



CLOUD MIGRATION 101

Considerations and best practices
for migrating your DevOps to the cloud

Copyright © 2023 JFrog Ltd.

Publish: October 2023

table of contents

Introduction.....	2
JFrog Cloud (SaaS) vs. Self-Hosted.....	3-4
Why our customers are adopting cloud DevOps.....	5
Common cloud migration roadblocks.....	6-7
Cloud migration strategies.....	8-9
How to prepare for a cloud migration.....	10-11
Migrating to the cloud with JFrog.....	12
Next steps: walking the walk.....	13

Introduction

Here's something that won't surprise you: DevOps is moving to the cloud. Organizations across industries are investing tons of resources in cloud technologies, and have been for some time. According to **Statista**, half of organizations have workloads in the public cloud already, and 7 percent are expected to migrate more workloads in the next 12 months.

Here's something else that won't surprise you: To deploy cloud workloads, you need to expand your DevOps infrastructure to the cloud as well – it's just common sense. Last year alone, we migrated hundreds of customers and nearly 5 PB of data from self-managed, on-prem to hybrid or multi-cloud environments.

In this whitepaper, we'll discuss the reasons we see customers adopting cloud-centered DevOps, the most successful cloud migration strategies, as well as cloud onboarding best practices we've learned along the way. Our goal is to make your cloud migration experience as seamless as possible, using lessons learned from helping thousands of customers migrate successfully.

JFrog Cloud (SaaS) vs. Self-Hosted

Before diving into the differences between cloud and on-prem more generally, let's talk about what it means to leverage JFrog in the cloud (SaaS), versus self-hosted. When it comes to the JFrog Platform, there are distinct advantages and disadvantages to consider when choosing between a self-hosted deployment or SaaS delivery model.

Self-hosted deployments give organizations complete control over their installation. This includes managing their own security, scaling, and data backups, as well as potential locality adjacent to their other DevOps toolset in their own datacenter. The downside is that this requires a lot of effort to manage, and even more to maintain, in-house.

On the other hand, SaaS deployments offer the convenience and time savings of an out-of-the-box solution with no hardware, setup, or maintenance required. Security, scaling, upgrades, and data backup are all handled by JFrog, removing the need for organizations to manage these elements themselves.

JFrog Cloud (SaaS)

A JFrog Cloud (SaaS) account is a service that's fully managed by JFrog and offers you the freedom to choose the cloud provider and region it will be hosted on (i.e. AWS, Google Cloud Platform, and Azure, as well as among supported cloud regions). When you create your JFrog account through the cloud provider's public marketplace, payment is made with

cloudcredits from your existing cloud provider account, making it much more convenient.

No installation is required as JFrog will take care of everything for you. All you have to do is sign up with the cloud provider and region of your choice, and you can instantly reap the benefits of the JFrog Platform. Plus, JFrog provides a **high SLA of 99.95%** and the system will be kept up-to-date so you can always use the latest version.

JFrog Self-Hosted

When you go the self-hosted route with the JFrog Platform, you'll be responsible for the installation, administration, and management of the platform on whatever infrastructure you decide to use, whether it's in your own data center or on a public cloud platform. It's up to you to keep the deployment up-to-date.

Self-hosting allows for greater control, as you can choose the database, filestore, and other configurations to suit your needs. You can also decide the level of redundancy and implement backup and disaster recovery procedures that are best for you. In addition, going with the self-hosted option means that you are responsible for:

- Infrastructure
- Installation
- Maintenance
- Backup and recovery

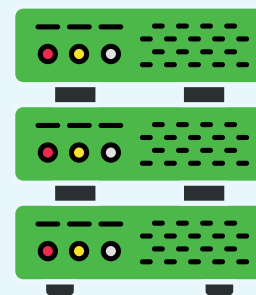
JFrog Cloud (SaaS) vs. Self-Hosted

- Logging (logging assistance available through plugins and extensions)
- Security (compliance assistance available)
- Performance
- Reliability

In certain cases, self-hosting may be necessary for security or compliance reasons, to ensure a higher level of isolation from external networks.

The bottom line is that the choice of a self-hosted or SaaS JFrog Platform deployment should be determined by the needs of your organization. Those requiring more control may prefer the self-hosted option, while those looking for convenience, and cost and time savings may prefer the SaaS model. Both provide the same powerful features and capabilities.

In the next section, we'll explore the reasons we see customers adopting the JFrog SaaS model and moving their DevOps to the cloud.



Why our customers are adopting cloud DevOps

As mentioned before, we've helped hundreds of customers move their DevOps to the cloud. We've noticed commonalities among the reasons they chose to migrate, including:

Innovation

When you have to constantly upgrade your internal platforms, it can be extremely tedious and pull time away from more impactful work. Using JFrog in the cloud, your platform automatically stays up to date, which means you can always leverage all the latest capabilities that JFrog has to offer with no extra effort required.

Security

Software supply chain security is an ever-changing game. As SSC management evolves and new attack methods surface, you need the latest security patches and most advanced security capabilities. The JFrog Platform provides these updates and enhancements, so it benefits you to have continuous access to them in the cloud.

Mission-critical infrastructure

Even just a few years ago, DevOps wasn't as mission-critical as it is now, and multiple days of downtime was a broadly accepted phenomenon. Today, you need your DevOps engine running 24/7, including vulnerability screening and disaster recovery. With JFrog in the cloud, you're always covered.

Strategy

Nobody likes spending time maintaining their DevOps platforms. With a cloud solution, which is managed and maintained externally, you get all that time back so you can focus on the real stuff: strategy, innovation, and the developer experience, among other priorities.

Cost of ownership

When discussing the cost of ownership (TCO), it's important to consider all of the expenses associated with managing and maintaining a distributed platform for supporting your DevOps. This includes direct costs (license, hardware, labor, etc.) as well as indirect costs (productivity, innovation, experience, performance, risk, etc.) Moving to the cloud reduces many of these costs, allowing you to reallocate that budget elsewhere.

Restricted resources

As expectations for technology increase and budgets remain flat – or even shrink – teams need to be able to do more with less. This means expanding the reach of your DevOps platform. You shouldn't have to spend any more money than is necessary on infrastructure. Cloud DevOps frees up the time, space, and resources so you can go further with what you already have.

Common cloud migration roadblocks

Although most technology leaders realize that a move to the cloud is imminent, they hesitate to take the leap for one or more of the following reasons:

Higher costs

It's true that moving to the cloud can be expensive in the short-term, but the long-term savings of cloud-hosted services often outweigh the initial costs. Another way to look at it is to compare one-off vs. recurrent costs. The one-time, upfront cost of migrating your DevOps to the cloud is worth it when compared to the ongoing costs of managing and maintaining your platform on-prem.

High migration effort

Migrating to the cloud can be a lot of work up front, but if handled correctly, the effort is well worth it. The cloud provides a more reliable, secure, and scalable platform for hosting applications, and it grants users access to a wide range of services and features that can help streamline development and operations in the long run.

New operational challenges

As with any new addition to your tech stack, there are always unforeseen challenges to tackle. But there are certain steps you can take to avoid common landmines, which starts with thoroughly researching the cloud service provider before selecting a cloud solution. And because the cloud provides a platform for quickly scaling and deploying applications, it can ultimately help reduce operational

overhead, making it easier to manage and maintain applications, even as you add new ones to your workflows.

Platform dependency

With your DevOps in the cloud, you no longer have to worry about managing and maintaining hardware and software. Instead, the cloud provides a platform for setting up applications in a way that's conveniently agnostic. This means you can focus on developing your applications without worrying about the underlying infrastructure.

Moving your data

Moving data to the cloud can be a daunting task that requires a lot of planning and effort. However, the upfront labor pays off in the long run since the cloud provides a reliable, secure, and scalable platform for quickly deploying and scaling applications. With JFrog, you can utilize existing tools that are purpose-built for cloud migration, as well as leverage JFrog Professional Services to save time and effort.

Keeping them in sync

Keeping data in sync can be a challenge, which is why it's important to work with the right provider and have the right tools in the right places. For example, with JFrog, you never have to worry about keeping data consistent, because it's constantly syncing across instances no matter where they are. In some cases, it makes sense to establish a hybrid set-up (part on-prem, part in the cloud), either as a short or long term migration solution.

Common cloud migration roadblocks

Identity & access management (IAM)

Identity and access management (IAM) is a crucial element of cloud security, and it's normal to have concerns about how to move existing security entities over to the cloud. Fortunately, there are tools that help ensure security measures are consistently applied across instances. For example, JFrog provides Access Federation that allows for the synchronization of identity and access entities between self-hosted and cloud platforms.

Unified release process

You need to guarantee that the same release process you're using on-prem (with all the stages, steps, checks, etc.) is also used in the cloud. To achieve this parity, you need to ensure that wherever you're running your workloads have the same capabilities and configurations.

Compliance and regulations

Compliance and regulations can be a challenge as there are always tons of evolving standards and requirements to meet. It's critical that you keep these the same across your DevOps platforms, whether on-prem, in the cloud, or in multiple clouds. Before committing to a cloud solution, research the cloud providers' ability to provide helpful information you can use to make sure that you have the same security checks and data protection standards on-prem and in the cloud.

As you might have detected, there are a few themes to overcoming these roadblocks:

- 1 Doing research and selecting the right cloud provider for your needs
- 2 Adequate preparation and planning before you migrate
- 3 Exercising patience as you wait to see your efforts pay off
- 4 Defining success criteria and KPIs to measure your success

With these in mind, you'll be well on your way to a steady and successful cloud migration in no time!

Cloud migration strategies

Every company is dealing with a unique set of circumstances, operational needs, challenges, and goals. Here are three different cloud migration strategies you might consider using as a base template for your own migration approach.

1. Lift-and-Shift

Lift-and-shift is a cloud migration strategy that involves taking existing applications and infrastructure from an on-premises environment and moving it to the cloud without making any changes. This strategy is often favored by organizations that are looking to quickly move to the cloud and minimize risk.

2. Replatforming

Replatforming involves taking existing applications and infrastructure from an on-premises environment and making modifications to it in order to take advantage of the cloud environment. This strategy is often favored by organizations that are looking to leverage the scalability and cost savings benefits of the cloud.

3. Refactoring

Refactoring involves taking existing applications and infrastructure from an on-premises environment and completely redesigning and rebuilding them in order to take advantage of the cloud environment. This strategy is often favored by organizations that are looking to significantly improve their applications and infrastructure, and take full advantage of the cloud environment.

A DevOps-centered approach to cloud migration

No matter which cloud migration strategy you apply, we always recommend considering the following DevOps pillars when preparing to migrate to the cloud: CI/CD orchestration, source configuration management, and binary management. Here's how these come into play:

Distributing build workloads

In many cases, it makes sense to leverage the cloud and its capabilities to help automate the process of building the software. So, when migrating your artifact repository to the cloud, it's best to migrate your CI/CD tools and processes as well.

Production workloads

You may opt to keep your build workloads as they are today, and then take the output of those workloads and distribute them across the cloud or multi-clouds. This could span multiple geographic regions, allowing you to take advantage of this environment and maintain harmony between your on-prem and cloud-based systems.

Disaster recovery

To set up disaster recovery, you need a considerable amount of infrastructure. The easiest way to achieve this is in the cloud, as it offers the whole tool set necessary for this purpose.

Cloud migration strategies

Security

When working with external assets, you may decide to keep a portion of them outside your internal network for security purposes. To facilitate this, it makes sense to have this process separated in the cloud. Further, the cloud provides access to a wide range of services and features that can replicate the same security checks on-prem and in the cloud, which is critical for hybrid setups.

What brings all of these patterns together is their focus on how to manage binaries in a hybrid or multi-cloud environment. The bottom line is this: no matter which cloud migration strategy you choose, you need to decide what to do with your binaries.



Cloud migration strategies

It's important to prepare for a cloud migration to ensure the transition is successful and beneficial for your organization. Below are the recommended steps that should be taken prior to your cloud migration.

Step 1: Evaluate Your Workloads

In the discovery phase, take the time to thoroughly assess the workloads that you plan to migrate to the cloud. This includes evaluating their complexity, required performance, and the amount of data that needs to be transferred. Additionally, consider the cost associated with migrating these workloads, as well as the security protocols needed to protect the data during and after the migration process. Getting a clear understanding of the cost and security implications will help you plan a successful cloud migration.

Step 2: Formulate a Cloud Migration Plan

Develop a comprehensive plan for the cloud migration, which should include the timeline, architecture, security and compliance requirements, migration process, and estimated costs. A well-defined plan with clear goals and objectives is essential for a successful transition. Additionally, consider the necessary changes that need to be made to the infrastructure, such as server configuration or data center setup. It's also important to ensure that all stakeholders are aware of the plan and are kept up-to-date with the progress of the migration.

Step 3: Move the Application(s)

Now is the time to move the application(s) to the cloud. This can be done either manually or with the help of automation tools. Make sure that the data is secure during the transfer and that all the necessary components are in place to guarantee a successful migration. It's also important to consider the scalability requirements of the application and ensure that the cloud environment can handle the increased demand.

Step 4: Migrate Data

The next step is to migrate the data from the on-premises environment to the cloud. This includes migrating both structured and unstructured data. Make sure that the data is secure and properly formatted for the cloud environment during the transfer. Additionally, consider the necessary tools that are needed to ensure the accuracy and completeness of the data during the migration process.

Step 5: Configure Networking

Configure the networking for the cloud environment. This includes setting up the necessary access controls, virtual private networks, firewall rules, and other networking connections. Additionally, consider the network bandwidth requirements and allocate the necessary resources for a smooth migration.

Cloud migration strategies

Step 6: Configure Access Controls

Set up the access controls for the cloud environment. This includes creating the appropriate authentication and authorization policies, access control lists, and other security protocols. Additionally, consider the necessary tools and processes that need to be in place to ensure that the access controls are enforced and regularly monitored. Also consider the necessary steps that need to be taken in order to guarantee compliance with any applicable regulations.

Step 7: Assess Production Readiness and Go Live

After the data is migrated and all configurations are completed, it's important to perform the relevant tests to ensure production readiness (i.e. that there's connectivity to all depending solutions, expected load is functioning according to expectations, and other checks).

For more detailed information about how to prepare for migrating to the cloud, be sure to check out our **[cloud migration checklist](#)**.

Migrating to the cloud with JFrog

Practically speaking, when we talk about “migrating to the cloud”, we’re really talking about creating a hybrid coexistence of DevOps platforms across multiple domains that are kept in sync.

The JFrog platform provides this solution. Whether you are using the platform on-prem or in the cloud, the tech stack remains the same. You never need to put in extra effort to keep multiple instances on par regardless of where you’re running your workloads.

Over many years of helping our customers transition to the cloud, we’ve had the opportunity to solve all sorts of scenarios or challenges and improve our own processes along the way. Specifically, there are three areas where we’ve improved by learning from our experience migrating customers:

- **Methodology**

Our professional services team has created a list of nearly 100 tasks that need to be reviewed and completed before migration, including: mapping the existing environment, security controls, environment, planning transition itself, timelines etc. We also ensure that you’re leveraging technology that helps you expedite the migration process without missing a step. Paying attention to these types of details reduces the effort required and greatly minimizes the risk of moving to the cloud.

- **Connectivity**

Not only do we provide end to end TLS support and allowed APs, we also give you private links on all the private clouds to seamlessly and securely connect VPCs to the JFrog Cloud. Additionally, the geo load balancer managed by JFrog (currently available on the self-hosted platform) allows you to use a single URL when you are pointing out to the JFrog services.

- **Data Migration**

A migration tool built on the JFrog CLI is a control plane that provides replication from source to target. This tool deals with everything you need to take care of when migrating your data and configurations from a self-managed instance to JFrog in the cloud. Not only does it take care of all these tasks, it’s fully self-service and can scale up or down. As long as you keep it running, you’re constantly syncing your source (on-prem) to your target (SaaS).

Next steps: walking the walk

It's one thing to support other organizations as they migrate to the cloud and an entirely different thing to do it yourself – which is one of the many reasons JFrog conducted our own cloud migration. We also wanted to streamline R&D efforts, save money on operational costs, focus on innovation rather than maintenance, and replicate the customer experience in JFrog's production environment so we could get ahead of pain points and better serve our customers.

[Read JFrog's cloud migration story](#)

It was this experience, in combination with the many cloud migrations we saw our customers through, that gave us the knowledge throughout this guide. We hope you've gained helpful insights as you consider your own migration to the cloud.

To learn more about JFrog on AWS, GCP, or Azure, visit **<https://jfrog.com/find-a-partner/>**.

If you're ready to take the next step with a free cloud trial, go to **<https://jfrog.com/start-free/>**