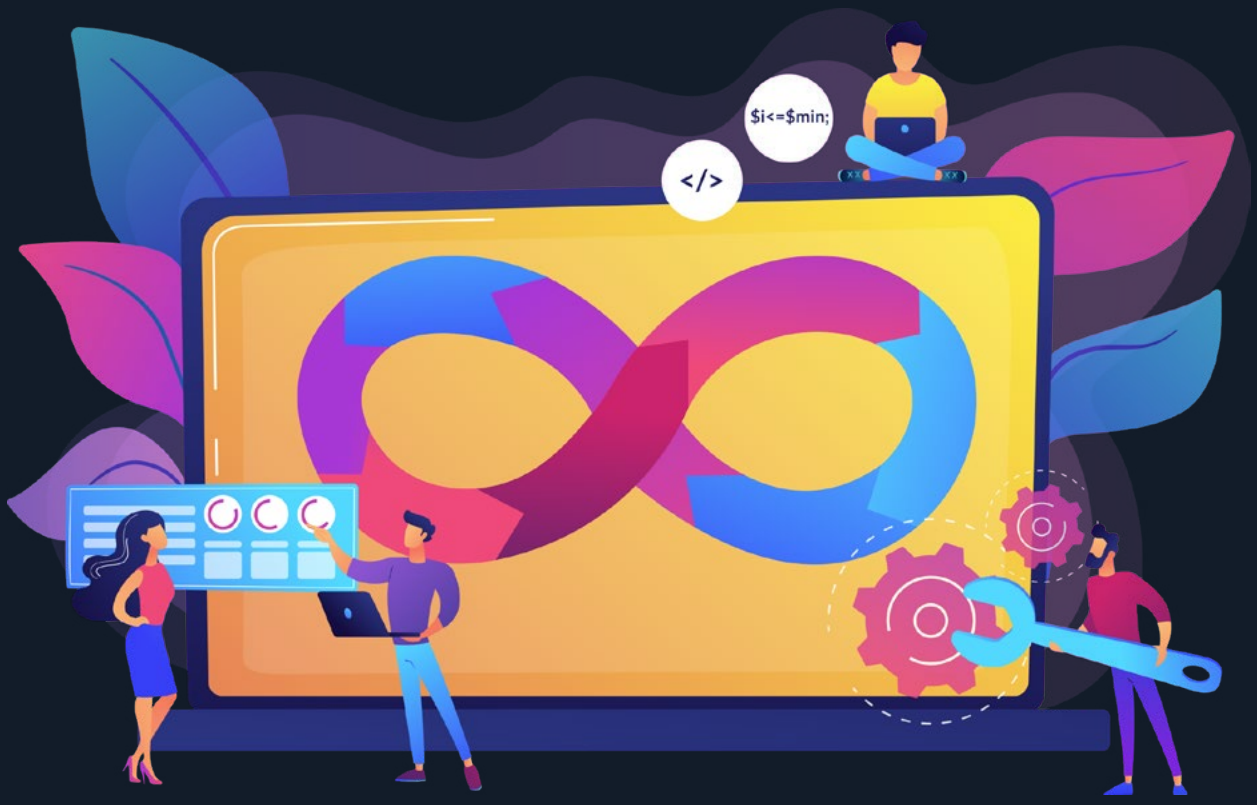
GO2GROUP

End-to-End DevOps Transformation: How to Get There



The DevOps revolution is a key cornerstone of the digital transformation that is spreading through the global economy. DevOps is an enterprise-wide revamping of the software development, deployment and iteration process on an end-to-end basis with the goal of delivering better software faster. This white paper aims to answer what 'end-to-end' really means in a modern enterprise context. It is important since DevOps processes – properly planned, implemented and managed – will become a prime means of creating value for customers and stakeholders in any enterprise. The answer also shapes follow-on questions such as: How do we do this?

Since every enterprise is different, there is no one-size-fits-all solution. We propose a set of approaches on the end-to-end question, which can serve as an initial base for making important decisions regarding your DevOps strategy across the enterprise:

- Assessment and strategy planning
- The environment and pipeline
- Scaling into the future

Each of these approaches offers a different take on what is meant by "end-to-end." The phrase implies a beginning point and a concluding point. By starting in different places, enterprises can modernize their thinking as well as their technologies.

Assessment and strategy planning

Assessing where you really stand in terms of organizational maturity, capability and technology is the place to start. A roadmap can be developed from there. Key questions to answer include:

Do you have the right people?

In many respects, this is a cultural transformation that has a broad impact not only on IT, but also the business units and the organization at large. DevOps requires more collaboration and communication among teams and different parts of the business, which makes people skills more important. Developers no longer are confined to working in silos defined by their individual projects. Rather, they are part of a diverse team that must communicate with one another in order to keep things flowing smoothly through the pipeline. The team may consist of members from different parts of the organization, not just IT. Managers looking to put together a DevOps team should look for candidates with a mix of process skills, soft skills, functional skills, and technical skills.

Navigating this culture shift and operating productively in this new way requires new skills among the IT staff. Coding will always be important, but this new environment places a premium on architectural skills. Deconstructing monolithic applications into their most efficient, productive services becomes a new discipline that might well be unique to any given enterprise (i.e. a retail operation might approach it differently from a manufacturing concern). In creating new applications within a microservices architecture (MSA), the building blocks are largely pre-formed rather than coded by hand. The ability to envision the various required components of a larger application and how data flows among those components is highly valuable.

How will processes need to change?

Many, if not most, of your DevOps processes will be automated, so it is crucial to establish a framework for what will work best for the organization and the business needs. Establishing and tracking metrics should be part of this initial discussion. Typical data points to measure include:

- Quality
- Speed of delivery
- Deployment frequency

- Change lead time
- Change success/failure rate
- Mean time to recovery from pipeline failure
- Passing rate of security tests

By tracking these (or your chosen) metrics over time, you can spot emerging **choke points** as your pipeline and application mix evolve.

Key considerations include establishing rules around what triggers an automated build or deployment. Those triggers might well shift after some experience is gained in practice. Critically, the goal is to build any application iteration just once. Version control and tracking require rules around what triggers builds, in addition to optimizing overall pipeline performance.

How will security be integrated into your DevOps implementation?

Part of starting at the high level should include security planning. Security teams should be part of this upfront planning strategy. Containerization becomes a building block for security as internal applications processes can be isolated from one another. Automating security testing should be prioritized, along with compliance testing.

It may be prudent to insert security gates into the pipeline so that human intervention can confirm all protocols have been followed before release. Putting these security controls into place requires thoughtful planning so that you are optimizing security without sacrificing speed. Understanding **best practices** from other implementations can help strike this balance.

The environment and pipeline

DevOps processes are most effective when built atop an environment defined by MSA and containerization. The key technologies necessary for building this framework consist of:

- Containerization
- Orchestration
- Monitoring

Containerization tools package a microservice with the additional data and components, such as an OS kernel, they need to execute. The important point about using such a tool is that it standardizes this process of constructing containers across your environment. By far, the most popular containerization tool is Docker, an open-source project that automates the process of developing, deploying, and running applications inside containers. It provides a consistent development environment for the entire team and enjoys broad support among developers.

Orchestration is a management layer that interfaces the individual containers to each other and to the ultimate application being constructed on the fly. The orchestration layer is responsible for maintaining service level agreements (SLAs) and scheduling services into optimal execution resources on the host. Orchestration implements the "loose coupling" function of the MSA model, maintaining all the necessary linkages while still allowing the individual building blocks to operate independently. Kubernetes, an open source contribution from Google, is the most widely used orchestration technology though Docker Swarm is also a well-known entry.

Monitoring tools provide visibility into how microservices are being used. Traditional monitoring tools from a monolithic application model are not built to comprehend the loose coupling, communication and multi-component nature of MSA and containers. Monitoring tools allow you to pinpoint where any errors occur and which microservice is causing the issue. Prometheus is a well-regarded option for monitoring.

If the environment provides a new starting point, then, the end point in this perspective is the construction of your continuous integration/continuous delivery (CI/CD) pipeline. One size does not fit all when it comes to matching CI/CD technologies to your needs. As you get set for technology evaluation, certain questions should be addressed to establish a framework for measuring the adequacy of any possible tools.

How much customization are you willing or able to do on your own?: This is a key tradeoff to consider. The answer has implications for costs, time of implementation and the resources you need in-house or through other avenues such as consultants (which will also add to costs). This could steer you toward one tool or another in your evaluation process.

Related to customization, what is the level of integration you want or need for your environment?: How easily does a given tool integrate with the entire pipeline environment? These questions lead to further issues of how much work you will

need to put into actually using the tool. Lastly, how well does this tool integrate with your larger architectural needs (i.e. where you fall on the monolithic to MSA spectrum), with your workflows and even the skillsets of your people?

Where will your CI/CD pipeline be hosted?: Your provider might have technologies already on offer or operate an environment more conducive to certain flows over others. It's also important to note where the data your applications rely on is hosted, since multi-cloud strategies are growing in popularity. New links might need to be built or API calls designed into the flow at the very least. Can the tool you are evaluating handle all those linkages and still give you the performance you need?

What level of automation are you looking for?: The key to CI/CD is automating as much as possible, of course, but establishing your desire for one-button automation versus key checks and balances in the pipeline flow is an important issue to address.

Many of these issues can be addressed by adopting more integrated suites along with managed services and cloud hosting. Much of the integration is already done across the suite, but because they are built on open-source foundations they link more easily to other technologies that are core parts of the DevOps tool ecosystem.

For instance, **Atlassian** offers technologies across the CI/CD pipeline. Atlassian Bitbucket integrates with Git to pull and publish code. Atlassian tools can be installed on an on-premise server or hosted in the cloud where Bitbucket is known as Bitbucket Pipelines. Atlassian Bamboo can build, test and deliver new software. It integrates with Bitbucket. Bamboo also offers an online marketplace with third-party plug-ins for added functionality or specific testing suites.

For orchestration, **Jenkins** is the de facto standard. It interfaces with most toolsets on both the development and operations sides of DevOps. CloudBees is an active member of the Jenkins open source community and offers a broad suite of tools focused on the delivery and deployment side of CI/CD. It brings automation capabilities and analytics to see where the pipeline might have bottlenecks. **CloudBees Core** helps manage multiple teams and applications across toolchains.

All of these integrated suites have options for **managed cloud hosting** and **managed services** approaches. Turning day-to-day management of these technologies over to a provider does not lessen your control. It simply allows your own teams to stay focused on their mission rather than managing licenses and cloud provisioning. When selecting a provider, look for a partner with deep experience in the technologies you're using and the cloud provider you select for hosting. **AWS**, **Microsoft Azure**, **Google Cloud** and others all do things such as

provisioning and shared security protocols differently. Choose a managed services partner who can manage the whole solution.

With today as a starting point, looking to the future

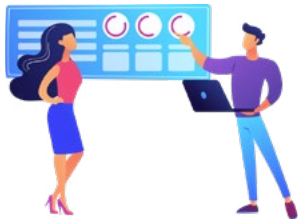
The last approach to consider is perhaps the most important. *End-to-end* can mean starting with today and scaling into the future. This requires asking a different set of questions: How will your business evolve? What options might you need to consider to accommodate growth or M&A activity or new applications? You are no longer simply managing hardware and software, but a roadmap into the future.

It is important to impress upon management that a new mindset is required in strategic planning with IT operations underpinning the customer experience, organizational efficiency and the very value creating mechanisms of the enterprise. In a fast-moving global economy, you will constantly be evaluating your DevOps needs. On an ongoing basis, you will need to learn best practices gleaned from other environments that might be similar to your own as well as stay informed of developments in the open-source community that involve DevOps technologies. Trusted advisors who can bring outside knowledge and perspective as well as hands-on support become key parts of the IT team in this new world.

Conclusion: From revolution to journey

DevOps enters enterprises as a new way of building software. It matures into a way to create value for customers and stakeholders. Technologies help us define how to do all these things on an end-to-end basis.

However, if you do all this right – from strategic planning to technology deployment and management roadmap – DevOps becomes a journey that is part of being a competitive player in a fast-paced world. **Trusted advisors** are needed to navigate a journey that actually will have no end point. Part of planning for that journey is understanding with whom you want to travel.



GO2GROUP

Go2Group, an IT consulting firm focused on providing superior quality professional services and digital solutions, helps enterprises achieve business agility. Founded in 2002, the company offers enterprise-level strategic services and solutions around application modernization, Agile and DevOps practices, and cloud.

Organizations in 70 countries benefit from its expert IT services that drive teams toward high-performance, empower teams to seamlessly accelerate application delivery, and enable increased efficiency through cross-team collaboration.

Building on our platinum-level partnerships with companies such as Atlassian, CloudBees and AWS, and with our strong and dedicated team of trained and certified technical consultants, enterprises can leverage the right mix of consulting-led strategies and technology-led solutions to direct their IT initiatives toward business growth.