the remote site can pull artifacts from the local Edge Node instead of the central server.
- Reduces latency and avoids bandwidth issues with long-distance connections.

Synchronization

- Edge Nodes synchronize metadata and updates with the central instance periodically or on-demand.
- Ensures all sites stay in sync with the latest releases.

Jfrog Artifactory System Requirements

Requirement Area	Minimum / Recommended	Notes
CPU	4–8 cores	Scale up for more users/HA
RAM	6–18GB	Scale up for >200 users or heavy use
Disk	200–300GB SSD (~3× artifact size)	Use high IOPS SSD for best performance
OS/Platform	Ubuntu 20.04+/Debian 11+/RHEL 8+/Win 2016+/K8s 1.27+/ARM64	See above platform support table
Java	OpenJDK 17+	Check your Artifactory version
Database	PostgreSQL (recommended), MySQL, MariaDB, Oracle, MS SQL	HA always requires external DB
Filestore	Local FS/NFS/Cloud (S3/Azure/GCS)	Not supported: install app on NFS
Ports	8000–8100 (internal), 8081/8082 (external)	Plus DB and other service ports as needed
Privileges	Linux root/Windows Admin	For setup/config/upgrade
System Time	Synchronized	Recommended for all nodes in cluster

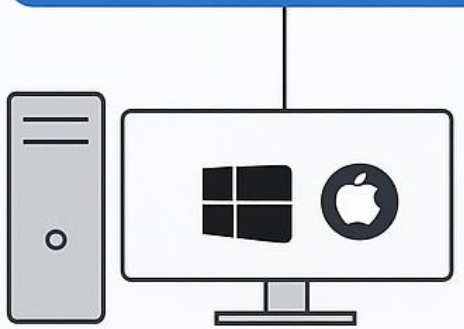
JFrog Artifactory System Requirements

Small Team / Evaluation

CPU 4 cores

Memory 8–32 G RAM

Storage 100 GB



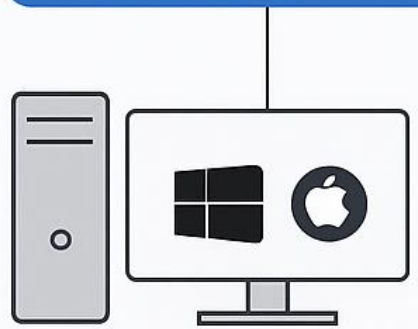
PostgreSQL

Production

CPU 8 cores

Memory 16–3G RAM

Storage 500 GB



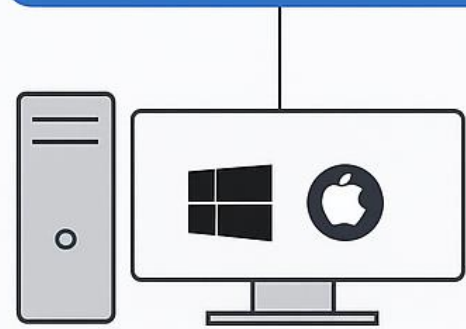
PostgreSQL

Enterprise / HA

CPU 16 cores

Memory 32–64 G BM

Storage 1 TB+



Object Storage

NFS

Postgre SQL

Recommended Hardware

The following table provides hardware recommendations for a single server machine:

Number of developers	OS/JVM	Processor	*Memory (RAM) for JVM Heap	Storage
1 - 20	64 bit	4 cores	4GB	Fast disk with free space that is at least 3 times the total size of stored artifacts
20 - 100	64 bit	4 cores	8GB	Fast disk with free space that is at least 3 times the total size of stored artifacts
100 - 200	64 bit	8 cores (16 cores recommended)	12GB	Fast disk with free space that is at least 3 times the total size of stored artifacts (backup SAN recommended)
200+	64 bit	Please contact JFrog support for a recommended setup.		

Artifactory Database

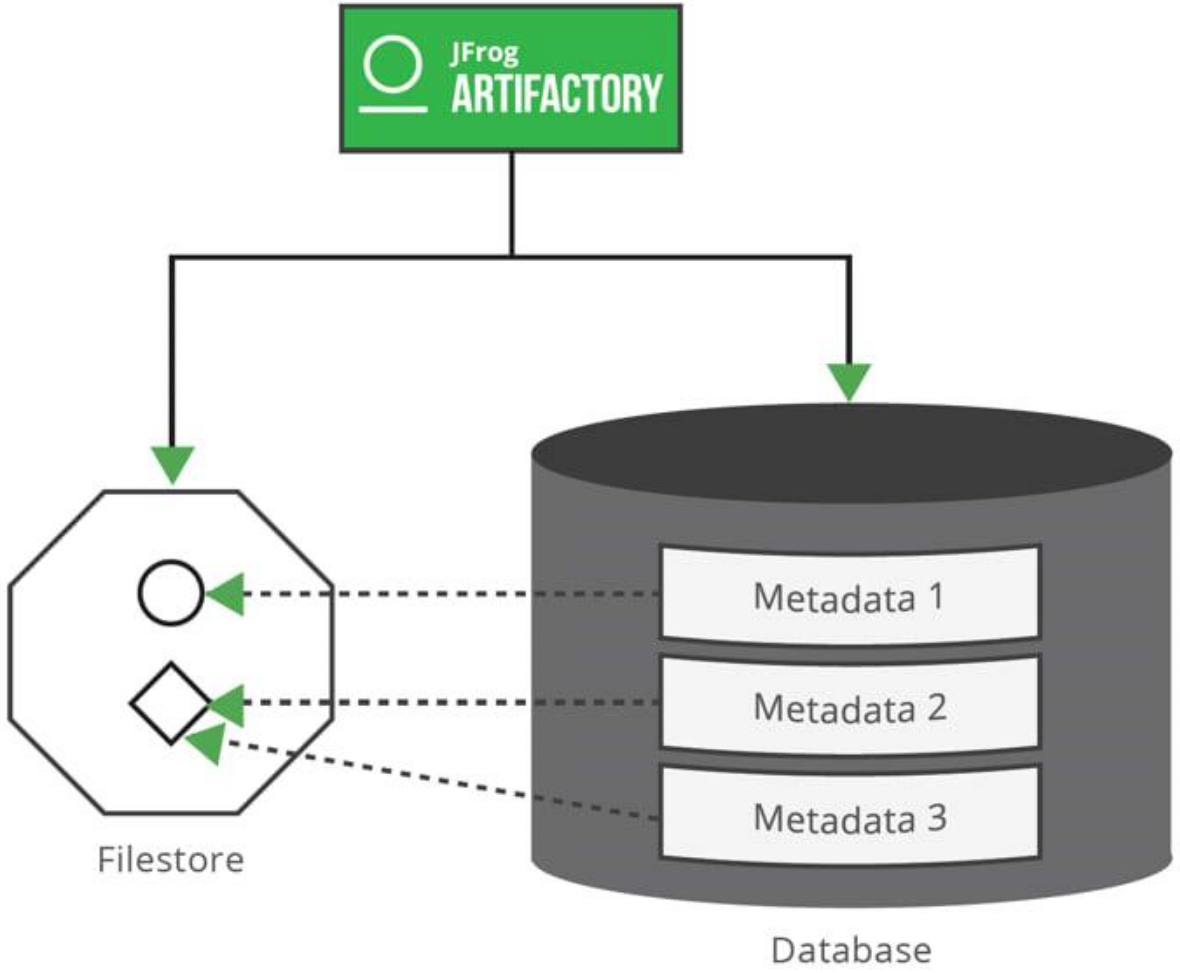
Artifactory currently supports the following databases:

- Derby (The default embedded database)
- MySQL v5.5, 5.6 and 5.7 with InnoDB
- Oracle version 10g and above
- Microsoft SQL Server 2008 and above
- PostgreSQL v9.2 and above
- MariaDB v10.2.9 and above

Artifactory: Modes of Operation

- Artifactory supports two modes of operation:
  - Metadata in the database and binaries stored on the file system (This is the default and recommended configuration). Artifactory uniquely stores artifacts using checksum-based storage
  - Metadata and binaries stored as BLOBs in the database

JFrog Artifactory Database and Metadata



Artifactory filestore

- JFrog Artifactory offers flexible filestore management built-in set of chains..
  - file-system
  - cache-fs
  - full-db
  - full-db-direct
  - s3
  - google-storage
  - double-shards
  - redundant-shards
  - cluster-file-system
  - cluster-s3
  - cluster-google-storage
  - cluster-azure-blob-storage

Artifactory Checksum-Based Storage

- Artifactory uniquely stores artifacts using checksum-based storage.
- A file that is uploaded to Artifactory, first has its SHA1 checksum calculated, and is then renamed to its checksum. It is then hosted in the configured filestore in a directory structure made up of the first two characters of the checksum.
- From version 5.5, Artifactory natively supports SHA-256. An artifact's SHA-256 checksum is calculated when it is deployed to Artifactory, and is maintained in persistent storage as part of the database.

Setup the New Database

To setup your new database you need to perform the following steps:

1. Create a database instance
2. Create an Artifactory user for the database
3. Install the appropriate JDBC driver
4. Copy the relevant database configuration file
5. Configure the corresponding db.properties file.
6. Start Artifactory
7. Import the metadata using Full System Import

How to Download Artifactory Package

https://jfrog.com/download-jfrog-platform/ https://jfrog.com/download-legacy/?product=artifactory

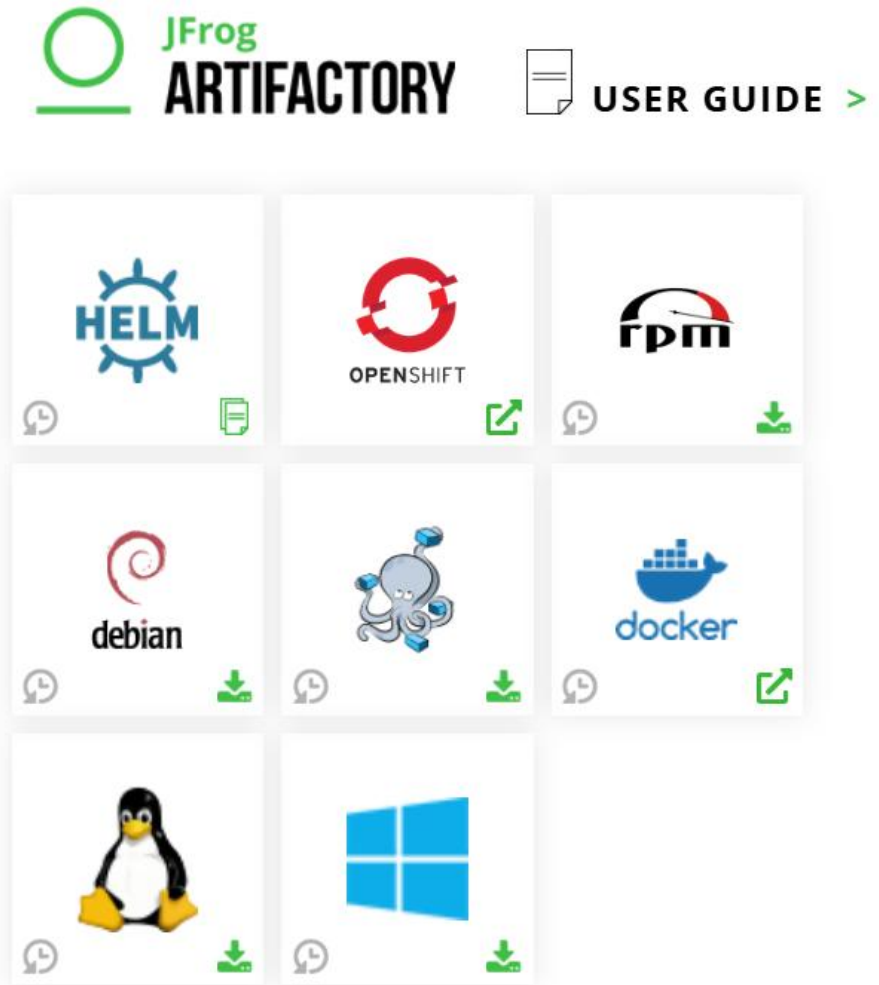
Artifactory 7.117.7
July 25, 2025
See Release Notes >
Installer type: HELM rpm debian

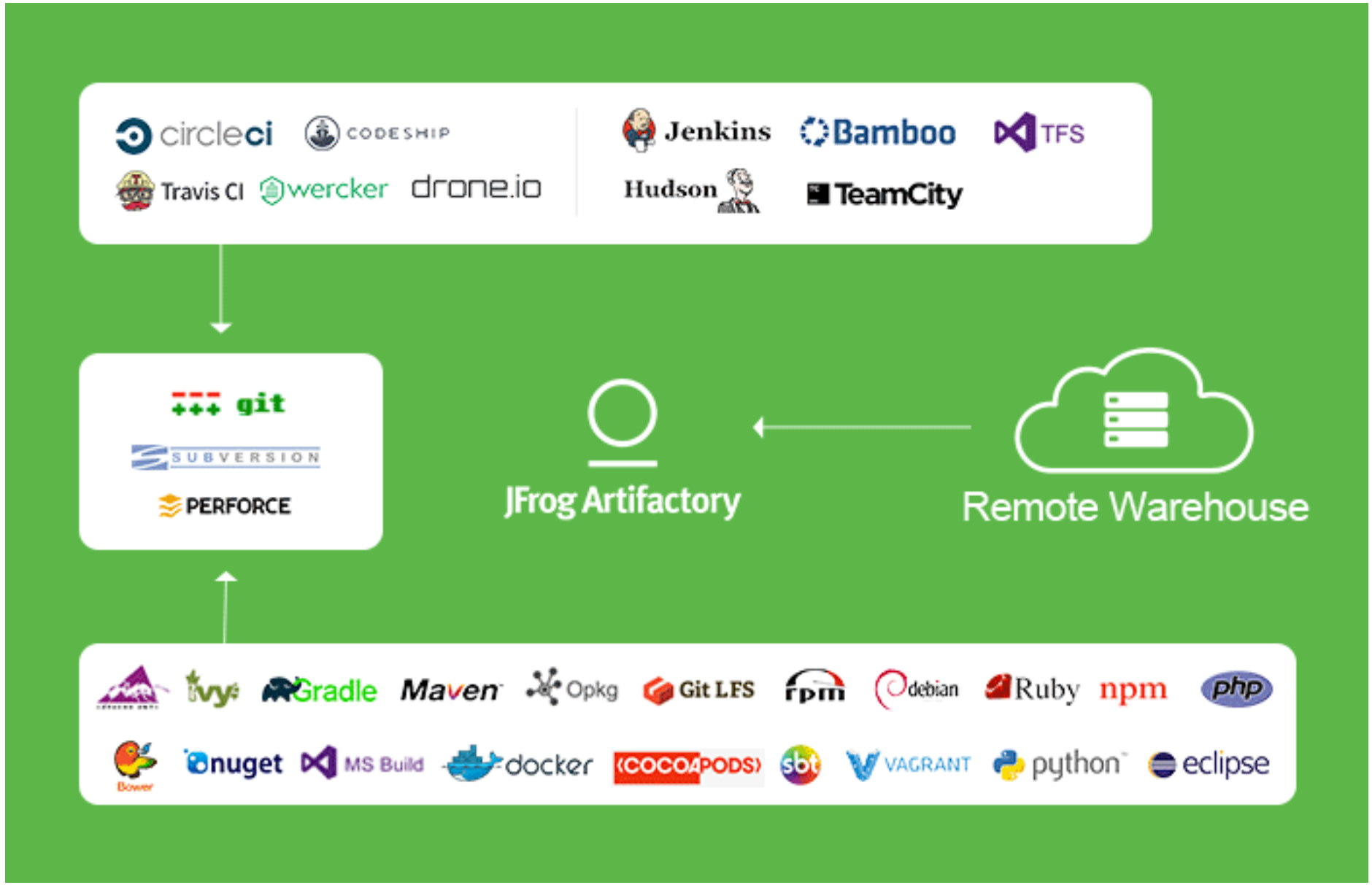
Artifactory 7.117.5
July 24, 2025
See Release Notes >
Installer type: HELM rpm debian

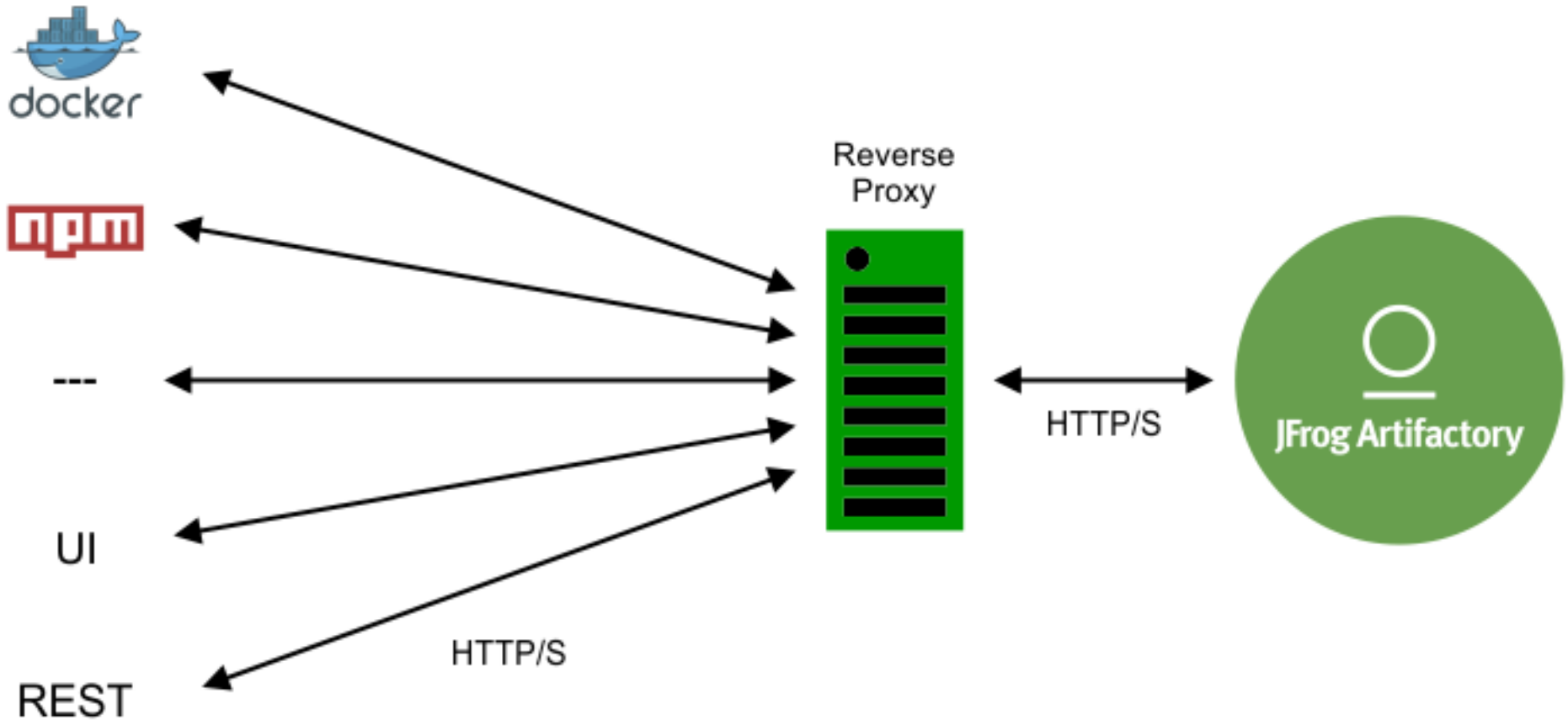
Artifactory 7.111.12
July 13, 2025
See Release Notes >
Installer type: HELM rpm debian

Artifactory 7.111.11
July 3, 2025
See Release Notes >
Installer type: HELM rpm debian

Artifactory 7.111.10
June 17, 2025
See Release Notes >
Installer type: HELM rpm debian




circleci
CODESHIP
Travis CI
wercker
drone.io
Jenkins
Bamboo
TFS
Hudson
TeamCity
git
SUBVERSION
PERFORCE
JFrog Artifactory
Remote Warehouse
ivy
Gradle
Maven
Opkg
Git LFS
rpm
debian
Ruby
npm
php
Bower
nuget
MS Build
docker
COCOAPODS
sbt
VAGRANT
python
eclipse



How to access?

Web UI:
Open a browser and go to
http(s)://<server>:8082/artifactory/
Log in with your username/password or SSO.

Command Line:
Use the JFrog CLI:

Configure: jfrog config add

Upload/download: jfrog rt upload … / jfrog rt download …

API:
Use REST calls to
http(s)://<server>:8082/artifactory/api/
Authenticate with API key, access token, or basic auth.

How to login?

- admin/password