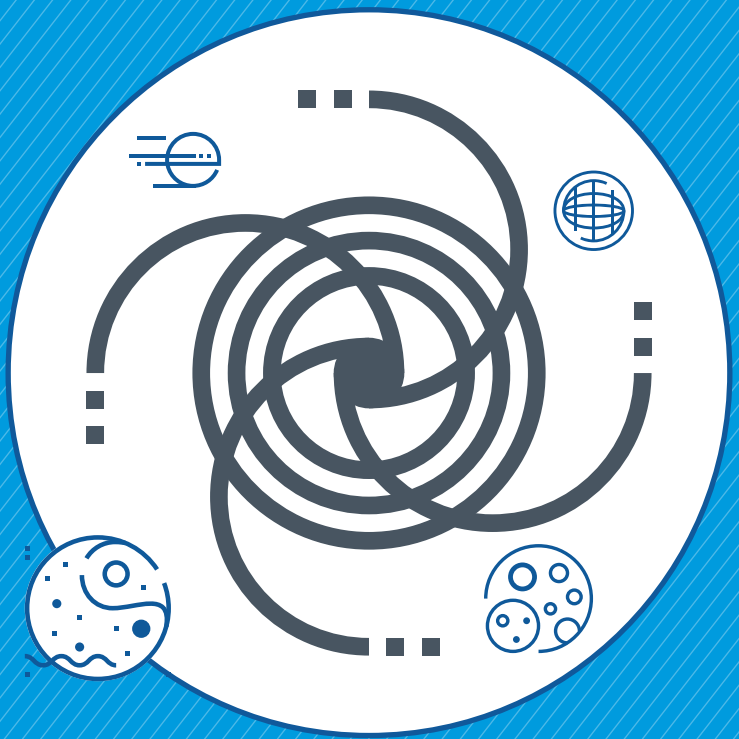
XebiaLabs

11 Black Holes of DevOps

How Not to Get Lost in Outer Space



xebialabs.com

11 Black Holes of DevOps
How Not to Get Lost in Outer Space

Executive Summary

As your organization kicks off its DevOps transformation, you're aiming for wild success... and headed for all sorts of adventures. This isn't a test flight to Pittsburgh and back—this is a journey into uncharted territory! As you scale to create an enterprise DevOps process that works effectively across hundreds of applications and thousands of people, you'll inevitably discover that things don't work the same way they did back on the developer's laptop with a single team, or a single environment.

This paper describes 11 "black holes" you can easily get sucked into. But before we blast off, let's orient our navigation equipment so we're all in the same universe.

A Primer: The Spacescape of DevOps, Pipelines, Toolchains, and Application Release Automation

DevOps is a software engineering practice that aims to bring Development and Operations teams closer together to achieve shorter development cycles and more frequent, more reliable software deployments. As companies venture off into galaxies unknown, the DevOps journey usually brings big changes to a company's culture, processes, and tools. Ultimately, DevOps is about delivering products, features, and services to customers and to the market faster—and automation of the complete application lifecycle plays a key role.

Reliably promoting software through a delivery pipeline, step by step and from environment to environment, typically requires a toolchain of myriad software tools designed to carry out a variety of specialized delivery functions, from Continuous Integration to configuration management, from build to Application Release Automation... and often many more.

Application Release Automation (ARA) orchestrates the overall build, test, provisioning, configuration management, and deployment tools in your software delivery pipeline to automate the release process, starting with the initial idea of a feature and ending with the ultimate delivery of that feature to the user. ARA also encompasses automating the deployment of applications across multiple environments in the pipeline, from Development to Testing to Staging to Production.

An ARA solution that spans your entire real-world software delivery pipeline can provide real-time status information to everyone who is involved in the release process: not just Developers, Operations staff, and release managers, but compliance officers, line of business owners, and many others. With ARA, you can improve collaboration across teams, and ensure that everyone has the visibility they need to keep the pipeline flowing. ARA solutions also collect important metrics about completed releases, so you can look back and identify bottlenecks and areas for improvement. ARA is specifically designed to keep software releases moving seamlessly through your pipeline and to alert stakeholders when things go wrong, so your DevOps vector stays straight and true.

ARA functionality is also sometimes called Application Release Orchestration (ARO) or Continuous Delivery and Release Automation (CDRA).

11 Black Holes in the DevOps Galaxy

So you're setting off on your DevOps journey. Maybe you've had a successful DevOps pilot with a team or two. And now it's time to reach for the stars! Now you're going to implement DevOps across your enterprise... and if you're like many who have gone before you, that means you have to plan for hundreds or even thousands of people, hundreds or thousands of applications, and thousands of deployments per month.

And if you're like those previous pioneers, you've written a lot of scripts to make your pilot projects work—scripts that create infrastructure, scripts that manage deployments, maybe even scripts that try to orchestrate release pipelines or string together your DevOps toolchain. But as you stretch to implement DevOps at enterprise scale, scripting DevOps can start to consume your teams completely, sucking them into black holes of perpetual maintenance and unmanageability, never to be seen or heard from again. Or at least never to reach the efficiency and productivity your business demands.

As you embark on your enterprise DevOps journey, you'll find you have many new complexities to consider. **Here are 11 core DevOps requirements that turn into black holes when you try to use scripts to address these complexities.**

Deployment plans that can span multiple stages and environments

Deployment can be simple if you're deploying a single application to a single environment that never changes. You can script that with ease. But today's enterprise applications involve deploying multiple times at different stages in the process, in different environments. Deployment has traditionally meant writing and testing scripts, customizing and configuring them for different applications and environments, and updating them every time something changes... all manually.

A more modern and scalable approach is to automatically generate standardized deployment plans based on a model of your application and environment. Relying on a model means that your deployment plans always execute the right steps at the right time, so extensive custom scripts are not needed. And when deployment plans are generated automatically, development teams can spend less time writing and maintaining scripts and more time innovating. Black hole avoided!

The hidden cost of scripting your DevOps pipeline

Trying to orchestrate your DevOps pipeline and related toolchain with scripts costs valuable time writing, testing, and maintaining customized scripts that have to be adapted for different applications and environments. Manual scripting costs you time and money in the short run and in the long run.

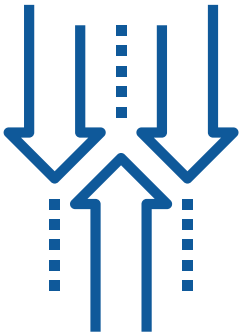
Example: 1000-person IT organization

Avg. time each developer spends scripting and maintaining deployment and release management related tasks	8%
Number of developers per team	8
Number of development teams	100
Average fully loaded cost per FTE/year	$125,000
Annual cost of scripting	**$8,000,000**

Release orchestration and dependency management to handle complex application architectures

As you scale to release hundreds or even thousands of times a month, repeatable release coordination and comprehensive dependency management become increasingly important. If a component release fails, how does that affect the entire release? If a task fails, how should the pipeline respond?

At scale, it's challenging to write scripts to coordinate the many tools in your DevOps toolchain and move your releases quickly through the pipeline. And it's nearly impossible to script management for dependencies that result from complex relationships between application components.

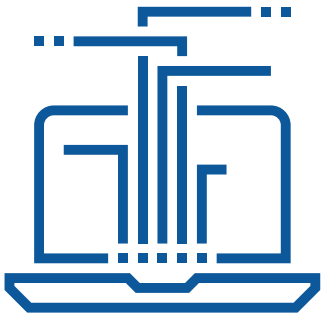


Effectively organized release orchestration involves stakeholders across the business who are responsible for both manual and automated work. An ARA solution can orchestrate all of that work, as well as manage the technical dependencies in your infrastructure and the logical dependencies among applications and microservices.

3 Software delivery pipelines that support advanced release and deployment patterns

The larger and more complex your environments are, the more benefit you'll see from advanced deployment approaches such as blue-green deployments, canary deployments, and rolling releases. Implementing these patterns with a provisioning, configuration management, or Continuous Integration tool requires extensive manual work and scripting, which means more complicated maintenance... forever.

Take advantage of the pre-built DevOps best practices in an ARA tool, rather than trying to script. When you have built-in support for sophisticated release and deployment approaches, adapting a deployment for a particular situation is as easy as selecting a different pattern. As teams mature in their Continuous Delivery adoption, they can experiment with the patterns that are best for their goals, while still benefitting from standardized processes and pipelines.



4 Intelligent, automated rollback from failures

When a deployment fails, you need a tool that can take action, fast. If you rely on custom, handwritten scripts, you won't have automated rollback when things go wrong—leaving you scrambling to resolve the failure and restore your application to a usable state. Of course, rollbacks could be scripted, but that just means the number of scripts you're dealing with multiplies exponentially, and the overhead to make sure those rollbacks scripts are all properly implemented increases dramatically. Depending on where the failure occurs... your rollback script might fail, too.

Intelligent rollback means you don't have to panic when a deployment goes wrong. The right ARA solution provides more than simple "undo" actions; instead, it takes a full approach to return your application and infrastructure to their previous state, no matter where a deployment failed. And most importantly, it's built in, so nobody has to remember to take the time and manually create custom roll back plans. Automated rollback ensures business continuity.

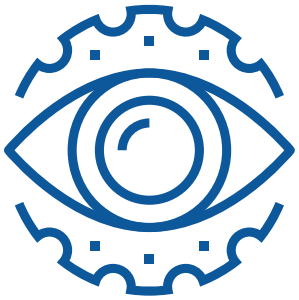
5 Standardized processes that scale DevOps across your organization

Relying on customized application deployment scripts that you have to write and maintain inevitably leads to disparate processes and one-off scripts that cause real trouble as you try to scale DevOps across the organization. Each one is unique... and that much harder for anyone but the original author

to maintain. Ill-advised teams can easily get sucked into a black hole trying to manage non-standard processes created by scripts. Precious DevOps resources may be forever tied up deciphering and maintaining these scripts, never to be seen or heard from again.

ARA provides the foundation to standardize processes for releasing and deploying applications. Scaling your DevOps initiative by modeling releases and deployments with a declarative ARA platform is a practice that enables teams to reuse delivery pipelines across tools, environments, and departments... all while ensuring that every team automatically collects data needed for compliance and audit purposes.

Visibility into your real-world release processes, covering both manual and automated tasks



Enterprises have many DevOps tasks that can't be automated and require manual intervention, and scripts are inherently not suited for many of these activities—especially interaction with non-technical users. The scripts you might add around Continuous Integration, provisioning, configuration, or deployment tools don't account for any manual work in the release pipeline, which means they don't represent your real-world process.

Realistically, you need oversight and management of your end-to-end, code-to-production delivery pipelines, including both manual and automated tasks. This visibility benefits business users and highly technical users alike by giving all teams the up-to-date information that they need to collaborate. Using a shared, purpose-built ARA platform means that everyone can participate and work together towards Continuous Delivery best practices, no matter what their role.

Collaboration with stakeholders who aren't script-savvy

Adopting DevOps means embracing collaboration across the enterprise so that everyone involved—regardless of their level of technical skill—can support faster software delivery cycles. For most enterprises, stakeholders who can't write release or deployment scripts still play a crucial role in the software delivery process.

ARA widens release scope to encompass the complete software delivery pipeline, providing a clear, visual user interface and configurable dashboards to manage releases and help share information. It's designed to bring both technical and non-technical users into the process, where they can take ownership of release activities.

The right ARA solution makes it easy for all types of users to create, monitor, and troubleshoot the steps in the software delivery process, whether those steps exist to meet technical or business requirements. For example, business users and security teams can easily participate in the release process at all stages, staying closely involved throughout the release.

8 Reports that reflect the end-to-end release pipeline

Of course, a fully scripted release or deployment pipeline can collect data that allows you to analyze your processes, but that data is inherently limited to what you explicitly script, and it requires manual review and analysis to find bottlenecks and areas of improvement. To fully optimize DevOps, your teams need reports that reflect the end-to-end release pipeline.

An ARA solution that brings together all of the steps and tools in your software delivery cycle can provide configurable reports that visualize the entire release process, showing bottlenecks and highlighting areas that would most benefit from improvements. Both business and technical users can take advantage of report data.

Also, the right ARA solution provides targeted goal-based insights—for example, allowing teams to set goals for release efficiency or speed. It can provide essential guidance based on DevOps best practices, machine learning, and historical performance to keep teams moving quickly along their path of continuous improvement.

9 Proactive assessment and mitigation of release risks

The status of a release—and the chance that it's going to fail—isn't just a technical concern. All stakeholders need to see release status at a glance, and they need proactive alerts when releases are at risk of failing. Any status information that can be gleaned from Continuous Integration and deployment scripts is only useful for technical users, and scripts don't notify you until something has already gone wrong.

Visibility into the true status of releases has to encompass the whole release pipeline, taking manual and automated work into account. Enterprises need built-in risk intelligence and proactive alerts about impending task failures and release failures, in order to ensure that all stakeholders are kept fully informed in the software release process and can quickly address issues that block delivery.

Automatic collection of compliance data as a built-in part of the release process

Trying to hand-code and maintain scripts that identify and collect the compliance data you need is highly error-prone, especially as you scale. Gathering compliance data via scripts means, at best, digging through logs to get the data you need is painful, and at worst, you may accidentally miss critical information needed for audits. And it's yet another black hole that will whisk your resources away into the void.

With the right ARA solution, comprehensive compliance data is collected automatically as part of your release and deployment process. When compliance data is easy to access and easy to read, satisfying compliance requirements and providing information for audits is a simple task instead of time-consuming, frustrating work. As a bonus, this data is delivered in an easy-to-use format. As an even bigger bonus, you can enforce the execution of all compliance steps as part of every release.

Governance to enforce separation of duties and enterprise access control

As development teams are empowered to deploy applications as part of their Continuous Delivery process, a simple approach to user management and granular access control becomes more important to ensure that code that doesn't belong in Production doesn't end up there. If access control is defined in code, then the burden of configuring and maintaining it falls to technical staff. Public companies and organizations in regulated industries are often specifically required to ensure that access control is carefully managed and that separation of duties is enforced.

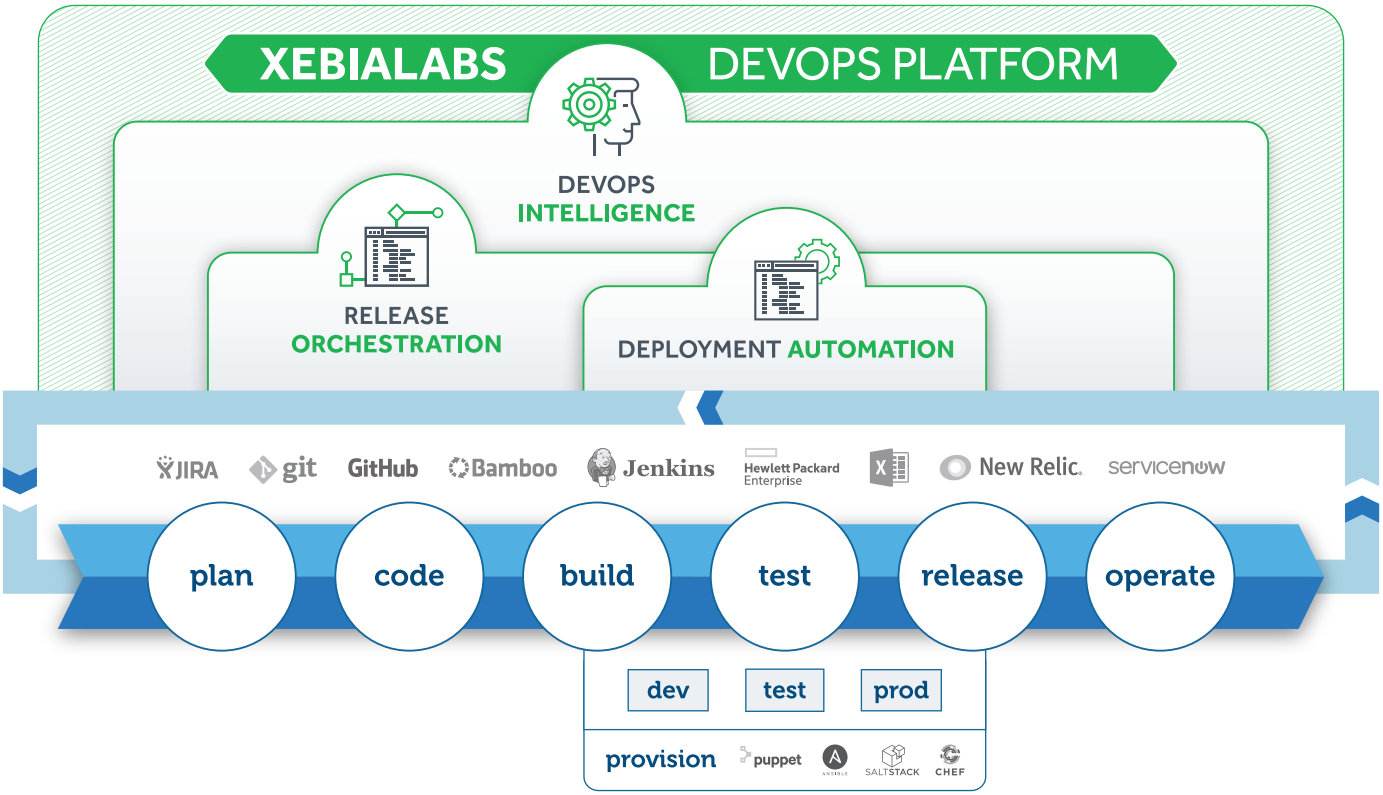
In contrast, an ARA solution with straightforward, easy-to-use user management and granular access control enables the right people throughout the business to take ownership, regardless of their level of technical skill.

Avoid the Black Holes with XebiaLabs

With its model-based, highly scalable approach to deployment, the XebiaLabs DevOps Platform is an ARA solution that empowers teams to spend less time writing scripts and more time innovating. It allows you to orchestrate complex release pipelines from end to end, across all stages and all environments. You can immediately deploy applications to new target systems as your infrastructure grows, without installing additional software or customizing configuration—

all while taking advantage of advanced patterns such as blue-green, canary, and rolling deployments, and reaping the benefits of intelligent, automated rollback when things go wrong.

With XebiaLabs, you can standardize processes by modeling them in release and deployment templates that can be used again and again, creating consistency across your organization. The friendly UI works for business users and highly technical users alike, giving all teams the up-to-date information that they need to work together and making it easy to ensure that compliance data is collected automatically as part of the software delivery process. A shared platform means that everyone can work together to implement DevOps best practices, no matter what their role. With these features and many more, the XebiaLabs DevOps Platform gives you the foundation you need to do enterprise DevOps on a large scale. Use it to avoid the black holes of DevOps and keep your fleet on course.



Continuous Integration tools, Configuration Management tools, and Application Release Automation solutions... what's the difference?

So now that you know some use cases that demand an enterprise ARA approach, let's talk about the differences between ARA and related tools in the software delivery pipeline.

- Continuous Integration (CI) tools such as Jenkins, Travis CI, TeamCity, and Bamboo automatically integrate code when developers check in their changes. CI tools use scripts to coordinate activities, such as building software artifacts and running automated tests against them.
- Puppet, Chef, SaltStack, and Ansible are Configuration Management (CM) tools that automate your infrastructure by creating and managing your environments. Like CI tools, they use technical specifications that you define in scripts.
- Application Release Automation (ARA) solutions, such as the XebiaLabs DevOps Platform, are designed for a very different need: they automate the process of releasing your software and deploying it to different environments in the enterprise release pipeline, from Development to Testing to Staging to Production.

While a CI tool integrates and tests code, and a CM tool lets you describe an environment and then sets that environment up for you, ARA orchestrates and manages the whole application release process from end to end. Use ARA to coordinate environment provisioning and application deployment with all of the other tasks needed to get your software up and running.

Get the job done with CM, CI, and ARA tools together

- ✓ Eliminate costly script maintenance and future-proof your deployments
- ✓ Integrate manual and automated work in the release process
- ✓ Take advantage of sophisticated orchestration, from release to deployment
- ✓ Get cross-team collaboration and visibility for technical and non-technical users
- ✓ Standardize and enforce DevOps processes
- ✓ Build compliance into the release pipeline

Conclusion

DevOps processes that are simple for one project and one team grow drastically in complexity as you scale them across the organization, seek to meet compliance requirements, and make plans for continuous improvement. Scripting release and deployment instructions may work at small scale, but scripts become prohibitively expensive to create and nearly impossible to maintain as your DevOps initiative grows. If any of the black holes outlined in this paper sound like core parts of your DevOps "spacescape," don't try to script them at scale, or you'll watch helplessly as your precious resources disappear into the void!

The XebiaLabs DevOps Platform is an Application Release Automation solution that models releases and deployments so you can easily standardize processes and spread DevOps innovations across the organization. It eliminates the need for unmaintainable release and deployment scripts, providing the release infrastructure you need to meet compliance requirements and continuously improve your software delivery process.

About XebiaLabs

XebiaLabs develops enterprise-scale Continuous Delivery and DevOps software, providing companies with the visibility, automation and control to deliver software faster and with less risk. Global market leaders rely on XebiaLabs to meet the increasing demand for accelerated and more reliable software releases.

The XebiaLabs DevOps Platform for Continuous Delivery at Enterprise Scale

Release Orchestration

Orchestrate, automate, and get visibility into release pipelines

Deployment Automation

Automate and standardize complex application deployments

DevOps Intelligence

Get unprecedented insight and decision support for your software delivery process

For more information and a free trial, please visit **www.xebialabs.com**.