

TOP CI/CD TOOLS

BY DEVELOPERS • FOR DEVELOPERS

You've decided it is time to get on board the DevOps movement and implement continuous integration and continuous deployment of your applications. Or maybe you've been doing this for a while but are looking to explore a new tool. The market is saturated right now with different flavors of CI/CD tooling and it can be overwhelming when trying to choose the right tool for your use case. I know I have been guilty of 'analysis paralysis' when there are too many options on the table. I'm hoping to lend some clarity by providing analysis of some of the most popular CI/CD tools in 2021 and what the best use case is for each. I'm not looking to crown a winner here, but hopefully I can help narrow down your choices.

I am a software engineer with a specialization in DevOps, and I practice CI/CD regularly. I'm hoping to provide a more technical perspective on these tools beyond what you can read on their marketing homepages. My goal is to help you narrow down your choices to a select few products based on your use case. You'll notice that I did not include pricing in these comparisons. This was intentional because I believe the benefits you gain from embracing CI/CD will far outweigh any cost associated with these tools. In addition, the pricing models for these providers can sometimes be complicated and it will be much simpler for you to do the pricing research yourself once you've narrowed it down.

Let's clarify a couple of terms before we get started.

CONTINUOUS INTEGRATION (CI)

This is the act of integrating changes made by multiple developers into a single branch of a code base. CI often includes steps to run tests and ultimately produce a deployable build artifact.

CONTINUOUS DEPLOYMENT (CD)

Continuous Deployment is the act of deploying the build artifacts from the CI step directly into an environment. Some would argue that in order for it to be true CD it would need to go all the way to production. Sometimes people use CD to mean Continuous Delivery, which is the act of being able to deploy a build to production at any time, but not always doing it.

SOFTWARE AS A SERVICE (SAAS)

A software solution that is entirely hosted for you and all you have to do is sign up. In the context of CI/CD tools we will be discussing whether or not a tool is a SaaS product that requires no setup or a self hosted tool that must be maintained.

SWISS ARMY KNIFE SOLUTIONS

SWISS ARMY KNIFE SOLUTIONS

These tools try to solve all of your software development lifecycle needs in a single solution. They contain git repositories, kanban boards, issue tracking, backlogs, and CI/CD pipelines. You can use all of or some of the features of these solutions. You can hook into a git repository hosted elsewhere or leverage Jira for your project management while only using the ci/cd provided by these tools.

AZURE DEVOPS

Azure DevOps is an evolving tool that grew out of Visual Studio Team Services (VSTS). It is a SaaS offering with a self-hosted option that does everything you'd expect a ci/cd tool to do. It has a nice offering of predefined community built pipeline jobs that you can plug and play. There seems to be some oddities related to legacy features from VSTS while Microsoft tries to support those that are making the transition to its new product. Most of this is related to the continuous deployment portion of their service. Much of their deployment documentation is marked as 'classic' and they are transitioning to what Microsoft is calling environments instead. This is a work in progress with new deployment targets being added regularly. Passing artifacts between build steps can also be a bit cumbersome as they need to be 'exported' and 'imported'.

Bottom Line: Azure DevOps is a full featured tool that will meet your needs. However, it doesn't have all the bells and whistles associated with some of the continuous delivery tools. If you and your team are already very familiar with Microsoft products and/or VSTS, Azure DevOps will likely be an easy transition.

SAAS	✓
SELF HOSTED	✓
SELF HOSTED AGENTS	✓
BUILD STEPS IN CONTAINERS	✓
PIPELINES AS CODE	✓
CI	✓
CD	✓

GITLAB

GitLab started with a self hosted solution only but now offers a robust SaaS product as well. GitLab will do all of the things you'd expect a CI tool to do but also has some nice features in the CD space if you've adopted kubernetes. GitLab supports cluster management and canary deployments to kubernetes clusters as part of its CD process. GitLab also has a feature called **Review Apps** that creates environments on demand with each Pull Request. These environments are assigned temporary URLs for access and can be used to validate deployments before they are merged into master.

Bottom Line: GitLab is a full featured CI/CD tool that will likely meet the needs of your organization. You will get more out of it if you have adopted kubernetes but it is not a requirement. Personally, I would choose this over Azure DevOps if I was looking for a swiss army knife like tool unless I was already living in the Azure ecosystem. GitLab is a more mature product and is the core business of the company.

SAAS	✓
SELF HOSTED	✓
SELF HOSTED AGENTS	✓
BUILD STEPS IN CONTAINERS	✓
PIPELINES AS CODE	✓
CI	✓
CD	✓ KUBERNETES ONLY

GENERAL TOOLS

These tools offer basic continuous integration. You will not get any fancy deployment strategies, but if you are early in your continuous integration journey, or you want to develop your deployment strategy yourself, these tools will get the job done.

CIRCLE CI

CircleCI has gained popularity over recent years as a low barrier to entry CI/CD tool. You'll get everything you need here at a reasonable price including pipelines declared as YAML, which seems to be the standard across these tools right now. CircleCI supports continuous deployment with some built in targets for all the major cloud providers as well as Heroku and CloudFoundry. These are fairly limited though and only include a handful of services within these providers.

Bottom Line: Good choice for organizations early in their CI/CD journey. Easy to set up and the tool will grow with you as you mature. You may need to adopt additional tooling as your Continuous Deployment needs become more robust.

SAAS	✓
SELF HOSTED	✓
SELF HOSTED AGENTS	✓
BUILD STEPS IN CONTAINERS	✓
PIPELINES AS CODE	✓
CI	✓
CD	✓

GITHUB ACTIONS

GitHub Actions has gained popularity over recent years as a low barrier to entry CI/CD tool. You'll get everything you need here at a reasonable price including pipelines declared as YAML, which seems to be the standard across these tools right now. CircleCI supports continuous deployment with some built in targets for all the major cloud providers as well as Heroku and CloudFoundry. These are fairly limited though and only include a handful of services within these providers.

Bottom Line: Good choice for organizations early in their CI/CD journey. Easy to set up and the tool will grow with you as you mature. You may need to adopt additional tooling as your Continuous Deployment needs become more robust.

SAAS	✓
SELF HOSTED	✗
SELF HOSTED AGENTS	✓
BUILD STEPS IN CONTAINERS	✓
PIPELINES AS CODE	✓
CI	✓
CD	✓

SELF HOSTED & LEGACY

These tools do not have SaaS models available for their use. This means you will be responsible for setting it up and maintaining the infrastructure that supports it. Consequently these tools have a higher barrier to entry than other tools mentioned in this article.

JENKINS

Jenkins is one of the original CI tools. Originally developed for the JVM, a plethora of plug-ins are now available to build components for almost any language or environment. You will often find it in established development environments. Someone has probably already figured out how to do whatever you are trying to do with Jenkins because of its maturity and large community. This can be a huge benefit if you are new to the tool.

Bottom Line: Jenkins is a bit antiquated and I wouldn't choose it if I was picking a new tool today. It does everything you need from a CI tool but I believe the other products here are better and I would personally lean toward a SaaS model unless you need to manage things in house for some reason.

SAAS	✗
SELF HOSTED	✓
SELF HOSTED AGENTS	✓
BUILD STEPS IN CONTAINERS	✓
PIPELINES AS CODE	✓
CI	✓
CD	✗

TEAMCITY

TeamCity is an older, self hosted, tool from JetBrains. Generally I'm a big fan of JetBrains products, especially their IDE's, but I would not recommend TeamCity for an organization newly adopting a CI/CD tool. It gets the job done like all of the tools here but its user interface takes some time to wrap your head around. In addition, the self hosting only option makes the barrier to entry pretty high. My biggest gripe with this tool may be a personal one. The pipelines as code feature is not YAML like every other tool out there. It used to be XML but as of recently you can now write your pipelines as Kotlin. However, this feature still feels like it was bolted on and most will use the user interface instead.

Bottom Line: I believe there are better options for your organization than TeamCity as a new build tool. I wanted to mention it in this article though as it still maintains a sizable market share in the space.

SAAS	✗
SELF HOSTED	✓
SELF HOSTED AGENTS	✓
BUILD STEPS IN CONTAINERS	✓
PIPELINES AS CODE	✓
CI	✓
CD	✗

KUBERNETES NATIVE

The adoption rate of kubernetes is undeniable at this point and as adoption grows we are seeing more and more tools emerge to run exclusively on kubernetes. Another interesting pattern that has emerged with these tools is the concept of 'GitOps', which involves pulling deployment changes into a cluster based on the state of existing git repos vs pushing changes to the cluster via traditional pipelines. The goal here is that the current state of your workloads is always defined in source control, and the tool will work to match the current workload state to the state defined in your repository.

ARGO

Argo is a newer tool that has grown up in the kubernetes ecosystem. It's first release was in late 2018 and built on the concepts introduced by **Weaveworks** around GitOps. Argo does not replace existing continuous integration tools for building your artifacts, it focuses exclusively on the deployment phase of your pipeline. Argo is hosted inside your kubernetes cluster and is installed via a typical kubectl apply command that will create custom resource definitions. Argo can then be configured to watch git repositories you tell it to and sync the configuration state of those repositories into your cluster. A typical build pipeline would involve traditional CI that pushes an image to a registry. Argo advocates for storing deployment configuration in a **separate repository** so once the image is created the deployment config repo would need to be updated with the latest image. This could be done manually or another build step could run to make a commit to the repository. Argo would see this change and automatically deploy the change into the cluster. It is also worth keeping an eye on **Flux** as an alternative to Argo if you like the GitOps model. However, Fluxv2 is currently under development and in my opinion not as mature as Argo yet.

Bottom Line: Argo is an awesome tool for Continuous Delivery and definitely worth a pilot if you have adopted kubernetes and are interested in the GitOps pattern.

SAAS	✗
SELF HOSTED	✓
SELF HOSTED AGENTS	✓
BUILD STEPS IN CONTAINERS	✓
PIPELINES AS CODE	✓
CI	✗
CD	✓

JENKINS-X

Jenkins-X is a deployment CI/CD solution that is completely disconnected from its predecessor Jenkins. The Jenkins-X team has gone all in on Kubernetes and it is a requirement to run the tool. Recently, **version 3** of the tool was released with some welcome improvements like support for **Helm 3**, **Kustomize**, and Secrets. In addition version 3 supports a cluster per environment (dev, staging, production) as opposed to requiring a single cluster that uses namespaces for environment separation. This is a welcome change as it allows for better isolation between environments. Jenkins-X offers a very opinionated solution for creating pipelines by leveraging open source tools like **Tekton**. This is a self hosted solution that may leave you needing to debug portions of other open source tools when things aren't working properly.

Bottom Line: Jenkins-X is a very opinionated tool and is worth a look if you have adopted Kubernetes. I believe it is still maturing and I find that it is too opinionated and 'black boxy' for my liking but your mileage may vary.

SAAS	✗
SELF HOSTED	✓
SELF HOSTED AGENTS	✓
BUILD STEPS IN CONTAINERS	✓
PIPELINES AS CODE	✓
CI	✓
CD	✓

CI/CD + HOSTING

NETLIFY

Netlify is free for small projects. Enterprise can be expensive. Netlify focuses on the front end developer that just wants their code running in production. Netlify will build your static sites and Single Page Applications and host them on their platform. This is a huge selling point as you don't have to think about infrastructure at all. Netlify provides a Content Delivery Network (CDN) as part of their service as well, making the delivery of your applications blazing fast all over the world. Netlify also offers a preview environments feature that is very handy for testing changes before they are merged into your trunk branch. Environment management can be a bit difficult but Netlify has a feature called branch subdomains that makes this a little more bearable as long as you are willing to devote a branch to each long standing environment you want to use. Finally, Netlify will generate and auto renew SSL certificates for you through an integration with LetsEncrypt, which is awesome!

Bottom Line: Netlify is a great choice if you want to adopt some DevOps practices but want a low barrier to entry. You don't need a lot of infrastructure or shell scripting knowledge to take advantage of their offering. If you can create build scripts with npm you'll be good to go. If you want to get more sophisticated with your Continuous Deployment strategy you may outgrow this tool.

SAAS	✓
SELF HOSTED	✓
SELF HOSTED AGENTS	✓
BUILD STEPS IN CONTAINERS	✓
PIPELINES AS CODE	✓
CI	✓
CD	✓

CONTINUOUS DEPLOYMENT

The tools in this section assume you have the Continuous Integration part figured out and are happy with your tooling. The focus of these tools is to enable more complex and sophisticated Continuous Deployment strategies.

SPINNAKER

Spinnaker is Netflix's Continuous Deployment tool and has been released to the open source community as of 2015. Netflix uses it for managing the infrastructure and deployment life cycles of its applications, not for continuous integration and building. Because of that it offers the most capabilities around deployment. Spinnaker has a lot of its own language for managing these lifecycles and has a large learning curve. It offers a multi-cloud solution with a deployment target list that includes the major cloud providers, Kubernetes, and Cloud Foundry. Spinnaker shines in situations where complex deployment scenarios are needed such as multi-region, multi-cloud, hybrid-cloud, blue green, and canary. Pipelines as code is a grey area here as Spinnaker has made strides in recent years with Pipeline Templates but is still working through adoption.

Bottom Line: While this product is awesome and has a lot to offer, this level of sophistication is for very mature organizations and tech giants that have complex deployments. You do not need this for simple websites or smaller application stacks. If you are looking for a mature tool to handle your deployments and don't mind self hosting this is worth a look.

SAAS	✗
SELF HOSTED	✓
SELF HOSTED AGENTS	✓
BUILD STEPS IN CONTAINERS	N/A
PIPELINES AS CODE	✓
CI	✗
CD	✓

HARNESS.IO

Harness is in the same category as Spinnaker, continuous delivery as a service, but seems to be a slightly more simplified offering while offering a SaaS solution. Harness integrates tightly with newer methodologies like containerization and serverless, making the process a bit more seamless than spinnaker. Harness allows for the complex deployment strategies like canary, blue/green, etc. Not much is offered in the way of Continuous Integration as you are expected to be running a Jenkins server or using bamboo if you want to integrate your build process into Harness. Otherwise you can use any Continuous Integration tool you want and simply leverage Harness for the Continuous Deployment. Harness offers deployment targets across the major cloud providers, Kubernetes, and Cloud Foundry and supports pipelines as code.

Bottom Line: Harness is the newer, shinier, simpler version of Spinnaker that requires less ramp up time for your team. It's SaaS offering is a major selling point, although they do offer an on-prem, self managed deployment as well. If you are considering Spinnaker it is worth evaluating Harness as well.

SAAS	✓
SELF HOSTED	✓
SELF HOSTED AGENTS	✓
BUILD STEPS IN CONTAINERS	N/A
PIPELINES AS CODE	✓
CI	✗
CD	✓

HONORABLE MENTIONS

The below tools don't fit any use cases the other products don't so I've chosen to leave them off the list. I believe one of the above tools would be a better fit for your organization due to ease of use, maturity, or feature set.

- **Travis**
- **Codefresh**
- **Drone.io**
- **Bamboo**
- **Tekton**



CLOSING THOUGHTS

At the end of the day all these tools are exactly that, tools. The success of your organization's adoption of CI/CD is not going to be significantly hindered by your choice here. Unless you choose a Kubernetes only tool and your organization doesn't use Kubernetes. It is much more important that you adopt the cultural changes that CI/CD brings, along with other [DevOps philosophies](#). My main consideration for choosing a tool would be based on my organization's DevOps maturity, and any existing familiarity with these tools among my software team. You can't go wrong and if somehow you do you can always change it. That's the great part about software development, we aren't building material structures like houses with large upfront and change costs.

Happy Pipelining!



AUTHOR

Dillon Courts

Cloud Practice Lead, Callibrity

Dillon has 10 years of professional development experience and a passion for building reliable, highly scalable software systems. He specializes in DevOps practices, Cloud Native solutions, and microservice architectures. In his free time Dillon enjoys playing sports, video games, board games, and spending time with his wife Alex, and 2 dogs.



**WE ARE ARTISTS.
WE ARE ENGINEERS.
WE ARE INNOVATORS.
WE ARE CALLIBRITY.**

Learn more about Callibrity
www.Callibrity.com

ABOUT CALLIBRITY

Callibrity is a software consultancy specializing in software engineering, digital transformation, cloud strategy, and data-driven insights. Our national reach serves clients on their digital journey to solve complex problems and create innovative solutions for ever-changing business models. Our technology experience covers a diverse set of industries with a focus on middle-market and enterprise companies.

Callibrity (kə'librətē) is a mashup of two different roots, calli and caliber. Calli means 'beautiful' in Greek, as in Calligraphy - beautiful writing. Caliber means 'a degree of merit or excellence'. We strive to do beautiful work with a high degree of merit and excellence.

< by **developers**
for **developers** >