The 13 Commandments of DevOps

Today, DevOps is a trending buzzword in the IT landscape. Rather than being a skill description for a position you can hire in your IT department or a specific tool you can purchase and install to start being "DevOps-compliant", DevOps is, above all, a cultural-shifting paradigm that impacts the main pillars of an organization (People, Processes and Tools.) Its ultimate goal is adding increased value to the customer and enabling organizations to react faster to change in the business.

Follow these principles and pratices and you will be much closer to achieving a DevOps-focused mindset in your organization.

Plan

Validate impact upfront

Define the expected impact of developing an application or feature upfront. Consider how this will affect areas such as network configuration, security, and architecture. For instance, accounting for the growth of concurrent users or transaction count ensures that your infrastructure capacity is fit for the workload increase.

Immediate provisioning

Automating your infrastructure provisioning activities will definitely boost your development and testing speed. Technologies such as Infrastructure as Code on top of virtualized servers or cloud providers can be a huge help.

Anticipate failure

Having a disaster recovery plan is not enough to avoid disaster. Practice failure scenarios on top of a self-healing infrastructure. Plan for simulations during development cycles, as well as in production during off-peak hours, to guarantee the effectiveness of your disaster recovery plan.

Develop and Test

Develop based on production

Mimic production as closely as possible. Validate your apps against production-specific constraints early in development rather than dealing with them after the rollout. Bootstrap production data, simulate integrations, and test on target user devices.

Continuously integrate your code

Regularly enforce a continuous integration approach and enforce the consistency of ongoing developments over the existing codebase. Run automated test scripts to pinpoint any breaking changes for developers to act upon them earlier in the development process.

Trace all changes

Use a tool to manage the full lifecycle of anything that impacts your applications, from defects up to business requirements. Tracking every action performed along the way lets you have full traceability from origin, all the way down to the application version that implemented it.

Register your configurations

Register all development configuration activities in a centralized shared repository. This ensures they are later applied in the deployment process to subsequent environments. Don't let these changes be forgotten only to be recalled when operational errors start piling up.

Deploy

Orchestrate deployment process

Create a deployment script where you define all steps of the deployment process for each application, including rollback procedures. Apply it in pre-production environments and validate the deployment process to ensure reliability and consistent behavior when your application reaches production. Automate your script execution to increase your efficiency even further.

Compress delivery cycles

The sooner you reach production, the sooner you can start adding value to the business. Compressing delivery cycles through process automation and scope minimization will help you increase the speed of delivery of new features with reduced risk.

Measure everything

Measure your apps' impact on your business and customer base once they reach production. Extract metrics from multiple sources and build dashboards with a real-time picture of your application activity and customer experience. Do it also while in development to identify potential risks and optimization points in advance.

Monitor

Identify trends and deviations

Leverage the metrics collected from your applications and infrastructure to establish a baseline pattern for your applications. Identify deviations from the expected behavior and proactively address issues to reduce response times.

Amplify feedback loops

Come full circle by impacting the focus and priorities of each stakeholder. Make sure all collected feedback is shared among everyone in the organization and incorporated back into the development pipeline.

Collaborate! Break the silos!

Ultimately, DevOps is all about bridging the gap between development and operations. Promote a continuous collaboration effort through informal channels, such as ChatOps, in order to add value to the business and quickly respond to change.

So go ahead and tear down that wall!

outsystems®