Simple IT Automation.
Smart DevOps.

Your guide to accelerating DevOps with Ansible automation

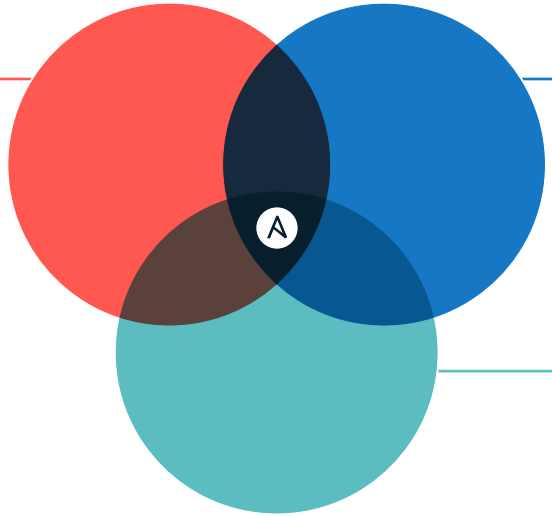
No matter where you are on your DevOps journey, **Ansible automation can help.**

Implementing DevOps tools and processes can help revolutionize your organization. But adopting a DevOps framework doesn't require updating your entire IT stack to newer agile implementations first. Quite simply, your organization can embrace DevOps through automation, even if you are running only on bare metal, migrating to the cloud, or already going full force into containers. And your team can start today.

In this guide we introduce three key phases of Ansible automation for DevOps and how you use them to deliver with greater efficiency and speed.

MODERNIZE

Automate existing processes one step at a time



MIGRATE

Define applications once so you can deploy them anywhere

OPTIMIZE

Encourage broad participation and knowledge sharing

Modernize

Becoming more agile in today's digitally-driven market is business critical. Your competitors are launching new offerings at an exponentially increasing pace, and new competitors are showing up every day. From releasing an app upgrade to deploying a full-fledge offering, you must innovate faster than your competition. These business demands can put unprecedented pressure on IT departments.

Many IT departments are running a multitude of legacy systems and stacks. However, the age and state of your hardware does not mean you cannot compete. Through automation you can modernize your existing infrastructure.

This section explores the easiest way to get started with automation and actionable steps you can take on the path to modernization.

Start with the OS

When you begin to look at how to inject more agility and speed into your existing infrastructure, start with the operating system. As you automate the existing tooling and other components of your OS build process, automation allows you to fluidly and dynamically make sure the same copies of your operating environment are then deployable everywhere.

You get a succinct flow when you're using a modern automation platform like Ansible to spearhead or frontend OS deployment. Automation replaces many of the individual or manual steps for delivering configured and approved servers to your app delivery team.

You may start with the OS, but a lot of the same processes and approaches are easily applicable to other aspects of your infrastructure as well. Whether application development team, additional security requirements, or networking—all are automatable.

As you implement a similar process across the board, you'll have a much more fluid environment. And you are going to be significantly more flexible wherever you need to deploy your applications in the future.

Focus on the process

Ansible automation provides continuous compliance, feedback loops, portability, cooperation, and coordination.

THE PROCESS ITSELF IS THE SECURITY

If you have a manual security baseline, it's not a strong security baseline. Manual processes make environments insecure. When you automate, security will be ingrained in every step.

UNIFY YOUR CURRENT BUILD PROCESS

You probably have a lot of disparate tools. Glue them together with automation to make a succinct flow. You might find steps you don't need because you can accomplish them in one.

ENABLE AUTO-APPLICATION OF UPDATES THROUGH CI/CD

Have patch day every day. CI/CD allows you to test ahead of the curve. Build an environment that is no longer patchwork.

You can make an impact

with an automation-first approach to modernization.

KEY TAKEAWAYS

1. Start with the operating system

Start with the OS because you understand it and you probably have a significant amount of control over it.

2. Focus on the process

Because you have a clear process, your organization will be ready for adoption. Automation will allow you to migrate and deploy your applications and environments pretty much anywhere you want.

ON DEMAND WEBINAR

From Legacy to Devops: Modernizing Application Infrastructure

GET STARTED

We can help you get started on your path to automation.

Talk to an Expert

Migrate

Migrating applications to a new infrastructure is typically full of challenges.

Whether you're migrating an application from physical to virtual, virtual to cloud, cloud to cloud, or perhaps even traditional to containerized—automation is critical in more rapidly understanding all of the components needed to effectively redeploy your application.

This section will explore modern migration techniques.

Manual approaches to migration

Whether it's formal or not, migrating without automation usually involves a checklist of tasks.

☑ DEPLOY SERVERS INTO AN ENVIRONMENT FROM SCRATCH

- Start with physical, virtual, or cloud
- Clone a template or use a kickstart file and install a brand new system from scratch

☑ GET CONFIGURING

- Follow your deployment checklists, walk through your deployment guides, etc...
- Work from a configuration management system

☑ SPOT-MODIFY CONFIGURATION

- Use when you have to layer an application on top of your deployment
- Continue to do so until it eventually works

☑ LATHER, RINSE, REPEAT

- Repeat this process for every environment and deployment

Why doesn't this manual process work?

IT'S NON-REPEATABLE

- Restricts you to a "build it once, run it once" process
- Assures that you can't treat two applications the same
- Once you get it deployed the way you want, any time you need to touch it, you'll need to treat it individually

IT'S 100% BESPOKE

- Bespoke might be good for a suit, but it's not great for your servers in an application environment
- Every process is highly manual
- Long delays every time you might need to make a change

IT'S ERROR-PRONE

- Spot modification and manual steps are inherently error-prone

There is a better way

Define, deploy, discover and repeat with automation.

DEFINE

As you're choosing individual components, you're defining them with Ansible Playbooks and you're modeling how your application is going to look. We then use those definitions to execute a deployment.

You can then accurately define each step, string them all together in a Playbook, and apply that Playbook to a new environment.

- Define as much of the initial deployment as possible
- Automate each component separately, and then pull them together
- Account for the process along the way

DEPLOY

By continually deploying these Playbooks in a repeated manner in your environment, you can slowly build up the sophistication of your automation to the point where you have your application working perfectly.

Ultimately it's up to you to decide the best way to deploy. It's about the process in which you help make applications repeatable across multiple environments.

- Apply your automation
- Test your newly deployed application
- Learn what you missed (e.g., firewall ports, permissions, one service didn't start properly, etc.)

DISCOVER

When you find gaps, that means you've discovered something new that you need to automate.

Remember this is a continuum. Because it's an iterative approach, you don't have to get it 100% right the first time.

- Discover gaps through testing
- Expand the breadth of automation
- Inform the beginning of a new cycle

REPEAT

You can make an impact
with an automation-first approach to migration.

KEY TAKEAWAYS

1. Migration without automation isn't efficient

If it takes you a full day every time you need to test a deployment, that's a painful process. But when automation makes it quickly repeatable, that same process may only take a few hours or even minutes.

Rapid repetition is critical. The end result is that you have a much more fluid mechanism to pick pieces up and move them wherever you want. Remember that repeatable applications are portable applications.

2. Define, deploy, discover and repeat with automation

Automating migration allows you to rapidly iterate and rapidly repeat your process. Every time you automate, you will learn how to refine your process further. These lessons will allow you to increase your use of automation.

ON DEMAND WEBINAR

Transitioning To Cloud: Migrating Application Infrastructure

GET STARTED

We can help you get started on your path to automation.
Talk to an Expert

Optimize

Ultimately the expansion of DevOps initiatives are determined by the ability to share and spread solutions across your organization. Once one person on your team learns how to do something, they can capture their solution with Ansible and enable everyone to use it.

When solutions are shared openly, you're one step closer to modeling everything and deploying continuously.

This section will explore the philosophy of a sharing-driven culture and how optimize DevOps approaches for success.

Heroics vs. Shared Knowledge

In IT we love heroics. Heroes always get the job done. You can always rely on them. It would be great if you could have a team made entirely of heroes, but that's not realistic.

HEROES ARE EXPENSIVE

- If you could afford all heroes, you'd have an entire company of heroes

HEROES ARE LIKE UNICORNS

- They're hard to find and harder to recruit
- And when you do find one, they are really expensive

HEROES BURN OUT

- They can become overwhelmed. And what happens when they leave?

You need a way to share knowledge across your teams. Heroes actually make that more difficult. If just one person or a small number of people are able to do everything, their first priority is doing, not sharing across your environment.

When you start to optimize your process and various aspects of your IT systems through automation, you'll be more successful at communicating and sharing solutions across the organization.

Success Story: **Splunk and Ansible**

Watch how Ansible has helped Splunk share solutions and knowledge through automation:

SOLVE IT.

"When we encounter a new problem, one person figures out how to solve it…"

AUTOMATE IT.

"Then we automate that solution with Ansible…"

SHARE IT.

"And then everyone knows how to solve it."

How to optimize for success

EMPOWER YOUR PEOPLE

- Make sure your people feel empowered to participate in automation. Enable teams to own what they focus on, but encourage broad participation in the process.
- Experiment with feedback mechanisms. Ask staff to make suggestions on how the overall process and automation can be improved in ways that help them and help the customer.

SHARED CONTENT, SHARED GOALS

- Provide individual teams with the ability to automate what they know and share that knowledge across multiple teams and the entire organization.
- Show team members exactly how what they do affects the end customer. For many teams, this may be their first experience ever in this area.
- Treat your automation as code and use version control. Software code is written in a common languages and protocols so everyone can read and repeat it. Your automation should be just as readable and sharable.

You can make an impact
with an automation-first approach to optimization.

KEY TAKEAWAYS

1. Make sharing a priority

When we take on and solve challenges completely by ourselves it doesn't help spread that knowledge across the organization. Ansible gives you a common language for describing your IT infrastructure and makes it easy to share solutions.

2. Empower your people

Encourage broad participation in the process of automating IT. Enable teams to automate what they know, share that knowledge and make suggestions on how the overall process of automation can be optimized.

ON DEMAND WEBINAR

Optimize Automation: Model Everything, Deploy Continuously

GET STARTED

We can help you get started on your path to automation.

Talk to an Expert

ansible.com/devops

Share this eBook:

RED HAT® ANSIBLE® Automation