



GET READY FOR HIGH AVAILABILITY

All you need to know about HA and how you can achieve it

In a world where software is running all around us, globally, 24/7, being highly-available is a must.

As your organization grows, the services you provide need to handle heavier loads, ongoing maintenance and any unexpected system faults. Keeping maximum uptime is top priority.

WHAT IS HIGH AVAILABILITY?

Availability is defined as the probability that a system is not in a failed state or undergoing a repair action when it needs to be used. Setting up high availability essentially means increasing your system performance with no single point of failure, seamless upgrades and most importantly meeting today's modernized software expectation. This is done by setting up clusters of multiple nodes that act as a single unit. Workloads can then be distributed over the cluster nodes, and all operations continue flawlessly as all nodes in the cluster handle cluster-wide tasks.

Supporting high availability and a multi-node setup is essential for any cloud native application.

In cloud native environments such as Kubernetes, high availability is crucial since nodes are dynamically provisioned in the cluster, according to the need. In this world of cloud native setups, the expectation is that a failure of one node in the cluster will not impact the availability as well as the performance of the whole cluster. Cloud native applications know how to run and adapt themselves accordingly in multi-instances on demand.

HIGH AVAILABILITY ARCHITECTURE

An HA cluster configuration includes three basic components: network load balancer, application(s) and common resources.

- **Network Load Balancer:** the entry point to the cluster, distributing and routing requests to the nodes in your system.
- **Application:** the service or application running in HA mode.
- **Common Resources:** the set of shared resources for your services. For example, an external database and filestore.

Note: to ensure good performance, install all the components of your HA installation on the same high-speed Local Area Network (LAN).

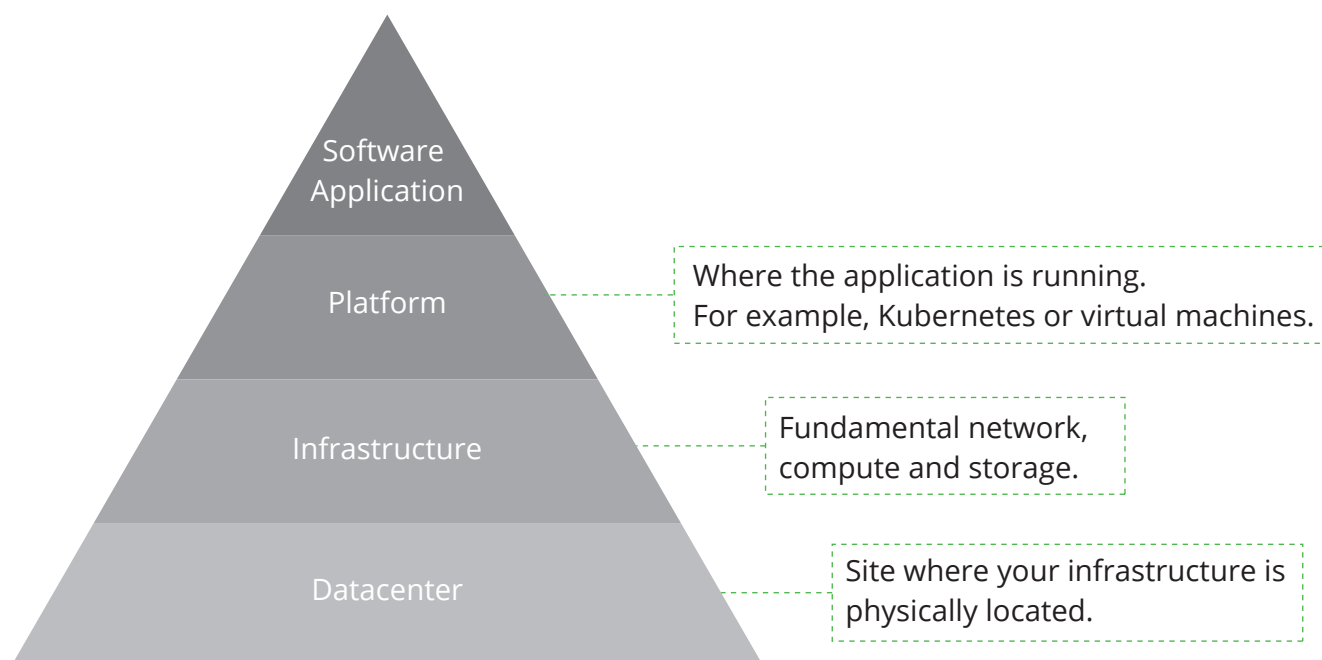
RECOMMENDED HA SETUP

The larger the cluster, the higher the efficiency. 2-node and 3-node clusters will have a higher performance from a single node.

A cluster will have at least 2-nodes, however it is recommended to have an n+1 redundancy cluster, where n represents the minimal number of nodes that are needed to be active with no service level deterioration. The idea is that if one of the nodes in the cluster fails, you are still running in an HA setup. Giving you the time you need to investigate and resolve the issue.

Your overall systems' HA is measured according to the weakest less-redundant component in your system. It is not enough to have an HA cluster for your application, all system components should be redundant as well.

There are different levels of HA, including:



SCALE LIMITLESSLY WITH NO DISRUPTION TO PRODUCTION

JFrog offers a mature HA solution proven to serve millions of requests and terabytes of data a day. Artifactory HA works with external shared resources, and includes live failover and non-disruptive production upgrades. Users are not limited to any number of nodes and can **scale without worrying about degraded performance**.

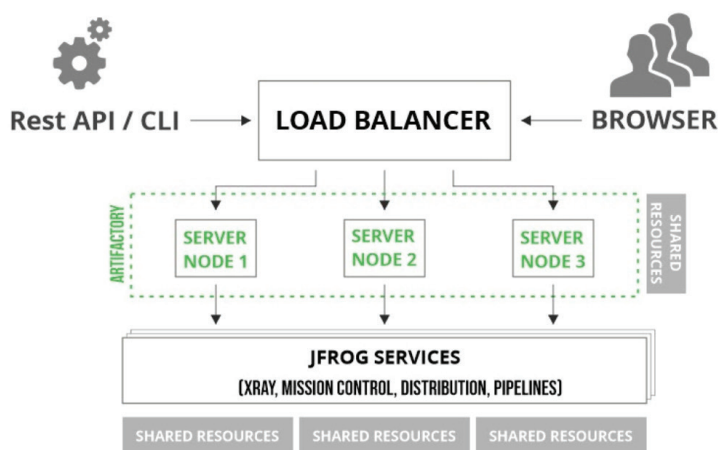
JFrog Artifactory offers a built-in active/active (all nodes receive traffic) cluster-based HA solution. Users can start with a 2-node cluster and scale with superior performance, supporting all Artifactory capabilities in HA modes.

JFROG HIGH AVAILABILITY ARCHITECTURE

The [JFrog DevOps Platform](#) fully supports an end-to-end highly available solution for managing, securing and distributing your software, with these benefits:

- **Optimal resilience**, to maximize your uptime.
- **Improved performance**, to scale your environment with as many nodes as you need.
- **Managed heavy loads**, accommodating larger load bursts with no compromise to performance.
- **Always being synchronized**, seamlessly and instantly across all cluster nodes.

The [JFrog Platform HA implementation](#) includes the following:



Load Balancer

The network load balancer is the entry point to your JFrog Platform Deployment and optimally distributes requests to the nodes in your system. The load balancer is only required for the Artifactory service, which is then responsible to route and balance between the nodes of the other services. For additional information refer to [Configuring Load Balancer](#).

Application

A JFrog service (or application) running in HA mode represents a cluster of two or more nodes that share common resources. All JFrog services (Artifactory, Xray, Pipelines, Mission Control, Distribution) can be run in HA mode, though only Artifactory requires a load balancer. Each cluster node runs all microservices, as described in the [System Architecture](#).

Common Resources

Each service requires a set of common resources. The resources vary per service but typically include a least one database.

Local Area Network

To ensure good performance, all the components of your HA installation must be installed on the same high-speed LAN.

In theory, HA could work over a Wide Area Network (WAN), however in practice, network latency makes it impractical to achieve the performance required for [high availability systems](#). Geo-redundancy should be used when it comes to preventing service failure in the face of catastrophic events such as natural disasters that cause system outages.

COMPETITIVE MATRIX

	 JFrog ARTIFACTORY	 sonatype	 GitLab		 Azure DevOps	 HARBOR	 Red Hat Quay
Supported Formats in HA Mode	27+ (All)	Limited (50% of supported formats)	< 10 (All)	< 5 (All)	< 10 (All)	< 5 (All)	< 5 (All)
Scalability	Unlimited	Recommended to limit to 3 nodes in a cluster	Unlimited	Unlimited	Unlimited	Unlimited	Unlimited
Rolling Upgrades (No Downtime Cluster Upgrades)	Yes	No	Yes	No	Yes	Yes	No
Monitoring	Yes out-of-the-box integrations	Limited	Yes out-of-the-box integrations	Yes out-of-the-box integrations	Yes out-of-the-box integrations	No	Yes out-of-the-box integrations

ARE YOU HA READY?

[Try it out for yourself >](#)

(Be sure to select the multi-server trial checkbox)