What does JFrog do?

I cannot for the life of me understand this company's name

Justin
Mar 1, 2021

9 2

The TL;DR

JFrog provides a bunch of products and services around **DevOps**, i.e. taking your software and deploying it to the world. Their most popular lines deal with **package and dependency management**.

- Software is mostly **developed locally** (on a personal computer), and then later **deployed remotely** to a group of powerful servers to run in "production"
- The process of deploying that software - especially as teams use **more and more open source packages** - can get pretty complex
- JFrog provides products that take your local software and **get it ready** for deployment, **manage** those packages, **scan** for security vulnerabilities, and other nice things

Adoption is comically high, with literally 75% of the Fortune 100 as customers (seems suspicious tbh). JFrog IPOd in 2020, and their market cap at the time of writing is around $5B. So they're definitely on to something.

Refresher: deploying software and packages

Why does JFrog exist, and what problems do they solve? The key is to understand the "DevOps" buzzword, which is what I'm here for. Quoting from the *What's DevOps* post:

> *To understand why DevOps as a concept has gotten really popular recently, you need to understand software delivery and how that's changed over time.*
>
> → ***What it means to deliver software***

When a team of engineers builds an application, it's just a bunch of code. And that code needs to run somewhere for users to access it and get value out of it. During development, developers tend to run that code on their personal computers, but that's obviously not scalable - so when they're ready to share the software with the world, they'll ***put that code on big powerful servers****, and let people access it via a domain name like https://technically.dev.*

So in general, delivering software means taking the application you've built and figuring out how to distribute it widely to whoever wants to use it.

→ ***How things have changed, a lot***

Just 10-15 short years ago, a lot of delivering software meant literally delivering software - Microsoft Office used to ***ship to you on a CD*** *that you'd install directly to your computer. And it wasn't web-based, so you didn't need internet access to use Excel or Powerpoint. The public cloud (AWS and co.) didn't exist, so if you needed to run software on a server or two, you'd need to literally build that infrastructure yourself, which used to cost millions of dollars up front. So naturally, software delivery was bespoke and on the slower side.*

But then people started consuming software via the internet, and public clouds like AWS made it cheap and easy to use a server without having to build a data center. That ***fundamentally changed 3 things****:*

1. *The scale of software increased - software is generally used by a lot more people than in the past. You could realistically need to support millions of users for your application*
2. *Infrastructure got more complicated - we're moving towards increasingly specialized managed infrastructure for different parts of the stack. Generally, you don't just throw your code onto a box and forget about it anymore*
3. *Teams started releasing a lot more often - changes in philosophy mean teams are now shipping code changes to users as often as multiple times a day, which means many many more opportunities to break things*

> *With these fundamental shifts happening, teams needed to start building processes for managing this stuff, and making sure their apps didn't constantly break and disappoint their users. And that's basically what DevOps is - making sure your app works at scale.*

There's one more thing you'll need to know to understand JFrog - **packages**. Today, pretty much all software you use is built on top of existing "packages" of code written by other people, most of which are open source. A few popular examples:

- React, a library for building interactive user interfaces
- Passport, a library for building authentication into Node.js apps
- Tensorflow, a library for building machine learning models

If (more like when) you're using packages like these in your applications, they save you a lot of time and headache - but they also add complexity:

1. **Versioning** - the folks working on React are constantly updating and improving it. Which version did you build *your* software with? Will newer versions of React *break* your software? How do you deal with upgrades?

2. **Packaging** - how do you include the underlying code for React in your application itself? Do you pull it from the web, or store it on your own servers?

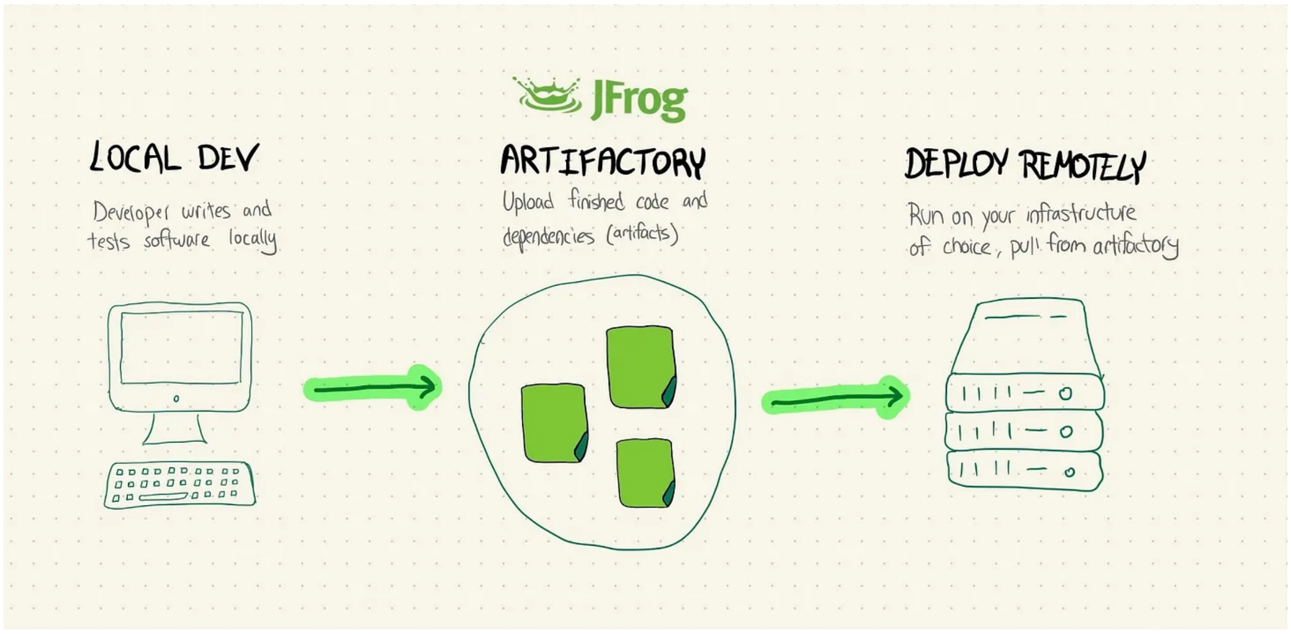
3. **Security** - are the packages you're using secure? If vulnerabilities are discovered, how do you patch them?

You'll also see these packages referred to as **dependencies**, because your software is dependent on them to run.

> 🔍 **Deeper Look** 🔍
>
> If you've heard of NPM or Yarn, these are package managers, and they exist to solve some of the problems we've outlined above. They'll register which packages your project uses, tie things to specific versions, and let you upgrade easily.
>
> 🔍 **Deeper Look** 🔍

The JFrog product suite

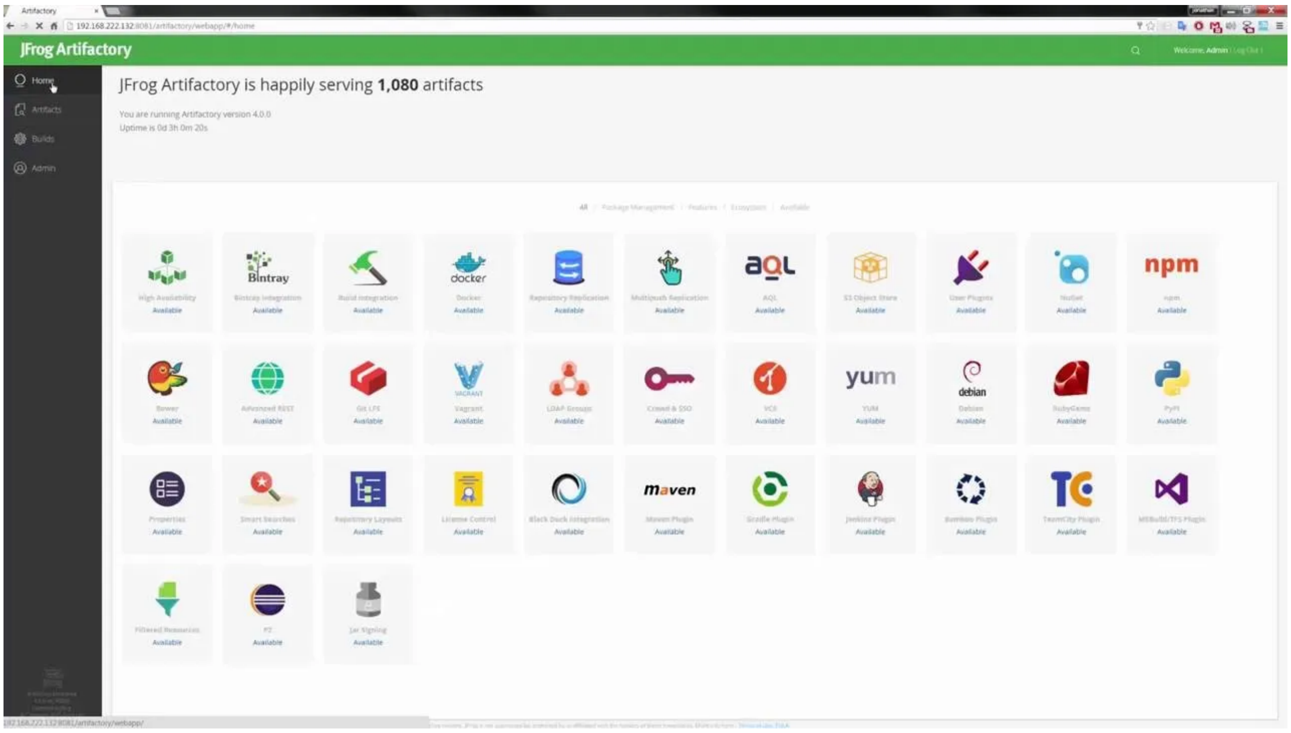
JFrog provides a group of integrated products that attempts to solve all of these DevOps issues together - moving your software from local to remote, managing packages, and building CI/CD pipelines.


JFrog
LOCAL DEV
Developer writes and tests software locally
ARTIFACTORY
Upload finished code and dependencies (artifacts)
DEPLOY REMOTELY
Run on your infrastructure of choice, pull from artifactory

Let's walk through a couple of them:

1. **Artifactory**

This was JFrog's first product (back in 2009!). In software, the word **artifact** refers to pretty much any static group of code (your application, a package). Artifactory lets you upload your code and dependencies (the packages you're using) through a simple REST API, and distribute them to your servers or customers across a globally distributed network (so it's fast).

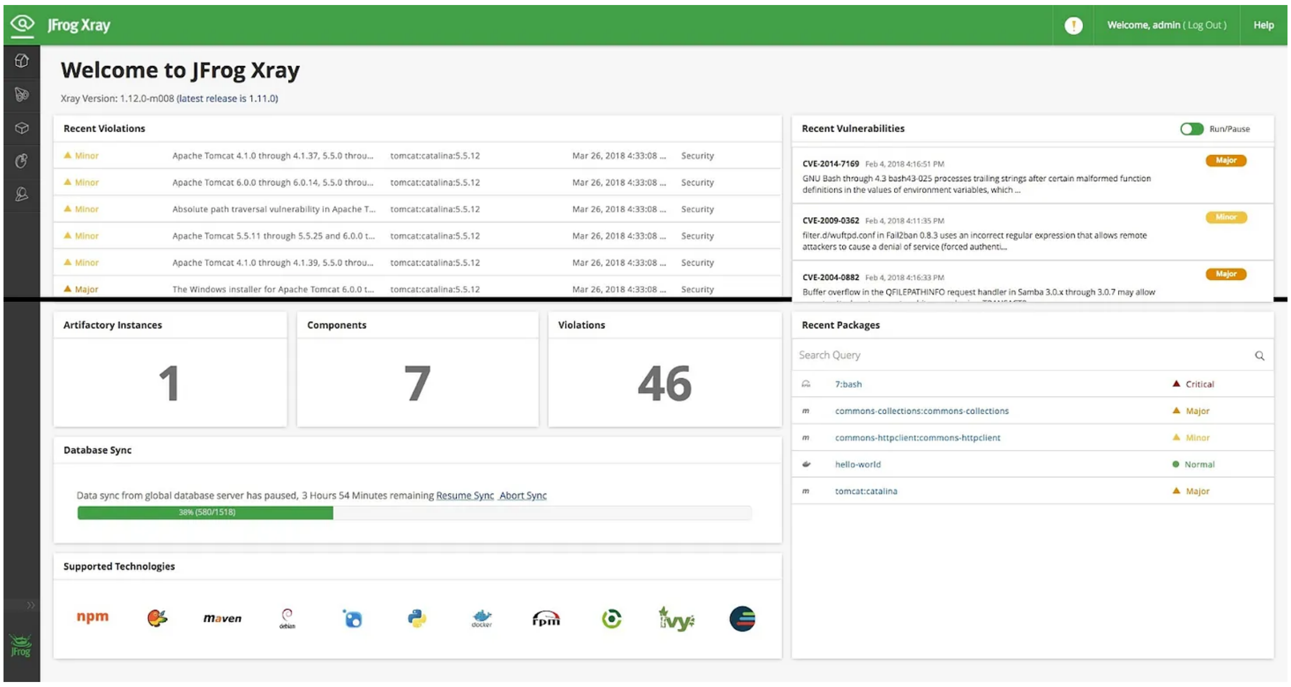

JFrog Artifactory is happily serving **1,080** artifacts

You are running Artifactory version 4.0.0
Uptime is 0d 3h 0m 20s

So if you're using NPM for your Node app and [Helm](#) for your Kubernetes setup, Artifactory sits on top of those and integrates with both. The value prop is having a single place for all of your artifacts instead of using multiple package managers and worrying about how they work together. Gitlab offers [a similar product](#).

2. **XRay**

XRay scans your artifacts (so again, your code and dependencies) for security vulnerabilities and compliance issues.



Welcome to JFrog Xray

Xray Version: 1.12.0-m008 (latest release is 1.11.0)

Recent Violations

Minor	Apache Tomcat 4.1.0 through 4.1.37, 5.5.0 throu...	tomcat:catalina:5.5.12	Mar 26, 2018 4:33:08 ...	Security
Minor	Apache Tomcat 6.0.0 through 6.0.14, 5.5.0 throu...	tomcat:catalina:5.5.12	Mar 26, 2018 4:33:08 ...	Security
Minor	Absolute path traversal vulnerability in Apache T...	tomcat:catalina:5.5.12	Mar 26, 2018 4:33:08 ...	Security
Minor	Apache Tomcat 5.5.11 through 5.5.25 and 6.0.0 t...	tomcat:catalina:5.5.12	Mar 26, 2018 4:33:08 ...	Security
Minor	Apache Tomcat 4.1.0 through 4.1.39, 5.5.0 throu...	tomcat:catalina:5.5.12	Mar 26, 2018 4:33:08 ...	Security
Major	The Windows installer for Apache Tomcat 6.0.0 t...	tomcat:catalina:5.5.12	Mar 26, 2018 4:33:08 ...	Security

Recent Vulnerabilities

CVE-2014-7169 Feb 4, 2018 4:16:51 PM — Major
GNU Bash through 4.3 bash43-025 processes trailing strings after certain malformed function definitions in the values of environment variables, which ...

CVE-2009-0362 Feb 4, 2018 4:11:35 PM — Minor
filter.d/wuftpd.conf in Fail2ban 0.8.3 uses an incorrect regular expression that allows remote attackers to cause a denial of service (forced authenti...

CVE-2004-0882 Feb 4, 2018 4:16:33 PM — Major
Buffer overflow in the QFILEPATHINFO request handler in Samba 3.0.x through 3.0.7 may allow

Artifactory Instances	Components	Violations
1	7	46

Recent Packages

	7:bash	Critical
m	commons-collections:commons-collections	Major
m	commons-httpclient:commons-httpclient	Minor
	hello-world	Normal
m	tomcat:catalina	Major

Database Sync

Data sync from global database server has paused, 3 Hours 54 Minutes remaining Resume Sync Abort Sync

38% (580/1518)

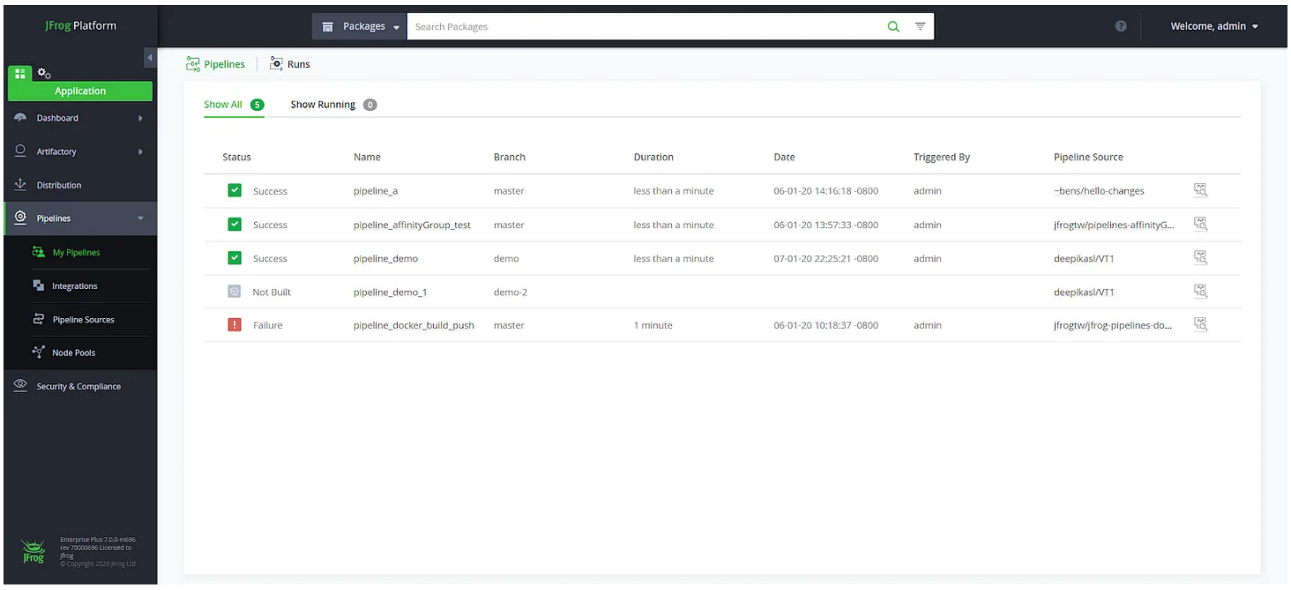
Supported Technologies

npm maven debian docker rpm ivy

Github [sort of does this too](#).

3. **Pipelines**

If you're using something like [CircleCI](#) or [Azure Pipelines](#) to build CI/CD pipelines, you can use JFrog's solution instead and integrate directly with the rest of their ecosystem.



Status	Name	Branch	Duration	Date	Triggered By	Pipeline Source
Success	pipeline_a	master	less than a minute	06-01-20 14:16:18 -0800	admin	~bens/hello-changes
Success	pipeline_affinityGroup_test	master	less than a minute	06-01-20 13:57:33 -0800	admin	jfrogtw/pipelines-affinityG...
Success	pipeline_demo	demo	less than a minute	07-01-20 22:25:21 -0800	admin	deepikasl/VT1
Not Built	pipeline_demo_1	demo-2				deepikasl/VT1
Failure	pipeline_docker_build_push	master	1 minute	06-01-20 10:18:37 -0800	admin	jfrogtw/jfrog-pipelines-do...

Enterprise Plus 7.0.0-m696 rev 70000696 Licensed to JFrog © Copyright 2020 JFrog Ltd

For more on CI/CD, check out [the original Technically post here](#).

There's a lot more to the JFrog product suite, like [Distribution](#), [Mission Control](#), and a [Container Registry](#). Few of these individual products make sense by themselves, and there are strong alternatives in the market for each. But JFrog's marketing and product strategy is all about **integration** - using *all* of these products together gives you a much more powerful experience than using them on their own. Pay attention to their website copy:

> *Universal package repository, SecOps, CI/CD and software distribution all in one platform.*

> *Discover the power of an end-to-end DevOps platform.*

It's all about the "platform" and how it's "end-to-end."

Further reading

- The JFrog S-1 has good contextual information on how they view the business, and more financial info
- Gitlab has a feature comparison between them and JFrog on their site (notice how similar the marketing is)

2 Comments

Write a comment...

Charles Wyman Mar 4, 2021

FWIW what I get from your notes I can't get anywhere else. Incredibly valuable.

1 Reply Collapse •••

1 reply by Justin

1 more comment...