Best practices for CI/CD

Terence Wong
November 30, 2022 • 5 mins

Continuous Integration and Delivery (CI/CD) take software development from code to a live product. CI/CD forms part of DevOps processes, with many commonly agreed-upon best practices you can follow to improve your deployment pipeline.

If you work in DevOps, you've probably used a build server like Jenkins and a deployment tool like Octopus Deploy to complete your deployment process. Octopus supports the Continuous Delivery side of CI/CD, providing a best-in-category product that makes complex deployments easier.

At Octopus, we believe in the power of 8. An octopus has 8 limbs, so here are 8 best practices to help your deployment journey.

You can also learn more in our DevOps engineer's handbook.

Adopt Agile methodologies

Agile methodologies are vital to CI/CD and DevOps. Agile is a project management approach involving continuous collaboration with stakeholders and continuous improvement at each stage of the deployment process.

The principle of Agile is having frequent feedback through small development iterations so developers can closely align the final product with the user's needs. Agile methodologies contrast traditional waterfall methods, where projects are scoped and delivered in a single phase.

We recommend software projects are managed according to Agile and Lean principles so the continuous feedback loop can improve the product. We've seen Agile implemented as a checklist for upper management to tick off. In the initial stages, software teams apply Agile to meet the checklist. As teams have permission to explore the Agile space, they start to see the real benefits.

Use version-controlled code, connected to the deployment process, committed frequently

If you work in software, you've almost certainly used Git. The wars on source-controlled code have been fought and won, and Git is now synonymous with source control.

Source-controlled code allows a complete history and rollback of code to previous versions. You can also resolve conflicts by using Git's merging methods.

Committing a code change should trigger a CI/CD pipeline build. This trigger allows developers to test and validate changes to the codebase earlier. After a code change is set up to trigger an automated build, developers should be encouraged to commit their code at least once a day. Daily commits trigger automated tests more frequently so developers notice any errors sooner.

Use Configuration as Code for your deployment process

Config as Code represents your deployment process in a Git-based system. Deployments inherit all the benefits of Git, such as branching, version control, and approvals as pull requests.

Config as Code lets you store your deployment process in Git. You can test changes to deployments in a branch and validate them through a pull request. Git-based deployments make it easier to transfer a deployment set up from one environment to another.

In 2022 Q1, we released Config as Code for Octopus Deploy, and believe we set an industry standard. Other Config as Code solutions sacrifice usability for functionality. In Octopus, you get all the features of Config as Code, whether you use the UI or the version-controlled implementation.

Choose a tool that lets you keep builds green

A green build in a CI/CD pipeline means that every test passed, and the release has progressed to the next stage. Software teams aim to keep builds green.

You should choose a deployment tool that surfaces information to help keep builds green. Many deployment processes only use a build server that pushes releases into production. In practice, only using a build server makes it harder to manage a release between different deployment stages. Using a dedicated deployment tool gives you a dedicated management layer to keep builds green.

A build server doesn't include the concept of deployment stages. Octopus Deploy, however, separates a release into Test, Dev, and Production environments, and environments can exist at different release versions in each stage. Our UI shows each release's deployment stage and transitions releases between stages. The Octopus UI also shows logs and error messages to help developers quickly identify failing builds.

Continuously automate your tests

Testing code changes is essential to producing reliable releases. The testing suite should cover all use cases for the product, from functional to non-functional tests. These tests should be automated so that a code change can trigger an automated test and build. Automated tests improve the agility of a software development project to get releases live faster.

A survey by Mabel on the state of testing in DevOps indicates that automated testing (at least 4 to 5 different types of tests) is key to customer happiness. In the 2021 State of DevOps DORA Report, continuous testing is an indicator of success. Elite performers who meet their reliability targets are 3.7 times more likely to leverage continuous testing.

Strengthen the feedback loop through monitoring

Developers use telemetry data (logs, metrics, and traces) to understand their system's internal state. Telemetry unlocks observability so developers can act on data to fix their system. When you have telemetry data, you can use observability tools to add monitoring capabilities to your system.

Monitoring key system metrics can help diagnose a system for vulnerabilities and identify improvements. In the DevOps community, DORA metrics are commonly accepted as crucial metrics for the success of a deployment pipeline.

Octopus lets you measure results, compare project statuses, and continuously improve with DevOps Insights focused on the DORA metrics.

Use technologies that are fit-for-purpose

Every year, there are a new technologies that people claim will revolutionize the IT playing field. Whether it's containerization, machine learning, or blockchain, some technologies change the playing field, while others are too immature to make a real impact. When managing a CI/CD pipeline, it's essential to only choose technologies fit-for-purpose.

While being cloud-first makes sense for some parts, forcing everything onto the cloud might not be the right solution. Adoption of new technologies can bring significant improvements, but taking a measured approach avoids unnecessary pain when the costs of adoption outweigh the benefits.

Take security seriously

As software projects get larger, the security risks increase with more data handling, users, and dependencies. Your deployment process should have a security strategy.

Many cloud providers, like AWS, Azure, and Google, have built-in security features such as IAM, secrets, and role-based permissions. You can use these features to manage some security concerns.

Customers are increasingly concerned with security, and companies need to invest in certifications such as ISO 27001 and SOC II to certify their compliance with security regulations.

On May 12, 2021, The US government released Executive Order 14028, "Improving the Nation's Cybersecurity". The Order requires all vendors of government software projects to produce a Software Bill of Materials (SBOM). The SBOMs detail all software components so that governments can screen software for cybersecurity. If you want an example of how to produce an SBOM and attach it to your deployment process, we created a free tool called the Octopus Workflow Builder that can help.

Conclusion

CI/CD is part of the DevOps model and helps bring software projects from code to customers. If you work in DevOps and implement CI/CD, you should follow industry-standard best practices for your pipeline. To help, this post covered 8 best practices you can use to make the most of CI/CD.

Many tools can help you with CI/CD, from build servers and deployment tools to monitoring solutions. Octopus Deploy fits into CI/CD as a Continuous Deployment solution making complex deployments easier.

To learn even more about best practices for CI/CD, check out our DevOps engineer's handbook.

Happy deployments!

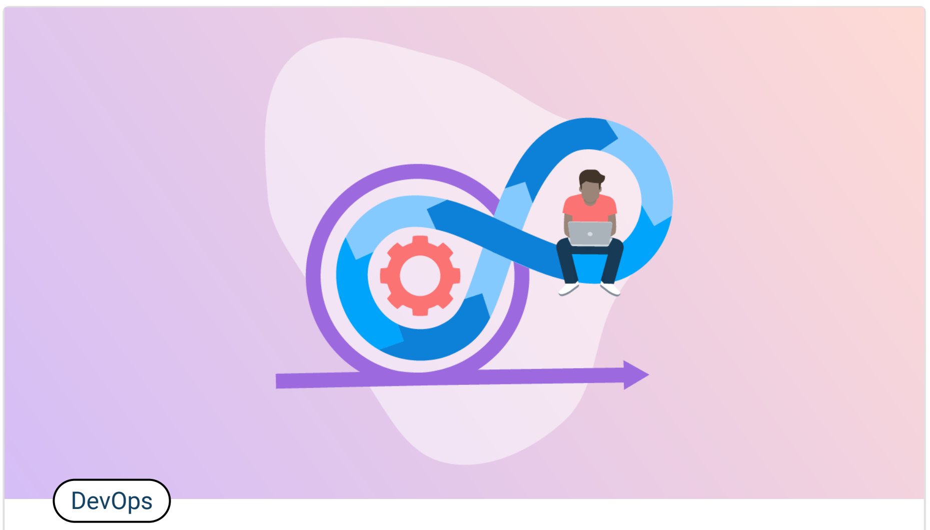
Tagged with: DevOps Continuous Integration Continuous Deployment Continuous Delivery

Related posts

DevOps

Common mistakes in DevOps metrics

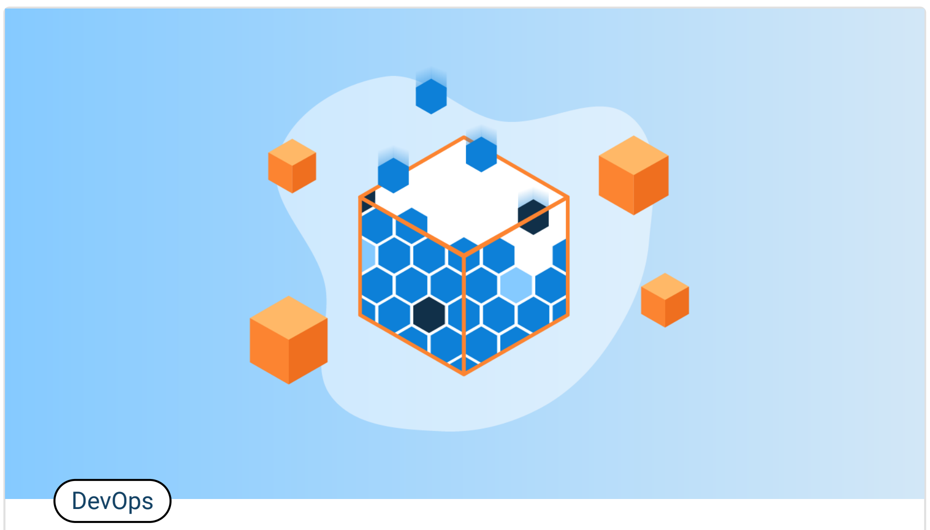
Steve Fenton
December 7, 2022 • 6 mins



DevOps

Comparing Lean, Agile, and Continuous Delivery

Steve Fenton
December 5, 2022 • 6 mins



DevOps

Granting federated user accounts to an EKS cluster

Shawn Sesna
November 28, 2022 • 4 mins

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.