



4 Tips for Scaling Enterprise Software Deployments

Help your organization achieve enterprise scale deployments

What is Enterprise Scale Deployment?

First, it's important to realize that scale is not just about technology performance. Scale is a combination of technology + people + process.

In addition to the number of deployable applications, artifacts, resources, as well as the number of targets where they need to be deployed, you need to consider the number of engineers working across the organization.

So in this way, enterprise scale deployment is the ability to orchestrate elite software deployment across multiple engineering teams and departments, across multiple deployment targets.

Learn how to solve some of your toughest enterprise scale deployment challenges in this eBook.

1

Assess Your Maturity

How do you determine the effectiveness of your current deployment processes?

Scale is a spectrum of running development, integration, and deployment manually or automatically. Assess where your team is in the spectrum of manual to automated.

- **Manual tasks are defined as needing human interaction or input.**
- **Automated tasks are performed via tools, scripts, or software.**

Establish the reliability of your code and deployments including error rates, incidents, and roll-backs. Document these metrics and establish a baseline for your upcoming progress and to share across the organization. Create action items to begin chipping away at tasks that can start to be automated in your team.



Need help improving your deployment maturity?

Armory's Continuous Deployment-as-a-Service automates many common manual tasks found in software deployment.

Automated deployment leads to:

- Improved code quality
- Improved efficiency
- Faster velocity
- Higher accuracy
- Fewer errors

2

Declarative vs. Imperative Deployment Strategies

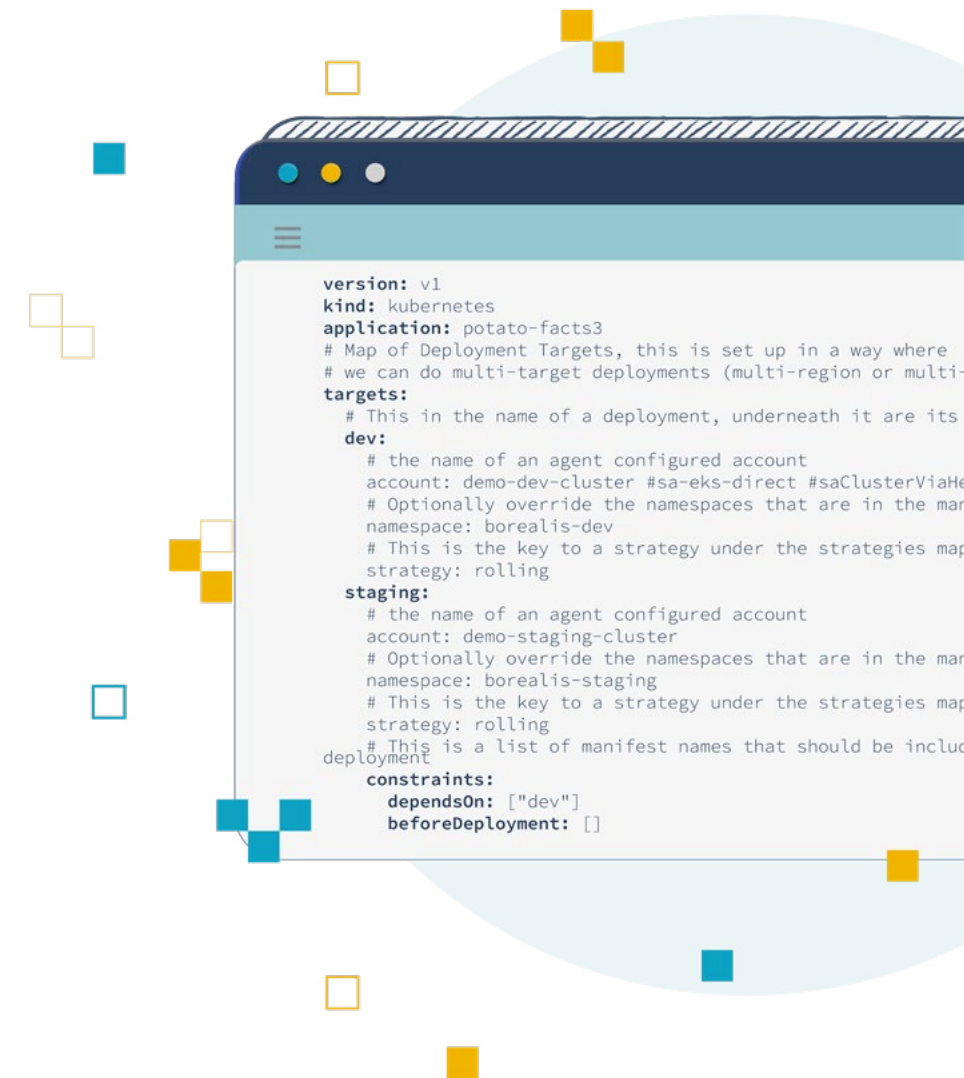
Which approach is the
right one to help your
enterprise scale?

Now that you have a baseline, it's time to put the right deployment strategies into play. There isn't one-size-fits-all for an enterprise, but there are a couple foundational options to explore.

Declarative focuses on “**what**” and imperative focuses on “**how**”. The more automated you get, the more declarative your software deployment process should be.

When you use declarative deployments, instead of writing a set of instructions for one deployment use case and describing exactly how you want something done, you simply state what you want done and Armory figures out how to do it.

This means you can cut back on the manual nature of your deployments and speed up how quickly you can bring new innovations, like a “suggested for you” feature, to your customers.



Need standardized and version-controlled deployments?

Armory makes it easy to create and enforce standardized advanced deployments using GitOps techniques with version controlled declarative deployments, while also enforcing policies like who can approve **what** and **who** can deploy **where**.

Standardized deployments lead to:

- Improved customer experience
- Accelerating your time-to-market
- Improved developer experience
- Decreased deployment risks
- Simplified operations

3

Spin Up New Teams *Fast*

Take advantage of advanced team deployment capabilities to enable organizational growth.

As we've discussed, scale is more than the number of Kubernetes clusters that you have. You're probably already using a powerful imperative tool to deploy to multiple clusters across teams and across the business.

But did you know that you can also use advanced GitOps workflows to quickly spin up new teams including access, and more, all in a simple declarative configuration? For growing enterprises, the ability to set up and empower new development teams and new capabilities can mean a faster time to market and more revenue.



Don't know where to start?

The best way to accomplish this is with Armory's solutions to drive more efficiency, productivity, and automation.

Advanced workflows enable you to:

- Add new teams and new opportunities faster
- Establish consistent processes across the organization
- Take full advantage of powerful continuous deployment capabilities

4

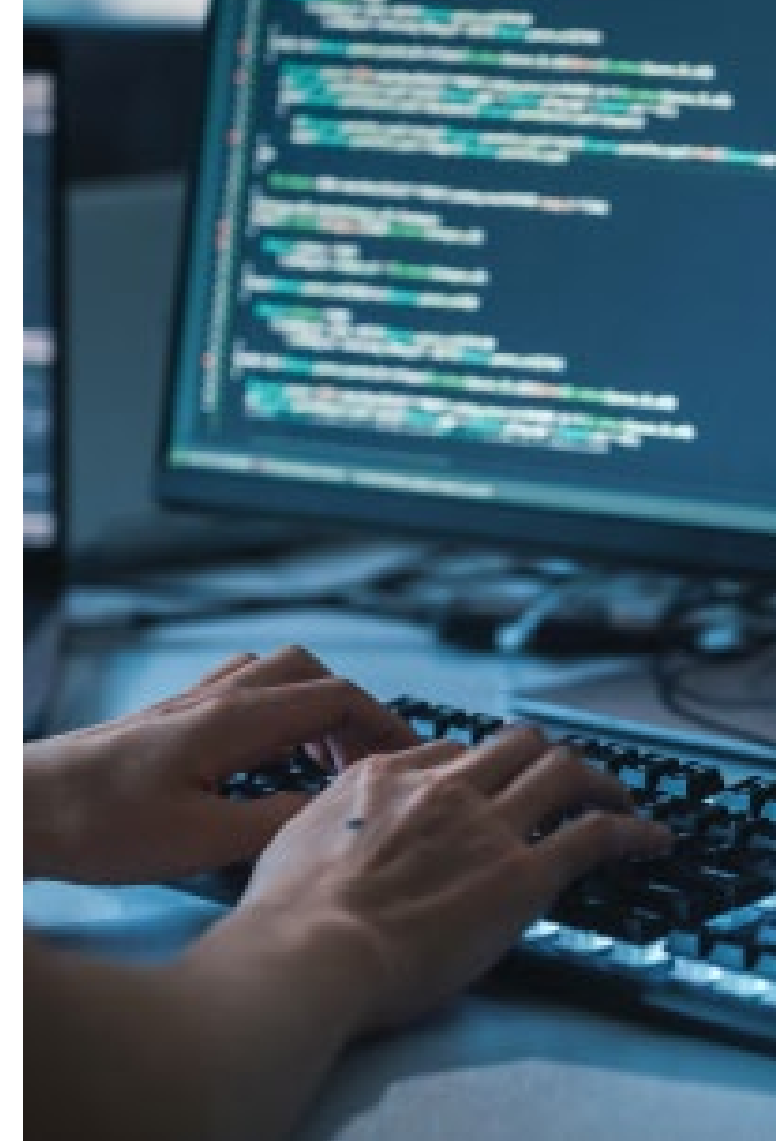
Monitor and Report

Use continuous deployment data to measure and share success.

How are you going to monitor and report on the progress or success of the developer experience and deployment improvements you make at your organization?

First, focus on the signals that matter. Because you took the time to assess and establish a baseline for metrics like deployment frequency (DF), lead time for changes (LT), mean time to recovery (MTTR), and change failure rate (CFR), now you can begin to report on improvements to these metrics over time.

Another important metric to measure is developer satisfaction. When you move away from manual deployment processes and into more declarative and automated workflows, your developers become more productive and are freed up to explore more creative solutions to the problems that your company solves.



Take Advantage of Deployment Data

Continuous deployment solutions capture key data that you need to show success over time.

Continuous deployment data enables you to:

- Unite data across the development and deployment lifecycle
- Find new opportunities for improvement in technology, people, and processes

Scale is more than technology. Continuous deployment helps you unite technology, people, and processes to achieve software development at enterprise scale.

Armory's automated processes and declarative deployment experience take the burden off of your DevOps team and drive increased efficiency, innovation, productivity, and velocity.

Commit. Deploy. Repeat.

Continuously deploying software at any scale should be achievable and easy for all developers. Armory's developer-first declarative and flexible continuous deployment solutions make that a reality.

Get Started for Free



info@armory.io

1-888-222-3370