

Setting Up IoT DevOps for Success with Advanced Simulation

DevOps has become a highly sought-after methodology in IoT projects because of its promise to shorten product development and enable iterative product updates. But not every project is necessarily the right fit for DevOps—especially when you're dealing with mission-critical systems.

Much of the problem comes down to testing. The DevOps approach aims to accelerate feedback loops among development, quality assurance (QA), and production teams. This is achieved in part by replacing the traditional development and QA process with automated QA as changes are deployed into the production environment.

But [any test failures that come up still require timely detection and mitigation](#) such as backing out of the latest changes or releases. While this can be a manageable trade-off for non-critical systems, it is often an unacceptable risk for mission-critical ones.

Using Simulation for DevOps QA

One alternative is to run advanced simulations of mission-critical systems that allow QA engineers to apply changes and run regression tests in the lab before ever deploying production units.

To a large extent, this is already done today. “For non-critical systems such as software-as-a-service (SaaS) offerings, free trial versions can be used as a lab of sorts, with the end users essentially performing QA,” notes [Uwe Zimmermann, CTO at Gambit Communications, a network and IoT simulation tool provider](#). But mission-critical systems require far more robust testing.

The challenge is to simulate the production system to a sufficient degree to detect problems early. IoT systems are often built from many subsystems—often from third parties—creating dependencies that are hard to predict. Therefore, any change in one component may have a ripple effect on the whole system, making it essential to perform frequent automated regression testing of the entire system.

Another challenge is the dynamic nature of IoT systems. As more nodes are added to an IoT network, some parts of the network may develop bottlenecks faster than others. This can lead to non-linear performance degradation that requires careful capacity planning and monitoring.

Not only does a simulator need to address these issues, it needs to do so quickly. The shorter feedback loops that characterize DevOps leave less time for testing, so the question becomes: How fast can the IoT simulator be reconfigured to run through regression tests in the allotted time?

In addition, DevOps brings together previously siloed teams, meaning the “walls” between development and QA teams are broken down, allowing them to collaborate on test plans. While this can speed development, it creates new pressures. “Removing these barriers means you have more fingers in the pie, and that can create problems,” Zimmermann explains. To keep work flowing, the simulator must fit into the DevOps pipeline and provide results that can be used by all stakeholders.

Building a Better Simulator for Mission-Critical DevOps

To meet these needs, Zimmermann says a simulator must excel in four areas:

- **Completeness:** A simulator must cover all aspects of the production environment that need testing. This includes being able to mimic complex scenarios at scale, such as high traffic, network failures, or security breaches. A simulator should also be able to reproduce pathological crisis scenarios that are rare but could have severe consequences.
- **Speed:** Real-world scenarios might take days, weeks, or months to unfold. Testers need to reproduce these scenarios in a matter of minutes or hours—which means the simulator must have high performance and scalability. The simulator must also be able to cope with potential growth in device counts and unexpected spikes in demand.
- **Flexibility:** A good simulator must be able to adapt to any automated workflow pipeline. The tool needs to have flexible APIs and support numerous programming languages. What's more, the simulator should have the flexibility to adapt to the rapidly changing technology that is endemic to IoT applications.
- **Determinism:** The simulation needs to reproduce problems, so that multiple teams can investigate and solve them together. The collaboration that is fundamental to DevOps gives people with different skill sets and system knowledge access to the code. Each change needs to be reviewed, verified, and tested.

An IoT Simulator Designed to Tackle the Challenge

As a result, Gambit Communications developed the [MIMIC Simulator specifically to develop and test large-scale networked applications](#). Long used for network applications by large organizations like Cisco, IBM, HP, and Intel, the tool is seeing increased adoption for IoT applications as well.

At its core, MIMIC Simulator aims to capture realistic interactions with the system under test. Typically, creating a simulation (the “model”) in MIMIC Simulator involves recording the interaction of a real device (or environment) with the system under test. From here, MIMIC Simulator tools extrapolate a basic simulation from that recording.

That basic simulation can then be customized to achieve desired effects, such as exhaustive test cases, boundary conditions, scaling, error handling, etc. Each scenario simulates a specific condition, and MIMIC Simulator can run through any number of scenarios, for example, in regression test suites.

“Simulations are successful if the system cannot tell the difference between the simulation and the real environment,” says Zimmermann.

The MIMIC Simulator is notable for scaling nearly linearly with CPU power. The tool can simulate up to 100,000 devices from a single workstation and upwards of 1 million devices in a multi-host setup. It is designed to integrate with equipment in the lab, allowing control of physical and simulated entities in real time.

Intel® processors are key to this flexibility. According to Zimmermann, large simulations require Intel® Xeon® processors with at least 32 cores and 256 GB of RAM. Many customers run the MIMIC Simulator on multiple hosts—either virtualized or bare metal—to create even larger and more dynamic simulations.

In any simulation, the crucial aspect is the interface to the system under test, according to Zimmermann. To ensure support for IoT systems, Gambit Communications has added a bevy of common network protocols like MQTT, CoAP, and HTTP/S. From a programming perspective, MIMIC Simulator can be controlled in Java, C++, Go, Python, Perl, TCL, JavaScript, PHP, or OpenAPI. The tool supports orchestration via Docker to fit into workflow automation pipelines.

The deployment model is equally flexible. Zimmermann reports that roughly 60% of users run simulations on-premises, with 10% running in air-gapped labs. The rest run at least part of the simulation on the cloud to minimize cost.

Mission-Critical Simulation for DevOps Workflows

This openness and flexibility combined with the MIMIC Simulator's speed and determinism can make DevOps QA and regression testing a reality for mission-critical engineering teams. And, in addition to enabling iterative updates and reducing development time, technical organizations are leveraging the platform for everything from technical training and sales demonstrations to pre-hardware modeling of IoT devices.

Now, with the ability to conduct large-scale advanced simulations, mission-critical systems and the engineers who build them can finally take advantage of modern workflows and drive the full benefit of iterative IoT update paradigms in the same way consumer devices have for years.

Let the mission-digital transformation begin.