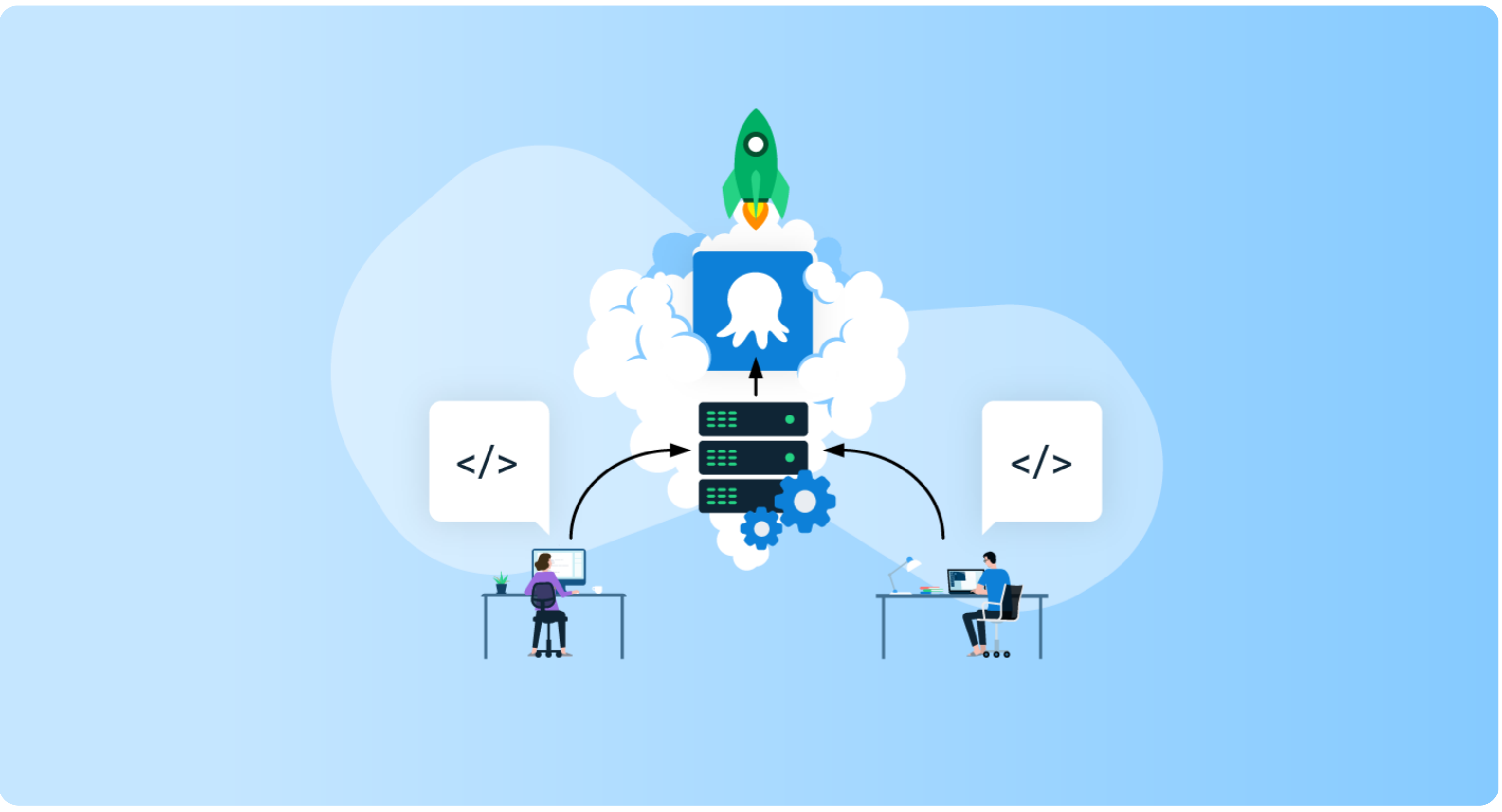


What is Everything as Code?

Terence Wong
September 13, 2022 • 4 mins

If you work in DevOps or cloud, you've probably worked with tools like GitHub Actions, Jenkins, or Terraform to deliver your DevOps pipelines. You may have noticed these tools all represent parts of the DevOps pipeline as code, letting you store and reuse parts of the pipeline later.

The code representation of the DevOps pipeline is part of a shift to Everything as Code (EaC). But what is EaC, and why is it important?

Everything as Code is an approach to software development and DevOps that uses code to define and manage IT resources. The code representation of resources makes it easier for developers to:

- Audit changes
- Improve consistency
- Scale resources
- Transfer settings from one environment to another

Taken literally, EaC is an ideal state where every part of the software lifecycle is code.

The implementation of EaC today is far from that ideal, though, with EaC used as an umbrella term to cover specific applications of the "as code" framework. Infrastructure as Code (IaC) and Configuration as Code (Config as Code) are widespread EaC applications, with other applications covering a range of IT fields.

In this post, I discuss some applications of Everything as Code, the benefits, and my thoughts on moving towards EaC.

Infrastructure as Code

Infrastructure as Code (IaC) is the process of provisioning IT infrastructure through code. Traditionally, developers provisioned IT infrastructure manually with no control over who could provision infrastructure and what methods they used.

You may have encountered a situation where one person, the gatekeeper, controlled all your IT infrastructure. Only that person understood where things were, but even they weren't sure about everything.

With IaC, the contents and location of all infrastructure are visible to everyone in the organization. Infrastructure is no longer controlled by a few people but is version-controlled in a configuration file. Much better!

Terraform is one of the most popular IaC frameworks. Terraform provides a configuration language developers use to define cloud and on-premises resources. Many cloud providers use IaC so developers can provision infrastructure on-demand in a repeatable way. If you can learn the Terraform language, you can use it with any cloud provider or deployment tool, like Octopus, to provision infrastructure.

Configuration as Code

Config as Code is the process of capturing all system configuration settings in code. In Octopus, the configuration settings specify the deployment process. We released Config as Code for Octopus in March 2022.

We designed Config as Code to deliver the power of Git (branches, pull requests, and a complete audit log of changes) with the usability of the Octopus UI. You can read about some of our design thinking in our post Shaping Config as Code.

In Octopus Deploy, Config as Code users can choose to use the UI or the source-controlled implementation without losing any functionality. You might choose to use the UI to make minor changes or source-control to make advanced changes.

We believe our no-compromise solution is one of the best "as code" implementations.

Other examples of Everything as Code

There are other examples of EaC, some more niche than others, such as:

- **Environments as Code:** Tools that provision computing environments like Docker and Vagrant. Many cloud providers have their own compute Environments as Code offerings, such as Compute Engine by Google and EC2 by Amazon.

- **Data Analytics as Code:** You can represent data pipelines and machine learning processes as code. Data pipelines as code allow data scientists to port data analytics components from one project to another.

- **DevOps Pipelines as Code:** Tools like GitHub Actions and Jenkins represent DevOps Pipelines as Code. When a developer pushes a code change, a process is triggered to build the repository and produce an artifact or deploy a release.

- **Security as Code:** If you're managing a cloud environment, you'll be concerned about security. Security as Code lets you represent security data such as roles and permissions in a configuration file. If sensitive data needs to be "as code", you should consider encryption techniques.

EaC is relevant in any scenario where developers can code processes and resources, so it applies to many more segments of IT. Can you think of any EaC applications in your business?

Benefits

Everything as Code lets you express IT resources as code. The main benefits are:

- **Consistency:** You can capture infrastructure and configuration settings in a standard framework such as Terraform. This framework reduces human error and improves reliability as the system is version-controlled.
- **Scalability:** If you want to scale, you make a small change to a configuration file, and you can roll back any issues to previous versions.
- **Portability:** You can export your infrastructure, configuration, or other system parts and replicate them.
- **Auditability:** You can audit your systems more easily as version control makes changes visible.

EaC has some clear benefits for you and your software stack, but it doesn't always make sense to move a system towards EaC.

While deployments as code in Octopus Deploy have significant benefits, there are diminishing returns on converting other platform parts to EaC. There's always an opportunity cost in developing an additional EaC feature compared with other features that organizations could develop in the same timeframe. In practice, organizations should apply EaC where it makes sense and has the most benefit to users.

Conclusion

Everything as Code (EaC) is an approach to software development and DevOps that uses code to define and manage IT resources. EaC has found many applications in Infrastructure as Code, Config as Code, and other areas of IT. If you work in DevOps and cloud, you've likely already seen benefits of EaC firsthand.

Although Everything as Code is a promising end-state for organizations, there's an opportunity cost to convert parts of a platform to EaC, which will inform where you invest your resources. There will undoubtedly be parts of your platform that could benefit from an EaC approach, and the key is identifying those areas.

Happy deployments!

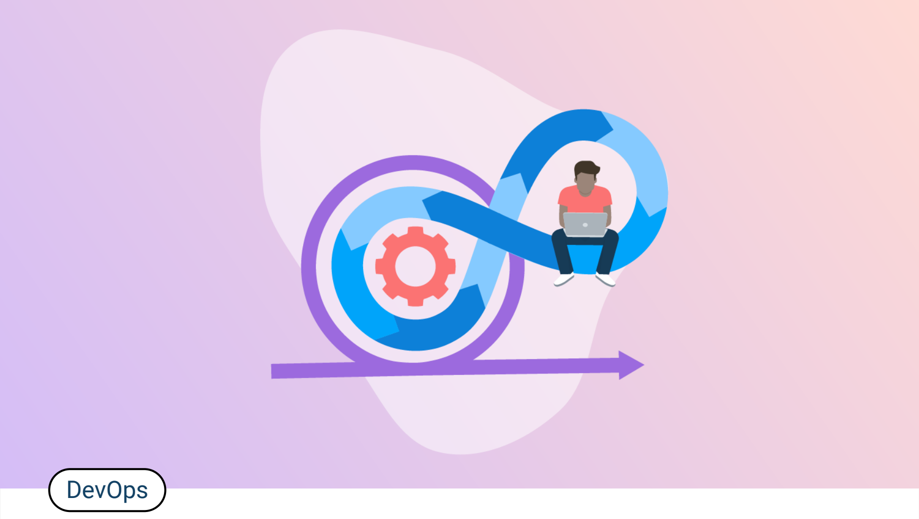
Tagged with: DevOps Cloud Orchestration Configuration as Code

Related posts

DevOps

Common mistakes in DevOps metrics

Steve Fenton
December 7, 2022 • 6 mins



DevOps

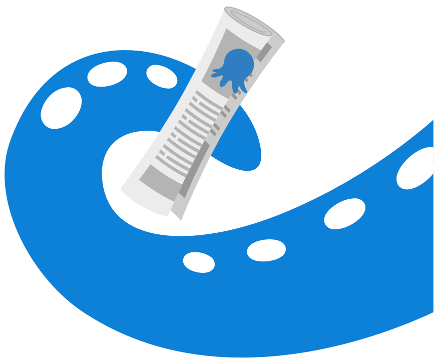
Comparing Lean, Agile, and Continuous Delivery

Steve Fenton
December 5, 2022 • 6 mins

DevOps

Best practices for CI/CD

Newsletter



Logged in as Terence Wong (terence.wong@octopus.com)

Join ~48,000 DevOps professionals and sign up for the latest Octopus news, events, and opinions. No spam. Unsubscribe at any time.

Subscribe

Your privacy is important to us. Read more in our Privacy Policy.