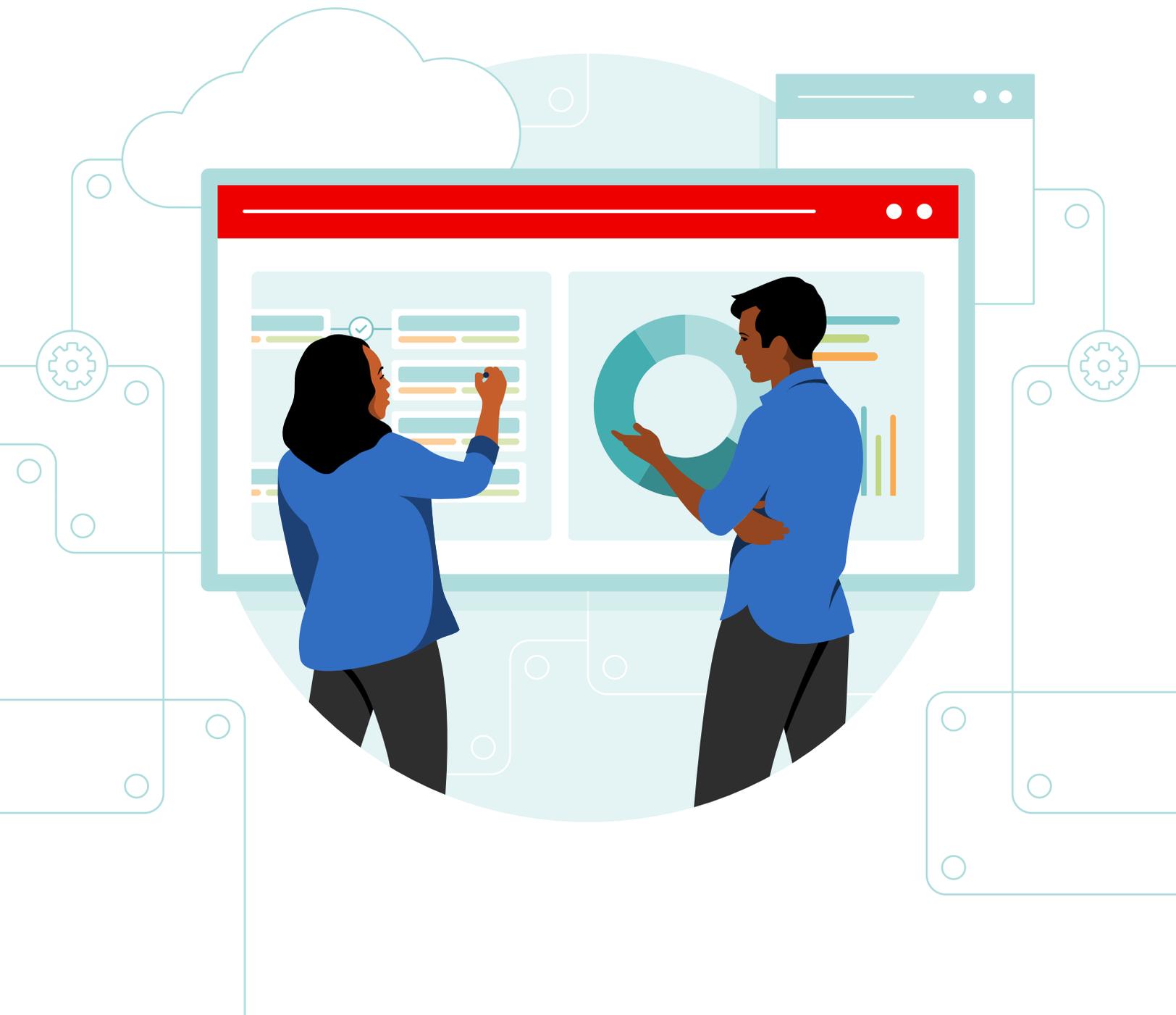


Tales from the field:
A system administrator's
guide to IT automation



3 Introduction

4 Chapter 1: Dispelling common IT automation myths

- 5 Why I was scared of IT automation
- 8 4 IT automation myths dispelled
- 11 How to automate routine tasks to avoid mistakes—and save time

14 Chapter 2: The benefits of IT automation

- 15 Numbers talk: Getting your team on board with IT automation
- 18 Use automation to combat your increased workloads
- 21 Take control of your operations: Forge a new way to work
- 24 Case study: Saving time with IT automation

27 Chapter 3: How to become an IT automation expert

- 28 The rise of the automation architect
- 32 8 skills you need to be successful in IT automation
- 37 6 ways to increase your Linux sysadmin earning profile and potential
- 41 5 ways to change your team to an automation-first mentality

44 Closing: How to get started with IT automation

- 45 Embracing IT automation: Why it's good for your career—and how you can get started
- 48 Get started



Introduction

By Emma Van Sant, Senior Program Marketing Manager, Red Hat

Welcome to *Tales from the field: A system administrator's guide to IT automation*. This compilation of short stories seeks to share the excitement, frustrations, successes, and challenges associated with incorporating IT automation into organizations and teams across the globe.

Few technologies impact organizational transformation the way that IT automation can. IT automation improves security and compliance, abstracts away complex tasks for those with limited technical knowledge, improves standardization across the organization, helps organizations scale, improves continuous delivery, and reduces operational complexity and cost. But when adopting a holistic IT automation adoption mindset and approach, these business benefits come with an equal number of challenges.

At the most fundamental layer, automation is really about people. A popular phrase within our Red Hat® team is “technology is easy, people are hard.” While this sentiment undoubtedly glosses over the technical difficulties associated with adopting a new technology, it’s clear that the phrase has merit. The results from a recent study indicate that lack of training, fear of replacement, and organizational culture were all top barriers to widespread automation adoption. Six of the 10 barriers cited in the survey were people-culture oriented barriers, not technical ones.¹ People are the key to any successful technology implementation and adoption. Without buy-in, training, support, planning, and security, any automation adoption initiative is set up for failure.

This book is a compilation of stories about these very challenges, written by Red Hat authors who have struggled with cultural, emotional, and practical barriers to holistic automation adoption within organizations. These writers have spent years either implementing automation within their organizations, or decades consulting on the implementation and adoption within other teams. This book is a good starting point as you become your organization’s IT automation expert—and explore what that means for you and your career.



Emma Van Sant

Emma is a Senior Program Marketing Manager focused on Global IT Automation at Red Hat. She’s spent the last seven years working in IT organizations in the Czech Republic, Germany, and the U.S. She’s passionate about communicating the value of using open source solutions to address our most challenging IT problems.

¹ IT Automation & Management survey, December 2020, Qualtrics and Red Hat. n=838.

Chapter 1

Dispelling common IT automation myths

- 5 Why I was scared of IT automation
- 8 4 IT automation myths dispelled
- 11 How to automate routine tasks to avoid mistakes—and save time



Why I was scared of IT automation

Learn the perspectives of three IT roles—and their common anxieties about IT automation.

By Allen Eastwood, Senior Architect, Red Hat Consulting

New technology often makes lots of promises, but experienced IT professionals are likely to greet these promises warily. There can easily be a disconnect between what the vendor's sales team is pitching as a solution and the actual problems or challenges you're trying to solve. Having anxiety about significant change is expected. In our experience working with customers, we've found that certain roles share common concerns about IT automation.

The executives

As the person who may be most directly responsible for budget decisions and whose name may be most closely tied to the expenditure, IT executives tend to be most concerned with two things. First, they'll want to know what they're going to get for the cost. And second, they'll want to understand how problems will get solved—not at a granular, technical level but what they can point to after implementation that reflects that a process has been improved or a problem remediated.

Fortunately, the nature of IT automation means these are easier concerns to address than other software solutions. The right IT automation project pinpoints the time-intensive, manual process to be automated. An accurate cost or time value can often be estimated or calculated depending on the tasks and their complexity. By working quickly to identify the different tasks that underpin the larger process, and automating those in a “building block” fashion, you can determine the improvement measure incrementally as you progress and the larger aggregate benefit once the process has been automated.

IT managers

Unlike IT executives, who are more involved in budgetary specifics, IT managers often feel more involved in the details of the purchasing decision. They're managing the work and the workload, but they sometimes lack key insights and statistics to fully understand how their employees are doing the work. As a result, when looking at automation use cases, they tend to want to do long-term waterfall planning.



But many automation use cases, particularly the more complex ones like compliance and patching, have a lot of lesser-known dependencies that arise later in the project and render that waterfall planning inaccurate and ineffective. Not being able to plan effectively is a credible source of anxiety, as is entering a project that is likely to uncover some unknowns as manual tasks gradually get replaced by automated processes.

In past consulting projects focused on automation, managers have felt exposed, forced to acknowledge that they had knowledge gaps about root causes. But it's often their day-to-day work that causes these gaps. Many managers have felt like they were constantly reacting to emergencies, dedicating staff to one issue or another, which ultimately led to feeling short-staffed and under-resourced.

The more knowledge you have about your teams and how they work, the better prepared you'll feel when starting an automation project.

IT managers can ultimately reduce some of those reactionary, disruptive management habits by automating some of the tasks that are causing the service tickets. Their focus switches to having their team use automation to more effectively serve business needs instead of pushing them to speed up manual processes.

Another way to tackle your apprehension about automation can be to delve deeper into your Git and Jira tools, which offer unprecedented visibility and statistics into who's doing what—and how they're contributing. This information can help you uncover some of the unknowns mentioned earlier, and it can help you better understand and predict how long it will take to solve an automation problem. The more knowledge you have about your teams and how they work, the better prepared you'll feel when starting an automation project.

Sysadmins and engineers

If the roles above have more abstract fears about automation, sysadmins and engineers typically have much more visceral, personal fears about adopting automation. Their typical day revolves around these manual tasks and processes, from simple to complex, time-intensive or short. Automating them, one could reason, would obviate the need to keep a person in that role (we explore this in greater detail later in this e-book). Is this a path to outsourcing to consultants? And even if you stay in your job, how do you keep it? Who's going to train you on it? How do you find new ways to deliver value through automation?

Getting away from a known process, however tedious it might be, can be a real stressor—all the more so when you're worried that getting away from that process could eliminate your



job. The truth is that sometimes personal reframing is necessary. Where you might have previously described yourself as an infrastructure engineer, you now have the opportunity to evolve into something new—an infrastructure developer. Where your manager previously looked to you to simply work manual tasks faster, you can now innovate automations and have a more direct, visible role in creating business value.

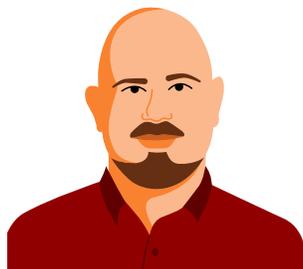
Not to downplay the efforts, knowledge, and stress it can take to make this type of shift, but it has been a proven, satisfying path for many who've taken a new interest in development practices. So, where to begin? [Red Hat Training and Certification](#) can make for a confident start, helping you build your skills while providing validation to your manager and others that you have the capabilities to deliver business value through automation.

Additionally, automation is a flexible solution—it can help you rapidly adapt to new business needs. At the same time, that flexibility can make it difficult to predict where and what you'll be automating in a year. Adapting to this flexibility, seeking out opportunities for collaborative automation, trying to understand your stack, and establishing a system of governance for playbooks can help enrich what you're doing in your own department and create a path toward enterprise automation.

Conclusion

Quelling fears about automation is a culture shift. Open mindsets can help accelerate automation initiatives and speed time-to-value by resisting some of the segmented, rivalry-based structures that can exist between teams. Transparency about knowledge gaps and a willingness to fail early can help teams ultimately deliver faster and more confidently.

→ [Read our e-book, 5 steps to automate your business.](#)



Allen Eastwood

Allen is a Senior Architect for Red Hat Consulting, where he brings over 20 years of experience. He enjoys playing ice hockey when he's not helping customers automate their IT systems.

4 IT automation myths dispelled

Identify and counter common IT automation myths.

By Kedar Vijay Kulkarni, Senior Software Engineer, Red Hat OpenShift networking

In my career as a DevOps engineer, I have spent hundreds of hours automating numerous mundane tasks. Whether you are just beginning to use automation or already have some experience, you may run into resistance due to many common IT automation myths. I'd like to address some of these myths based on my experience.

Myth 1: Automating a task takes more time and effort than it's worth

If it takes more time to automate a certain task than simply to accomplish the job manually, it is not worth automating.

You are likely to get resistance from your peers or management about automating tasks based on time savings. In reality, every job you do as an engineer is worth automating, but you have to be cognizant of the time and deliverables. When certain tasks appear to be not worth automating, I have often found that what is actually meant is that it's just not possible to automate it at this time. However, in the future, your objective should be to automate the task—you are likely to get less resistance from your team if you keep this perspective. Just make sure to communicate the automation proposal in a way that meets your immediate goals and improves future effectiveness.

Myth 2: You don't need to automate a one-time task

I only need to do this once, why should I bother automating it?

This is probably the biggest myth I have seen during my career.

Here's a real-life example: A product customer raised a specific issue, and it needed a particularly complex setup. The person working on the task asked my fellow DevOps team members and me for help. We did our best to automate the task we were asked to do. The task owner did his verification and then asked if we could keep it up and running for a little while. And that "little while" quickly went from days to weeks to months.

We then received another request to tweak a virtual machine (VM) instance, then a follow-up request to create a snapshot, and eventually a clone of the VM. Once the VM had been cleaned up by automation, they needed to set it up all over again.



Every time a new request came up, we kept asking the person to automate it, but we were met with resistance from the individual and management because they considered it a one time only task. It was technically one time only, but it had been a lot more effort than that. Had they approved more time automating it up front, they could have saved a lot of time for themselves and us.

Every time they needed to make a change, they could have noted it as part of an update to their automation and made everyone's lives easier. This is just one such example. While it is sometimes hard to know how much work will be involved in solving a problem in the beginning of a project, it is essential to step back at an appropriate time and evaluate whether you need to invest in automation—before it's too late.

Myth 3: Automation breaks, so don't waste time doing it

It is not worth investing time in maintaining automation because it breaks often.

It's true that automation breaks periodically as the various scripting languages change over time—or the system you interact with through the automation deprecates or introduces features.

But does that make your automation effort worthless? Not necessarily. I have often experienced this first-hand when we used automation to build [Red Hat Virtualization](#) environments. The automated tasks were initially designed for version 4.2, but we were soon upgrading to build on the 4.3 and 4.4 versions.

Automation maintenance empowers you with confidence.



We kept tuning and tweaking our automation and handling the various scenarios and quirks of each version. The results were a versatile combination of Red Hat Virtualization deployment automations that acted as engines that when fueled with appropriate inputs for each of the versions, produced fully deployed and configured Red Hat Virtualization environments.

Did I spend time debugging and fixing the automation? Yes. But I definitely found the effort worth it every time I had to rebuild the Red Hat Virtualization environments. We had six to seven environments, each with its own version, sizes, and other characteristics. Whenever new builds became available or an environment became stale due to abuse during testing, it just took a single button click to reprovision the resource.

Automation maintenance empowers you with confidence. While maintaining automation is time-consuming, in my experience, it has been more effective than I initially thought. And all things considered, what IT work doesn't include maintenance?



Myth 4: It is impossible to automate this task

It is tough to automate this. It just can't be done.

There are times when you are faced with specific processes that are more difficult to automate than you'd hoped. It is not uncommon for you to hear from your peers (or read on the internet) that a given task is just too tough to automate—nobody has ever done it, and it probably can't be done.

I ran into that situation early in my career when I was automating various infrastructure tasks. I lacked experience, and others told me it couldn't be done. My boss at the time still wanted me to pursue the challenge. His willingness to give me more time on the problem, acknowledging that it was tough, gave me additional motivation.

After spending about a month exploring options and trying various methods to create proofs of concept, I saw that I could "semiautomate" the task under the given constraints of the situation, meaning there were a few manual interventions. So it was indeed tough, but it was still a myth that it just couldn't be done. It takes determination and courage to go after a tricky automation problem.

Here's another perspective: If it has never been done, you invent something new. And that's what I did. The more experienced people were right: It couldn't be automated. However, what they really meant was that it couldn't be fully automated. And what are engineers if not creative problem solvers?

Conclusion

People tend to believe automation myths. It is important to understand that automation can be time consuming, but it is likely that it will be of value to you and your team. There will be times when automation breaks and fixes need to be made, but it will pay for itself by saving time and reinforcing the benefits it provides.

Sometimes it is difficult to automate tasks, but take those challenges as an opportunity to innovate and share the knowledge you gain. Lastly, it is very likely that when you do a certain task once, you will have to do it again, and if you automate it, you set yourself up for success.



Kedar Vijay Kulkarni

Kedar is a Senior Software Engineer at Red Hat, working with Red Hat OpenShift® networking. He focuses on functionality, performance, and scaling of software-defined networking. Previously, he worked extensively with Red Hat Ansible® Automation Platform, Red Hat Satellite, and Red Hat CloudForms® (upstream ManageIQ) project, primarily looking at deployment and management of internal infrastructure as a DevOps automation engineer.

How to automate routine tasks to avoid mistakes—and save time

By Shashank Nandishwar Hegde, Solutions Engineer, Red Hat, and Kedar Vijay Kulkarni, Senior Software Engineer, Red Hat OpenShift networking

The job responsibilities of engineers, sysadmins, and DevOps roles include answering dozens of emails, upgrading hundreds of servers, and patching all the systems in your datacenter or your favorite cloud. All of these tasks take a considerable amount of time to complete—and they are sometimes boring but always important. A slight mistake can open up a potential threat to the security of the entire system. For example, failing to apply a security patch can make a system vulnerable. And let's be honest, people make mistakes. But what if you could automate these mundane tasks, save some time, and simplify your team's tasks?

What tasks should be automated?

Repeatable and routine tasks

These are the tasks that need to be carried out on a regular basis. For example, we need to update my GitHub fork, including the latest changes from an upstream repository. Other examples include:

- Collecting system backup logs.
- Sending out a weekly system upgrade email.
- Carrying out upgrades.

Simple, yet time-consuming tasks

You might be required to collect system utilization in your lab or send a daily email to the team about system availability. There will be some tasks that take a lot of time for the system to complete, such as a continuous integration/continuous delivery (CI/CD) pipeline or a particular installation. It's always better to set some alerts in such cases. We use an email alert that I receive after each Jenkins pipeline completion. This way, we can keep track of progress and also use the time to do other tasks.

Complex tasks

The combination of several jobs, such as installing an operating system (OS), installing the latest packages, and making sure a weekly CI/CD pipeline is run into the system, is not complex, but think of repeating these steps for hundreds of systems in your datacenter or cloud environment. Other examples include setting up load balancers and proxy servers, or making entries for systems in your Domain Name System (DNS) infrastructure.



Automating virtual machines

One of the points of automating tasks is to get the mundane, repeatable tasks scripted and deployable. Once you accomplish this, it frees up your time to work on more high value or strategic, proactive tasks for the team.

One example of a mundane task to automate is creating virtual machine (VM) templates out of new builds of Red Hat Enterprise Linux®. Let's say we were using Red Hat Enterprise Linux 7-based VM templates, and then Red Hat Enterprise Linux 8 was released. We need to have all our templates updated with the new version. That would be a painful and error-prone task to perform by hand.

Another mundane task is provisioning VMs requested by your team members. Automation makes such tasks much more efficient. Even better, create a self-service portal that your team can use to request new VMs. If the VM has a unique configuration, you'd intercept that request and take appropriate action. Otherwise, let the automation handle VM creation and send out a notification (email or chat message) when the VM is ready.

Another important aspect people forget about is removing VMs to preserve resources. Engineers often forget to clean up their resources in a timely manner. As an admin, you could keep manually deleting VMs after asking if anyone is still using them, or just create a set of rules for the longevity of VMs. When the VM fails to comply with those rules, it gets automatically deleted.

It's never about automating yourself out of a job—it's about deciding when you have done enough of the work and want to change your focus.

Conclusion

We have given various examples of simple, repeatable tasks that are often time consuming. If properly planned, and with the help of some tools, these tasks can be completely automated and stored in a centralized repository like GitHub. By automating these steps, we can save a lot of time and increase team productivity. Most importantly, we can avoid making mistakes when a complex task is involved, which usually depends on taking input from one segment of the system and passing it on to another segment, and then moving on to completion.



It's not always necessary to automate every job. If you feel automation is difficult, you can usually start by working with small, low-risk tasks and then build up your repository by partially automating other tasks. Automation will bring a huge change to the team's work style in the long run.

In summary, it's never about automating yourself out of a job—it's about deciding when you have done enough of the work and want to change your focus.



Shashank Nandishwar Hegde

Shashank works as a Solutions Engineer at Red Hat. He works primarily with Red Hat OpenShift and Red Hat Ansible Automation Platform. He's passionate about open source software and cloud, security, and networking technologies.



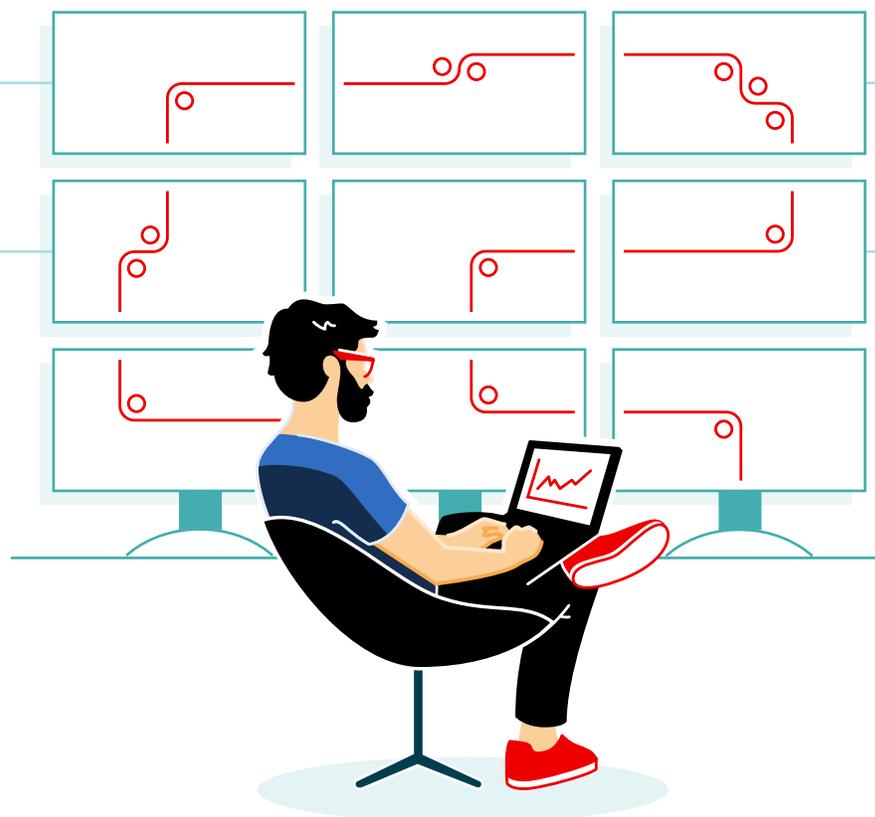
Kedar Vijay Kulkarni

Kedar is a Senior Software Engineer at Red Hat, working with Red Hat OpenShift networking. He focuses on functionality, performance, and scaling of software-defined networking. Previously, he worked extensively with Red Hat Ansible Automation Platform, Red Hat Satellite, Red Hat CloudForms (upstream ManageIQ) project, primarily looking at deployment and management of internal infrastructure as a DevOps automation engineer.

Chapter 2

The benefits of IT automation

- 15 Numbers talk: Getting your team on board with IT automation
- 18 Use automation to combat your increased workloads
- 21 Take control of your operations: Forge a new way to work
- 24 Case study: Saving time with IT automation



Numbers talk: Getting your team on board with IT automation

See how you can automate rote tasks and shift your focus to more attractive projects.

By Andrius Benokraitis, Senior Manager, Technical Marketing, Red Hat Ansible Automation Platform, and Cindy Russell, Product Marketing Manager for Red Hat Ansible Automation Platform

Automation enthusiasts are often borne out of the results that teams achieve through successful automation projects. For many of us, “sleep more” is perhaps the most compelling benefit when updates can be done consistently, correctly, and rapidly across hundreds or more systems, networks, and cloud implementations. No one really enjoys midnight calls to resolve an issue.

And when it comes to security and compliance needs, we have seen teams face pressures to resolve risks quickly. When you can create the proper fix in an automation playbook, it can be executed across all impacted solutions in the global infrastructure. This solution can be fast and effective, so the risk is remediated quickly, without the human error that can occasionally creep in when you are doing the same things repeatedly.

But there are other benefits, too, such as allowing for more time to work on key priorities that add value or the ability to advance experience and skills. Here are some statistics we’ve used when trying to get some of my teammates on board with IT automation.

Hours to minutes stories

The benefits of automation can be dramatic. There are numerous “hours to minutes” stories when the right automation solution is employed. Let’s consider a couple of them.

20 hours > 30 mins

Blue Cross and Blue Shield of North Carolina found that provisioning a single virtual machine (VM) or server took more than 20 hours of work—completed across 10 working days—and incurred charges from its managed service provider. With Red Hat Ansible Automation Platform, VM provisioning for a single server can now be completed in 30 minutes. Therefore, when they needed to quickly provision 1,000 VMs, the process was complete in fewer than 3 days. Contrast this to their non-automated time estimate of 16 weeks with 12 staff to complete the same work.²

² Red Hat case study, “Insurance provider saves 70,000 work hours with Red Hat.” May 2021.



20 minutes > less than one minute

The [City of Denver](#), without Ansible Automation Platform, required 20 minutes to create an individual Microsoft Teams environment. With Ansible Automation Platform, the same provisioning process took less than one minute. When the pandemic hit, Microsoft Teams use grew 514%, and having automation ready to help saved an enormous amount of time.³

When IT must respond rapidly to needs like these, automation makes the job fast and efficient for you. You can find more stories like these on our [case study page](#).

Automation skills in demand

Everyone is always looking to advance their skills, which is an essential need for all technical professionals. Automation is an important and emerging skill to add to your repertoire. Let's consider a few trends:

Ansible is ranked ninth on the list of loved technologies

Automation skills consistently show up in surveys and trend reports. In 2020, the [Stack Overflow Developer Survey](#) found that Ansible ranked ninth on the list of most loved tools.⁴ In the year prior, Indeed.com's [Hiring Lab report](#) found Ansible to be the third fastest-growing skill, at nearly 1,300% growth between 2014 and 2019.⁵

Automation is at the forefront of enterprise technology strategies due to the impacts of the pandemic on operations, where automation helped with remote management.

More than 40K LinkedIn jobs mention Ansible skills

Automation is at the forefront of enterprise technology strategies due to the impacts of the pandemic on operations, where automation helped with remote management. Companies saw automation's benefits first-hand, and this has accelerated the enterprise IT automation and management strategy. If you do an internet search, you will see many analyst viewpoints on the automation growth trend. This growth is also evident in available jobs. A LinkedIn search yielded around 40,000 job openings that include Ansible in their descriptions.⁶ Obviously, Ansible automation skills are sought after, making them a good skill to acquire.

Deliver projects that matter

To best support the business, we believe that cross-functional IT teams must come together to deliver on important initiatives. For example, digital transformation applications, expanded hybrid cloud platforms, modernized technology stacks, edge or Internet of

3 Red Hat case study, "[Denver supports remote work with Red Hat Ansible Automation Platform](#)." April 2021.

4 Stack Overflow. "[2020 Developer Survey](#)." Accessed July 2021.

5 Flowers, Andrew. "[Indeed Tech Skills Explorer: Fastest-Rising Tech Skills](#)," Indeed.com Hiring Lab: Nov. 26, 2019

6 LinkedIn job search for Ansible in the description yields 40,600 results as of June 23, 2021. This number is even higher for "Ansible engineer" on this date.



Things (IoT) implementations, and more. Often these initiatives will involve learning or using emerging technologies, and they help demonstrate and deliver value to key stakeholders like customers and your company’s leaders. When you have more time to focus on these advanced projects, professional development and growth can follow.

Automation can help you automate rote tasks and complete lower-value work, so you can instead focus on these more attractive projects. For example, let’s say you have to apply a patch to 500 systems to remediate a security vulnerability. Automation will dramatically reduce the time required, as we see in the customer stories above. As a result, you have much more time to focus on more innovative projects, which is good for both you and your company.

With freedom from manual tasks and plenty of automation skills, you may be interested in advancing your career as an automation architect. This handbook will help you learn more about the best practices for forming an “automation-first” community and leading your company’s automation strategy.

We hope you can see the automation benefits “by the numbers” for you and your company. We wish you the best in your learning journey, and we hope to see you at [AnsibleFest](#) as an automation enthusiast.

Recommended learning resources:

- [Red Hat training module](#) (free)
- [On-demand and upcoming webinars](#)
- [Ansible Automation Platform videos](#)
- A free trial where you can [build and run your first automation projects](#)



Andrius Benokraitis

Andrius is a Senior Manager in Technical Marketing for Red Hat Ansible Automation Platform. He brings more than 20 years in the computer software industry from companies such as IBM, Nortel, and Cumulus Networks. Andrius is skilled in network automation, enterprise Linux, business analytics, technical writing, and strategic alliances.



Cindy Russell

Cindy is a Product Marketing Manager for Red Hat Ansible Automation Platform. She brings many years of experience in developing technical practitioner and other marketing programs for software products, including automation, analytics tools, database and related machine learning technologies, and developer tools. She has worked at companies including IBM, Sun Microsystems, and several startup and smaller independent software vendor companies.

Use automation to combat your increased workload

Tired of mundane, tedious, boring tasks? Automation improves your efficiency and frees your time to focus on new and innovative opportunities.

By Ricardo Gerardi, Senior Consultant, Red Hat

As the information technology footprint grows in many corporations, so too does the role of a sysadmin. A modern sysadmin deals not only with physical machines and operating systems but also with myriad virtual systems, cloud environments, network devices, and container workloads. The complexity and number of systems managed, even in small companies, is typically one or more orders of magnitude higher than a few years ago.

Your role as a sysadmin consists of deploying, maintaining, updating, and ensuring that all these heterogeneous systems work correctly, guaranteeing that both customers and the business benefit from these technologies. No doubt, this is a dynamic job that provides plenty of opportunities for performing tasks that are challenging and exciting. It also entails learning new skills and working on stimulating projects.

However, due to the job's nature and the number of systems you manage, typically, you also have to perform several mundane tasks. Examples include applying a minor update on a group of servers or cleaning up some logs. You may be asked to report on installed software versions in preparation for an upgrade or in response to a security audit. These tasks, while simple, still take a long time to complete when combined with the number and variety of devices on which you need to perform them. Spending time with these tasks takes your time away from working on more valuable or more exciting activities.

While working on repetitive and boring tasks, have you ever felt a low sense of fulfillment or felt overwhelmed? If you have, you're not alone. I've heard that from colleagues throughout the years, and I have felt it myself.

Making a change

So what can you do to improve the situation and make better use of your time? The answer can be complex and, in some cases, require the company to address it systemically. For other problems, you can use your technical expertise to improve the way you work by automating these repetitive mundane tasks.



When we think of automating an IT process, it's normal to evaluate spending the time and effort to develop automation artifacts to address large or complex issues. There are many benefits to doing this. You can realize the same advantages by applying automation to simple and repetitive tasks, especially when you can reuse the automation to manage hundreds or thousands of systems.

Some of these benefits include:

- 1. Efficiency:** Execute tasks more efficiently across a large number of targets, allowing the computer to do the hard work for you.
- 2. Standardization:** Run tasks in a consistent way, ensuring the same results every time. For example, avoid common distraction-led mistakes where you configure one device with lower case characters, and another with one upper case character, leading to issues that are hard to troubleshoot later on.
- 3. Less error:** Fewer people touching the systems lowers the probability of configuration mistakes or accidents leading to outages. For example, have you ever missed the WHERE clause in a DELETE or UPDATE SQL query?
- 4. Personal fulfillment:** Working on more fulfilling and intellectually challenging tasks typically leads to higher job satisfaction.

How do I start?

Applying automation concepts to simpler tasks may also be a good way to get you started with automation. You can see immediate benefits by tackling quick wins while learning and preparing to automate more complex issues later. In fact, that's how I started with Ansible Automation Platform a few years ago. I used Ansible Automation Platform to replace some scripts that ran periodic checks on a few hundred servers that I managed. I spent a few hours doing the initial configuration, and after that, I could use the same environment to automate these checks and many other small tasks, performing them quickly and consistently.

In my opinion, this is a good strategy for starting your automation journey. Look for an issue or task that you perform regularly and takes much of your time or is annoying to do, select your automation solution, and apply it to solve this issue while learning the inner workings of the solution and automation as a whole.

Applying automation concepts to simpler tasks may also be a good way to get you started with automation.



If you don't know where to start, here are some ideas of tasks that are good candidates for your first automation project:

- Connectivity tests
- Packaging maintenance: Install/update/delete packages
- Reporting installed application versions
- Managing users
- Resetting user passwords
- Cleaning up unused files such as core files and logs
- Executing a backup or restore of a system or configuration

Where to go from here?

By automating simple and repetitive tasks, you can complete them more efficiently, freeing up some of your time to work on more important, or perhaps, more exciting, projects.

The automation journey may be challenging, but it's definitely one that I recommend. Give it a try, and you may collect many benefits for your business and for you personally.



Ricardo Gerardi

Ricardo Gerardi is a Senior Consultant at Red Hat Canada where he specializes in IT automation with Red Hat Ansible Automation Platform and Red Hat OpenShift. He has experience in the telecommunications sector, having worked as Senior Architect at TELUS, and had previous experience as Senior Consultant and Pre-Sales specialist for Network Management solutions at IBM Brazil and IBM Canada.



Take control of your operations: Forge a new way to work

A case study about developing tools that automate and streamline mundane tasks to improve delivery time, reduce human error, and free up more time for new work.

By Thomas Tuffin, Technical Account Manager, Red Hat

You are a system administrator in an operations team that works using agile methodologies, following a framework based on scrum, kanban, or even scrumban. Your team takes care of a constantly evolving environment. On top of the daily support tickets, there are deliverables to stakeholders that must be completed according to deadlines. The workload is demanding, and the backlog is only getting longer. This situation creates a high-pressure environment where tasks that deliver immediate value to stakeholders are considered top priority and are given the most attention. Tasks that do not provide immediate value are given a lower priority. Unfortunately, it is all too common for the development of automation tools and frameworks to fall into the low priority basket, and so your team continues to do things manually. If some of this sounds familiar, you are not alone.

Redirecting focus to developing automation is not always a straightforward matter. Hopefully, you will receive strong support from your team, and maybe positive reinforcement from management. However, you may hit some heavy resistance when it comes to prioritization—and for good reason. Your team has stakeholders to deliver to, and your primary mandate is to ensure they get what they need on time.

As an operations team, you likely have dependencies on other teams and suppliers. One delay up the chain will have a domino effect further down, thus delaying your deliveries. For teams working scrum, it is not uncommon for stories and subtasks that were not completed by the end of the sprint to spill over into future sprints. This backlog increases the pressure on your team, delaying future projects and pushing back the tasks that are primarily focused on improving your environment. Such a situation can leave a team feeling as though they have lost control, and they may feel frustrated as little attention is given to long-term solutions. Again, this is a common situation for many operations teams.

How to get started with automation

So, what can you do about it? You should, of course, begin by raising the topic with your team, product owner, or project manager, and, if necessary, your manager. Providing examples of how much time and effort will be saved by investing resources into developing



automation is one good way to get management's attention. You can begin by drafting a proposal that outlines how long tasks take the team now, how much time you think it will take to automate these functions, and how much time tasks will take after automation.

You can also start allocating some extra time for those tasks that can be automated. Begin with the smaller jobs, such as streamlining the process of adding items to your configuration management database (CMDB). A good CMDB will have an application programming interface (API), so creating a shell or Python script to take advantage of that API is an integral part of the automated workflow for deploying and managing servers. Once you have the API figured out and a script that interfaces with it, you could take it one step further and build a library that other scripts can use. A library will standardize the way your team interacts with the API. It will also prevent duplicate development effort since all future scripts can use this library whenever they need to access the API. It may be a little bit more work up front, but it will save time and effort in the future development of scripts and tools.

Armed with a library with standardized access to the CMDB API and a script that can pull asset information, you can begin automating parts of your server deployment process. If you already use preboot execution environment (PXE) and kickstart files as part of your deployment procedure, why not automate most of the work? You could achieve this through scripts written in a language of your choice or by using [Ansible Playbooks](#). By employing some of Ansible's built-in modules, such as the template module, you can easily generate PXE boot files and kickstart files with data pulled from the CMDB. Ansible also has modules available for connecting to various manufacturer's out-of-band management interfaces, so tasks like setting out-of-band hostnames, IP addresses, and power management are straightforward.

You can, of course, achieve the same result through other tools, custom scripts, and various open source and proprietary software solutions. There's more than one way to approach the challenge. The same is certainly true for server deployment. Use the skills and tools at your disposal to streamline processes and reduce time spent on repetitive tasks.

Use the skills and tools at your disposal to streamline processes and reduce time spent on repetitive tasks.

With a few of the smaller tasks now automated, you should start to see some positive effects. For one, you and your team no longer need to click through that clunky CMDB interface to input and retrieve information. In addition, part of your server deployment process is now automated. Finally, there is another big benefit of automation—a reduction in issues due to human error. It is well known in the IT industry that human error is a frequent cause of various system failures, outages, and security breaches.⁷ Automating even the most simple tasks, especially the mundane ones, will help to alleviate this.



Taking back control of your environment

As more of your team starts to get involved in the development of automation tools, you will find that the ecosystem of tools grows organically. This growth will raise other challenges, such as code quality, maintainability, and responsibility. It is important to establish guidelines early on to ensure certain criteria are met and ways of working are followed. You can achieve part of this by using a version control system and tools such as Git and Gerrit to track changes and support code collaboration within your team.

Once you have put together a healthy toolset to take care of the repetitive tasks, you will find that your team defaults to using automation. One of the first questions they most likely will ask when planning something new will be, "Can we automate this?" The more automation you implement, the more likely you are to change the status quo and take back control of your environment. Developing tools that automate and streamline mundane tasks will not only improve delivery time to stakeholders and reduce human error, but it will also allow you and your team more time to focus on working on the future of your environment.



Thomas Tuffin

Thomas is a Technical Account Manager for Red Hat. An Aussie expat in Sweden, he is passionate about open source software and has a keen interest in emerging technologies such as blockchain. Always looking to discover something (or somewhere) new, Thomas prefers to either be traveling or working on a project.

Case study: Saving time with IT automation

Create time to work on new, exciting things and see how automation can make tasks go from days to minutes.

By Kedar Vijay Kulkarni, Senior Software Engineer, Red Hat OpenShift networking

In 2020, I was working on a team automating the process of creating new VM images for the latest [Red Hat Satellite](#) builds. Our goal was to automate VM deployments, snapshots, cleanup, and template creation. It sounds easy, but it was a lot of work. Automation was obviously needed to save time for our team, and we picked Red Hat Ansible Automation Platform as the automation interface. That's where this story begins.

If you have worked with [Red Hat Ansible Automation Platform](#), you know that there are many things one needs to configure to make it useful. For example, you need to have login and authentication set up, then projects, credentials, inventories, inventory sources, job and workflow templates, notifications, schedules, and so on. All that work is why I helped create Red Hat Ansible Automation Platform [Configuration-as-Code ATCasC](#).

This automation project transforms all the things you need to do in the Ansible Automation Platform user interface to the YAML serialization language. The settings are then executed with a single playbook command that takes your entire Ansible Automation Platform from a fresh install to a fully functional service.

This is a huge win. Why? Once the configuration is written, the time it takes to stand up a new instance using the Configuration-as-Code method is less than 30 minutes. Prior to using this approach, it took us a day or longer (depending on who you asked to do it and their level of expertise) to deploy, set up, and configure a new instance and make it production-ready.

Before I developed the Config-as-Code method, a manual deployment could take 1-3 hours, and the configuration could take the rest of the day. It would likely be a two-person effort to speed up the process. For example, if there was one project with five credentials, two inventories, two inventory sources, 20-40 job templates, and 5-10 workflows, it could take hours to create them via a mouse-driven user interface. Let's say you did it once—painfully. What happens if you lose your instance? If you don't have any configuration written, reproducing this is based purely on your memory or team documentation.



This is why we found it essential to write the configuration first. Getting the configuration written correctly became a learning opportunity for my team. Why? The configuration didn't have a standard programming language, so the team had to learn the schema for YAML constructs. Once we got past that learning curve, we became more efficient.

Now that we had automated setup time, we were confident we would be back up and running in no time with the proper config files when a disaster happened. But what about getting to fully completed and tested YAML configurations?

To put this challenge into perspective, if you write a new playbook that runs in Ansible Automation Platform as a job template, you need to add appropriate projects to the project's YAML file, then add the proper credentials, inventories, and job templates in the correct files. This is a minimum of about 50 lines of code. To figure out this code and write it can take between 30 minutes (if you know what you are doing) to 3-4 hours (if you are new).

The process of writing the code only gets quicker as you get more practice. It is worth the time spent, though, as you get repeatability and consistency. You can apply all the pros and cons of [Infrastructure-as-Code](#).

Next, you want to have your configuration (code) tested. This is where my team spent another couple of hours standing up a test instance that looked like the production instance and contained all the proposed changes. Next, we'd figure out what jobs were needed to test the merge request fully. Finally, we would merge it. This burden was a total workload of a day or two.

To tackle testing with automation, we devised an automated approach using GitLab continuous integration (CI). With our automation, every time a new pull request (PR) is opened, GitLab CI would create a new test instance for that PR. Automation saved 2-4 hours depending on who was tasked with deploying the instance. Now that GitLab was deploying it, more time was saved.

The next challenge was figuring out how the pull request should be tested. With some smaller PRs it was easy enough to quickly figure out what to test. Complex PRs touch more than a dozen files, and it was tough to anticipate what might break if the PR was not properly tested before merging to the master branch. Keep in mind, the production instance was running based on the code on the master branch.

To get over that challenge, saving hours spent analyzing and then testing PRs, we devised a new project called Ansible Genealogist, which examines PRs in minutes and documents what needs to be tested.

The process of writing the code only gets quicker as you get more practice.





Task	Time spent manually	Time spent using automation ⁸
Deploying a new Ansible Automation Platform instance—production ready	~1-2 days	<30-45mins
Deploying and configuring a test instance for testing new configuration before pushing it to production	~4-6 hours	<30-45 mins
Determining what needs to be tested for each new PR	~1-2 hours	<5-10 mins
Run tests	~2-6 hrs (or longer for complex PRs)	<5 mins (just fire the automation test script and come back later to check outcomes)
Redeploy a production instance as you just lost the one you had running in the production due to some outage	No estimate, it's a disaster, all hands on deck (maybe ~1-2 days if your team members know what to do and manage to get it all done)	<30-45 mins
Make changes to production, such as adding a new job template or updating a credential	Dreadful task—if you do something wrong, it is bad. If you decide to test changes before updating production, then you are looking at ~1 day of work	<30 mins—as your changes would be already tested as part of the PR process, pushing to production is essentially CD of CI/CD

As you can see, through automation, we made tasks go from days to minutes. And no, we didn't automate ourselves out of a job because we kept getting more tasks to automate. The goal of our group was to automate standard administrative tasks for virtual machines—deployments, templates, snapshots, etc. Time savings was a critical part of this project. We also wanted to create repeatable events in case of disasters. Automation and templates gave us the opportunity to be far more efficient in disaster recovery situations.



Kedar Vijay Kulkarni

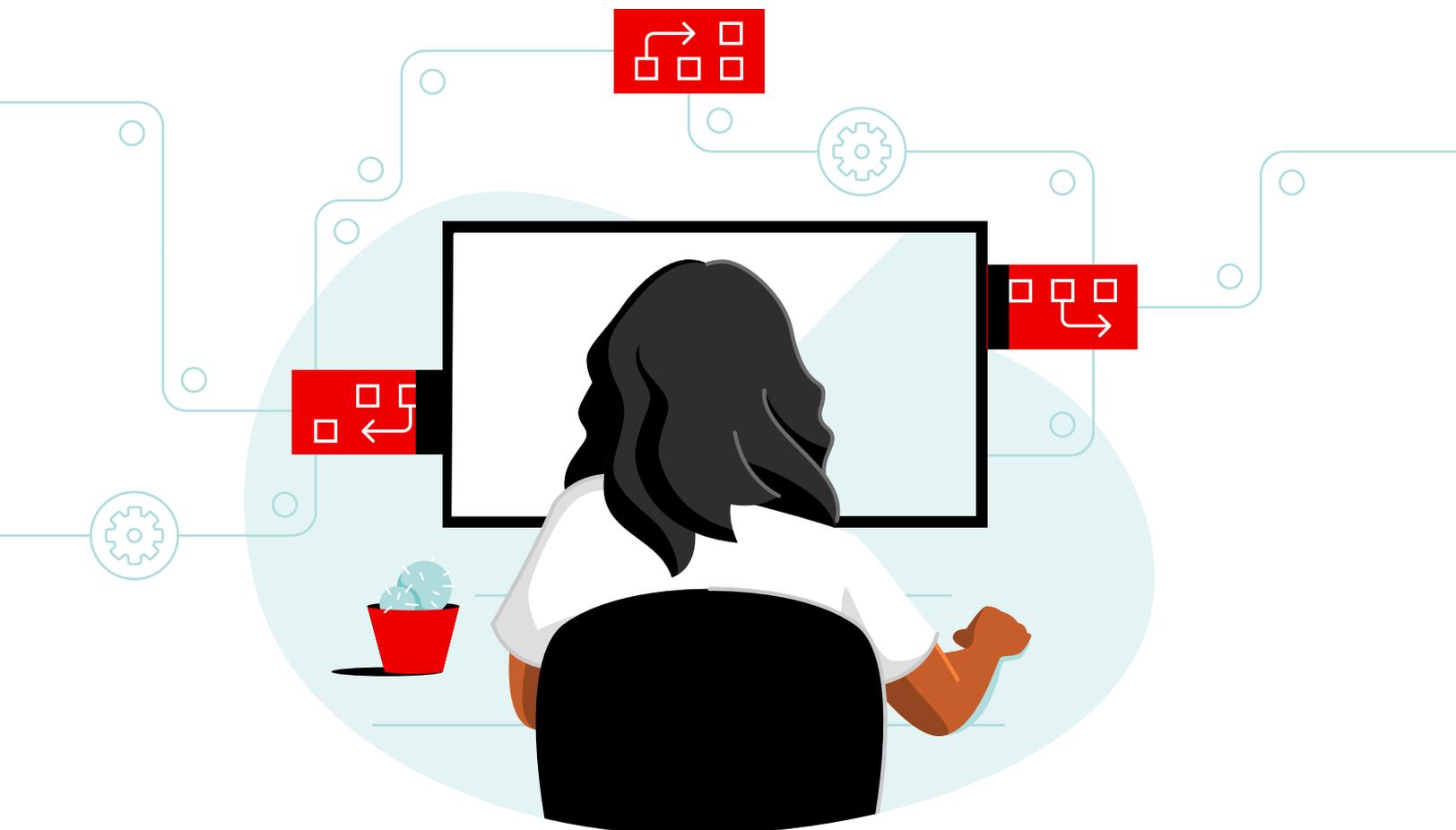
Kedar is a Senior Software Engineer at Red Hat working with Red Hat OpenShift networking. He focuses on functionality, performance, and scaling of software-defined networking. Previously he has worked extensively with Red Hat Ansible Automation Platform, Red Hat Satellite, and Red Hat CloudForms (upstream ManageIQ) project, primarily looking at deployment and management of internal infrastructure as a DevOps automation engineer.

⁸ Author's data and observations

Chapter 3

How to become an IT automation expert

- 28 The rise of the automation architect
- 32 8 skills you need to be successful in IT automation
- 37 6 ways to increase your Linux sysadmin earning profile and potential
- 41 5 ways to change your team to an automation-first mentality



The rise of the automation architect

Use these tips to advance your IT career and establish yourself as an automation architect.

By Joseph Tejal, Technical Account Manager, Red Hat

Most of us have seen it unfold before our very eyes—coming out from the smoke and darkness of automation challenges, there’s a new role evolving—the automation architect.

Automation was a hot topic starting in 2015—open source projects like Ansible were on the rise and becoming more popular. There was a lot of excitement and interest but there was also plenty of skepticism and much trepidation in operation centers. Fast forward to today, and automation is now at the core of technology strategies for most organizations. And the landscape continues to evolve and become more complex.

This [Forrester survey](#) about boosting innovation with automation shows that firms prioritize automation initiatives over an array of competing goals.⁹ Companies have recognized the importance automation holds for their plans and are seeking to automate all aspects of their business—someone needs to lead this effort for them to be successful.

Now, let’s go back in time and reflect on how automation architecture advanced to what it is today.

Starting an automation journey

In the beginning, operations teams and individuals found automation fun and easy. They learned how to automate some daily and routine tasks even if most didn’t have a development background. Easy-to-learn tools like Ansible made it easy for most eager squads. Everyone from systems administrators to infrastructure and database engineers and even our service desk folks learned to automate.

Interest was growing, from the small coffee talks to hallway conversations to the practical demo to convince management. We were provisioning machines, restarting applications, performing patching and maintenance, and the list kept growing. People started to collaborate, form ideas, talk about their cool stuff running on their space and how they can all work together.

But when they began to link small individual pieces of automation together, teams started to encounter problems. You probably know why.

9 Forrester Consulting, sponsored by Red Hat. “Enterprise open source innovation drives automation.” July 2020.



Facing the challenges

Soon, things became more complex. There were more use cases to cover, and now automation routines and workflows needed to be interlinked. We began to see the challenges, and even worse, some individuals became victims of their own success by getting ever more significant requirements from management.

Here are some of the issues you may face with automation:

- Competing priorities and goals
- Lack of standards, policies, and governance
- Bottlenecks due to segmented work culture
- Absence of some required skills
- Implementation security concerns
- Lack of consideration for reusability, scalability, and control

Many operations teams turn these challenges into opportunities. When first presented with some of these challenges, people feel confused and start to lose direction due to the lack of ownership and accountability. However, these challenges present an opportunity to take a leadership position to build successful automation practices, no matter how complicated the environments and tasks may get.

How the automation architect saves the day

The road to a holistic automation approach needs a driver who can navigate essential competencies for success as outlined in this [Forrester research](#) about infrastructure automation maturity.¹⁰

These competencies include:

- Strategy
- Process
- Prioritization
- People
- Structure
- Operations

We need an automation architect to lead the organization on a journey so it knows which direction it should be going—and how to get there successfully. The automation architect's role is to address the challenges I mentioned above to help everyone get the full value of automation.

Some of the crucial tasks required to steer the organization effectively include the following, as discussed in [Red Hat's automation architect's handbook](#):

¹⁰ Forrester Research. "Gauge your infrastructure automation maturity." Aug. 2019.



Understand the current state of automation and set goals

This process gives the architect a good view of what existing tools and capabilities may be useful—and what else is required. It's imperative that you set achievable goals and identify priorities, which align to keep everyone on track and moving in the right direction—even if there are setbacks and detours. This effort also allows everyone to focus on the essential tasks ahead with the goals in mind.

Promote unity and collaboration

There will be continuing gaps, differing opinions, and conflicting working styles. The automation architect should bring everyone together by focusing and aligning the efforts to the business objectives and priorities. These efforts should translate into well-defined requirements that everyone can refer to and discuss openly. Use an open, collaborative approach and tools that encourage the team to work together. Incorporate activities to break down segmented groups and celebrate collective and inclusive success.

The automation architect should bring everyone together by focusing and aligning the efforts to the business objectives and priorities.

Improve automation assets and capabilities

Once you map out an automation inventory, centralize the resources in a core repository so everyone can start working collectively on these resources. Setting standards, proper review, and version control focusing on security and best practices provides a reliable method across teams. The team can now start working on capability gaps based on these tools and workflows using learning styles that are effective based on your team structure. There are now many free resources available to learn across various tools.

Shift everyone to advance the automation mindset and culture

Building automation advocates and champions is key to creating a holistic approach and bringing a shared understanding of how automation works and benefits teams. The automation architect should find creative ways to build trust, enthusiasm, and interest in automation. This approach can range from brown bag sessions, success dashboards, and demos, to a code-a-thon challenge that brings fun to the table.

Scale automation solutions with proper governance and management

A good framework allows automation to scale and grow within teams without sacrificing security and best practices. The automation architect must establish governance adhering to the organization's policies, making people accountable through a solution with controls and robust auditing, while giving everyone the chance to succeed and innovate within properly managed and identified risks.



The established automation architect

It's inspiring to see the rise of new roles such as automation architects in this fast-paced era of evolving complex technologies. It's a great opportunity for sysadmins, subject matter experts (SMEs), site reliability engineers (SREs), and engineers to make the most of their time across operations and automation journeys. It encourages them to level up and widen their horizons by going outside of their constrained fields of expertise through the power of automation. They become strategically positioned to understand and analyze the existing organizational challenges and pains. Then they can show how they address these issues by aligning solutions with business goals—collaborating with others and using the best approach and tools for successful end-to-end automation.

To move forward in this role, you need to continuously grow and educate yourself so you can understand automation problems better and approach them with the right solutions. To learn more, read [The automation architect's handbook](#).



Joseph Tejal

Joseph is a Technical Account Manager at Red Hat. He is a Unix systems administrator at heart who participates in service improvement initiatives and automates mundane tasks using Ansible. He worked on automation project deliveries with a number of clients, and has been actively co-organizing Ansible and OpenShift meetups in New Zealand.

8 skills you need to be successful in IT automation

Advance your automation skills with scripting, collaboration, source code management, and more.

By Chad Ferman, Senior Application Deployment Solutions Architect, Red Hat

Looking back now, I was lucky . . . I started working in an IT shop before the year 2000 that already had an automation team. This team had some sort of magic that proactively fixed issues before they happened—or at least addressed them when they happened without waking someone up in the middle of the night. As a 19-year-old, I had no idea that this was not the norm or really what automation even meant. Then, one day it hit me: This is how I can make time for all of the other things in my backlog of work, and even more importantly, I don't have to do this manually ever again. The realization that anything you can do on a command line could easily be saved as code and run again systematically without human intervention completely changed my life and set me on the path I am on today.

The power of automation

In my last role, as the architect responsible for the automation strategy for an entire company, my mantra was: Automation is not just scripting. There is so much more to automation, so let's talk about that a little bit first. Yes, scripts are the basis for automation, but they are not the only piece of it. To make something repeatable, you need information coming in to tell you what state your systems are in and if they are behaving as they should be. This is where observability and monitoring come in. They let you make informed decisions about what needs to be done programmatically to accomplish your end goal. Once you have a feedback loop of information coming in and automation going out, you have a continuous cycle of improvement for your service delivery.

Why should I add these skills to be successful in my career?

Automation powers everything from application development to infrastructure deployment to business processes. The opportunities to add value are endless. You can work as a DevOps engineer, site reliability engineer, agile coach, product owner, integration engineer, artificial intelligence (AI)/machine learning (ML) operations, or business process engineer, just to name a few. Understanding how systems talk to each other to provide business value is a sought-after skill in many industries, and if you are a person who likes to create repeatable processes that work autonomously, then this is the job for you.



Doing more work with fewer people is something many organizations are dealing with, especially in economic downturns. By eliminating manual tasks, you make time for improvement. This approach builds trust with management through reliability and timely resolution of unexpected downtime. Successful patterns of automation can help everyone understand and unite around a common goal.

For example, I was given greater responsibility and opportunity through demonstrating successful patterns in automation. In a previous company, I was promoted from DevOps engineer to enterprise architect in three years by improving existing processes—we went from deploying servers in months to providing full stack application servers in 30 minutes. This was end-to-end deployment of business applications, configured, running, and providing business value.

Automation can be extremely rewarding because it's fun. It's a thrill to watch a 40-step CI/CD pipeline run and validate security and APIs, perform code analysis and linting, confirm that user interface elements are in the correct places, and perform regression tests that show the status of the pipeline in a dashboard as green/successful.

Knowing that what you do is valuable and repeatable by anyone else you work with is a great feeling. Also knowing that your changes will not affect others, and their changes will not break your work is a great feeling. Would you rather watch TV or act out a play yourself? You can relax and know that things will work as intended, or if they fail tests, let you know what happened so you can fix it later.

What skills do you need for automation?

Have you heard of minimal viable skills (MVS) for automation? These skills include, but are not limited to scripting, collaboration, source code management, Kubernetes, security, testing, observability, monitoring, and network awareness.

Scripting

Having the ability to script proficiently with your platform's built-in language (PowerShell for Windows or bash for Linux) is a great place to start. However, once you get into more complex automation, understanding a universal language like Python is desirable. I only call out Python as it has become more of a de facto standard for networking, server, storage, and AI/ML over the past 10 years. Entire automation frameworks have been written in it.

Collaboration

Scripting I get, but why collaboration? It takes multiple teams working together to enable true end-to-end automation. Very few people know how the network, storage, firewall, proxy, etc., really work, so there needs to be a commonality to tie all of these things together. This is why it is vital to have a company strategy for how and where you store your



automation code and how you deploy it. Without a standard place to share not only code but architecture designs and APIs to communicate with each different part of the architecture, it is impossible to properly automate the delivery of infrastructure, applications, and services for our customers.

Source code management

Centralizing all code into a **Git** management tool such as GitHub, GitLab, Azure DevOps, or Bitbucket will make it much easier to collaborate with other teams and people in your group. Getting comfortable with putting in issues if you find a bug and documenting it well (please not, "It's broken") is a great way to get started if you are not ready to start committing code. Once you are comfortable with issues, begin making some pull requests and commit fixes to code or review pull requests that others have put in so you can help test their functionality. People really appreciate code reviews as no one is perfect, and you may see something they didn't consider.

Application programming interfaces (APIs)

Creating a centralized catalog of APIs and playbooks that everyone works from is crucial to automation success. This skill is more than simply having APIs in your applications. Having them available for anyone to use without calling you to ask how to interface with your service is the best way to automate service delivery. This way, when a developer needs a traditional infrastructure service, it becomes exactly like a cloud resource that they can request without having to put in a ticket or pick up the phone.

Containers and Kubernetes

Containers and Kubernetes have become the de facto way to deploy modern applications across a hybrid cloud. Having a solid understanding of how to build a container and then deploy, scale, monitor, and redeploy it is highly sought after by companies. This skill applies across many different parts of companies, from machine learning and application development to business intelligence and cybersecurity. With containers, you can ensure components that run on your local machine will work exactly the same on any platform at the application level. When you take the next step to deploy the container with Kubernetes, you can ensure via code that everything you need is in place for seamless deployments across all environments.

Network awareness

The way I always start the network awareness conversation is that the cloud does not solve physics, i.e., the speed of light. It is critical to understand where data and users are located and where the data computation happens. First, understand the end-user's location compared to the data that they will be accessing. Make sure processing is in the same place as the user. Failing to do so is a common mistake that we see repeatedly that causes



the application to perform poorly. Of course, the application gets blamed, not its poor placement. Taking latency into consideration is another thing we take for granted, especially if you are in a country with good bandwidth. Make sure you conduct round-trip tests for latency to see what the end user's experience will be. For example, I had an application once that someone wanted to build in Texas even though all of the end users were in Singapore. After much deliberation, the application was replatformed in Singapore, and the user experience to get what they needed from the application went from minutes to seconds.

Testing

Testing is another skill that is dismissed as nice to have, but this is the thing that saves you from the one bad keystroke that takes down a production environment. Validating that the things you have set in motion perform as you expect them to is extremely important to being successful and making sure that you do not cause unintended consequences that will have you working through the night to resolve an issue. This should not be limited to back-end testing. There are great tools to test and validate user interface (UI) elements and APIs to make sure changes do not affect existing functionality or end-user experience.

Security

Building security into your application is crucial in today's world of ransomware and bad actors taking over cloud deployments to mine cryptocurrency. Security integration should be part of the CI/CD pipeline that deploys the application. Within this pipeline, there are key things you need: static code analysis, artifact management and tracking, secure libraries, and code signing to make sure that when it is deployed, it is the same code or artifact that you think it is. CI/CD only covers the instantiation of the application. You also need to have security hardening on the platforms you are deploying to. In addition, you want something validating that you are not running a library that has a known vulnerability in it—and can alert your team if the library needs to be patched.

With cybercrime on the rise, security testing is another role many companies are hiring for.

With cybercrime on the rise, security testing is another role many companies are hiring for. Building security testing into your application supply chain is becoming something that many are doing throughout the life cycle, from build to deploy and runtime validation that the code is executing only what it should be. Signed libraries and executables are becoming the norm, as is the validation of the sources of libraries and artifacts like containers. Using trusted signed libraries and containers and providing your organization with a custom library and artifact repository has become a standard for any security conscious organization.



Observability and monitoring

Understanding the application state and how it achieved that state is another skill needed to automate tasks properly. Unless you know what is happening with your service, it is impossible to create proactive automation to fix issues or apply a consistent state that avoids the problem in the future. Most people stop at monitoring, and then if there is an issue, use a root cause analysis (RCA) to discover what happened. Observability provides the tooling you need for an RCA, so you always have it and know what is happening well beyond the up or down status that monitoring usually leaves us with. A great resource to learn more about this is the [DevOps monitoring guide](#).

Wrap up

I believe automation is one of the most rewarding jobs a person can have in the modern IT world. It requires big-picture thinking and an understanding of how things work end to end. If you are a tinkerer and can't just be told, "That's how it works," this is the job for you. Every time you take something that people do manually and make it into a repeatable process so they can focus on more valuable work, you save your company money (we all know a lot of the time this is what it really comes down to), as well as help people work on much more exciting projects. All of the skills listed above build upon each other to help you become a better automation expert. These skills are not acquired all at once, so take your time, enjoy the ride, and stop doing things manually.



Chad Ferman

Chad is a Senior Application Deployment Solutions Architect at Red Hat. He has worked in enterprise IT in public and private sectors in retail and oil and gas for more than 25 years. His roles have covered everything from infrastructure operations to microservices application development and enterprise strategy. He recently came to Red Hat to help customers be successful in their enterprise software deployments and their cultural transformation into modern methodologies.

6 ways to increase your Linux sysadmin earning profile and potential

If you need a career or salary boost, here are some great tips to put you on the right track.

By Joseph Tejal, Technical Account Manager, Red Hat

Introduction

It's that time of year—you're sitting with your manager about to discuss your performance. The question is—are you prepared to talk through this opportunity to increase your salary or be a candidate for promotion?

The best-case scenario is that you don't have to do much talking. Your achievements and the value you've added to your organization speaks for itself, and your manager thanks you for making his or her life easier, justifying your advancement. This is just one way you increase your earning potential as sysadmins—by growing within your organization.

When other opportunities come from outside your workplace, are you prepared with your success stories to convince potential employers and get them to buy your pitch during the interview? The best case is that you effortlessly share your initiatives and successes, and they listen with interest and want to hear more—then end up hiring you with a good offer. Better yet, your profile and brand in the local community are so outstanding that different companies battle it out to win you.

These are some of the possible ways to increase your earning potential as a sysadmin. The big question is—how do you get there and prepare yourself for these conversations and opportunities?

Here, I share some of the tips, advice, and ideas based on my experiences and those of others I've spoken to.

Make yourself dispensable

I know this sounds counter intuitive, but I've learned that to move forward, you need to leave your existing tasks behind and focus on adding value.

Add value to your team and organization—*improve, automate*, and document your day-to-day tasks so that anyone can do it. Better yet, so that no one needs to do it. Target to fix your main pain points to make everyone confident in your space. This way, you have time to participate and gain everyone's trust for you to become involved in more valuable initiatives.



Don't stick to the status quo—challenge yourself and the norms, especially if they're not efficient and already outdated. Your stakeholders will be impressed with the value and improvements you're making.

Take control of your career and goals

Managing your career and goals allows you to conquer your own limits—and your organization's. You sometimes hear that you don't get the support you need, but with self-initiative, you can learn and move forward through the resources available to you. There are numerous open source projects that you can experiment with even before you request that technical training.

Do the research and proof-of-concept studies on new technologies such as [Red Hat Ansible Automation Platform](#), hybrid cloud, Kubernetes, and [Red Hat OpenShift](#) using the free trials and workshops available online. These resources help you become a good candidate for future learning and development investment by your company, helping you qualify for formal training and [certification exams](#).

You may want to find a learning buddy, technical coach, or mentor as it may be easier to commit when you're accountable to someone else. It's not just technical development—you might also want to grow vertically into leadership and manage other sysadmins.

It's not just technical development—you might also want to grow vertically into leadership and manage other sysadmins.

Work on your soft skills

In this era, having technical skills is not the only way to increase your earning potential. Emotional intelligence, attitude, the ability to work with others, and communication skills are some of the things that can give you a unique advantage and distinction.

Begin by looking for a peer or coach that you respect in your organization. You can also join speaking clubs such as Toastmasters to enhance your interpersonal and communication skills in the meeting room and on a speaking stage. Writing is also one way of getting noticed. There are many venues, such as [Enable Sysadmin](#), [opensource.com](#), and [medium.com](#), where you can contribute and learn from others.

Don't be overwhelmed or afraid to get out of your comfort zone. I know many people who surprised themselves with their strengths away from their keyboard, terminals, and technical expertise.



Collaborate with others

Gone are the days when some sysadmins know the top secrets and gain advantages, and everyone else remains left out of critical knowledge.

We live in a world where innovation springs out of collaboration. Be sure that you work with other teams to expand your knowledge and contribute to theirs—you might even take the initiative to do rotations on teams that interest you. Try to organize small projects that improve some processes within teams or explore updated tools and technology that may be better than what you currently have. This collaboration helps to break down divisions, start a cultural change, and spark interests—and it allows everyone to succeed.

Ensure everyone's work and progress are documented publicly for easy access to those who were encouraged to participate.

Connect with others and increase your profile

Career connections and references give you an advantage. Great feedback from key people you've worked with gives your manager an idea of how you perform and exceed expectations.

Being part of communities also allows you to connect with different people who may be a potential employer. **Attend or organize events**—lunch and learns or meetups within your company or local community. Getting yourself out there helps others and helps you get noticed.

You might even start with small groups or brown-bag sessions within your team or department to discuss your successes and exciting projects. Once comfortable, you might want to submit papers or proposals to tech conferences and engagements about the cool things you are working on—this further raises your profile.

Have fun at work

Find the things that fuel you. Work can be daunting at times—but it's how you react to difficult times that define you. If you're enjoying your career or significant aspects of it, you're excited to be in these situations where you can help and add value.

Make it fun and look for opportunities to display your skills and strengths, and include some opportunities to develop your weaker points. Work isn't boring when you expand your horizons and discover newfound enthusiasm.

There are many ways to make it exciting by connecting with others, being creative in your space, and reaching out for opportunities outside of your normal routine. Make sure to take recharge days outside of work to renew your energy.



The future is bright through collaboration

Your ability to earn and sustain yourself is one of the motivations for why you work. Increasing this earning potential alongside career growth, personal fulfillment, and enjoyment gives you the impetus to go further. Make sure you have a success narrative ready to share with others. But keep in mind, it's not an overnight process. It's all of the hard work, investment, commitment, and enthusiasm that you put in daily with your goals in mind.

Take every chance you can to build your brand and story, so when the opportunity comes, the impact of your narrative speaks for itself and helps guarantee your success.

Adopted from "6 ways to increase your Linux sysadmin earning profile and potential" published on Enable Sysadmin, published under a Creative Commons 4.0 BY-SA license available at: <https://www.redhat.com/sysadmin/increasing-earning-potential>.



Joseph Tejal

Joseph is a Technical Account Manager at Red Hat. He is a Unix systems administrator at heart who participates in service improvement initiatives and automates mundane tasks using Ansible. He worked on automation project deliveries with a number of clients and has been actively co-organizing Ansible and OpenShift meetups in New Zealand.

5 ways to change your team to an automation-first mentality

DevSecOps can provide a competitive edge for your organization. Use these five strategies to get started.

By Allen Eastwood, Senior Architect for Red Hat Consulting, and Larry Spangler, Principal Solution Manager, Red Hat Services Portfolio Management team

An automation-first mentality is likely a significant transformation for any organization, typically starting with task automation, moving to complex workflow orchestration, and ultimately innovating intelligent operations and “push-button” end-user services. It represents a solid commitment for DevSecOps—acknowledging the competitive edge this type of cultural change can provide. But getting there, and finding and building the necessary support for it, are real challenges—even when there’s been some initial success running automations in individual departments.

Win early and often

Although it can be tempting to try and automate the most important, most time-consuming manual processes that live within your domain from the outset, there’s a lot of merit to seeking smaller wins early in the process. Automating the individual pieces of a more extensive process to start gives you the building blocks for more complex automations later and lets you showcase the value of automation early.

Progress should be iterative, and you should consider each iteration you’ll include and what it will feature. It should feel like you’re doing developer work. By increasing scope and capabilities iteratively, you’re building to larger goals while also benefiting from automations you’ve just built and tested, meaning you can build confidently.

Having these wins to point to helps you evangelize the automation initiative with your team, your manager, and other teams.

Embrace upskilling and third-party support

Training and certification are critical to all aspects of adopting an automation-first mentality. Not only is it key to helping your team to deploy automations confidently, but it also helps build momentum for broader automation adoption across your organization. You can benefit from a “train-the-trainer” approach that empowers other teams while ensuring compliance with the established automation standards and governance.



Many organizations use strategic outside consulting engagements to help **accelerate the automation adoption process**. That can be unnerving to IT professionals within the organization—fearing this represents permanent outsourcing or job loss. But good consulting engagements have measurable, finite goals that enable and empower these teams through a mentoring relationship—leaving them with automation in production and the capabilities to continue work on increasingly complex projects and use cases.

Find your champion

Large-scale change can be daunting, and initiatives like enterprise automation require someone who relishes the opportunity to overcome the natural resistance that occurs with this type of transformation. Frequently this is someone relatively new to the organization, and often they have a mandate to “go out in the wild” and tackle problems. Maybe this person is you. At any rate, they are a necessary force to get the idea started and begin spreading it across teams, even teams that may be reluctant to work closely together—like development and operations.

Getting to an automation-first mentality requires collaboration.

Establish governance and standards

An automation-first mentality assumes a readiness to automate new tasks and processes. This means establishing a source of truth, documenting as you go so that other teams can securely benefit from your work. You’ll want a way to track code and changes—separating your automation code from your information and data model—securing privileged information with trusted tools, and avoiding file shares to distribute the source code. By using code as your documentation, not only do you foster collaboration, but you also transform the organization to have repeatable automation practices that you can share across the enterprise.

Beyond centrally managing automation code, it is also essential to define and evolve standards to integrate and orchestrate automation across teams and projects more readily. As broad adoption and reuse increase, look to establish a governing group to address cross-team challenges. These typically include standard tooling, requirements for automated testing, and best practices for deploying or rolling back changes.

Create a community and collaborate

Getting to an automation-first mentality requires collaboration. This reality can put off many people, who may greet the word with skepticism. But the root of DevSecOps is having these teams together at the table, whether it’s fun or not. The process isn’t magic, and it will not solve all problems, but the result is complex orchestrations—including a comprehensive standard and a process for governing the automations. These automations should focus on actual pain points within your organization, and they should provide real, measurable business value to the teams in play.

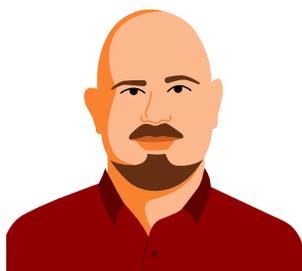


One way to make collaboration faster and more effective is to establish an automation community within your organization. This community should be the core automation team and include people from across the organization. Their mission is to share approaches, address challenges, and help other teams adopt automation successfully. They do this by helping engage these other teams, showing them the successes and possibilities, guiding them on the standard practices, and generally sharing knowledge (and playbooks) that help others solve problems and accelerate their use of automation.

Where does it end?

Automation is most powerful when it's focused on business impacts. Those impacts may be internal productivity (like push-button environments) or customer-facing (like faster customer response times). As automation gains broader adoption within your organization, you're likely to find that to continue the evolution and increase value, workflows and fundamental approaches need reinventing. For example, processes that used to have approvals that sat midstream are rebuilt to be fully automated. Rather than strictly looking at the tasks to automate, you're building a platform that lets you continually optimize workflows and processes far beyond replacing a manual task with an automated one. You're building a platform capable of performing event-driven, automated remediation, automating compliance through patching that minimizes downtime and customer service interruptions, and designing processes and workflows that innovate more secure ways to remove superfluous human intervention.

At its root, automation is a straightforward solution, but it's a very open-ended one. And change is fast. Your goals will change rapidly because automation is, by nature, a flexible and adaptable solution to business needs. The end result is an automation-first mindset, but the workaround automations, the iterations and minimum viable product (MVP), and work optimizing should and will adjust to your capabilities and evolving business needs.



Allen Eastwood

Allen is a Senior Architect for Red Hat Consulting, where he brings more than 20 years of experience. He enjoys playing ice hockey when he's not helping customers automate their IT systems.



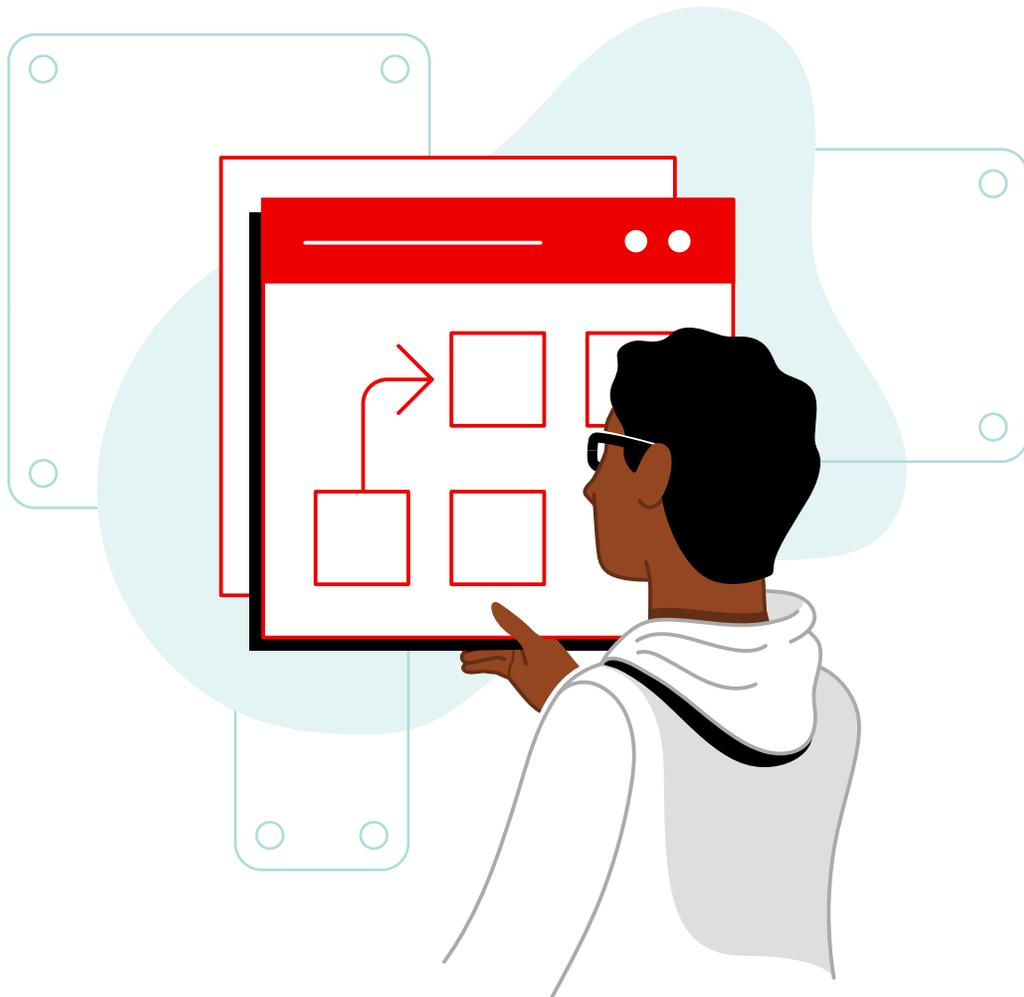
Larry Spangler

Larry is a Principal Solution Manager on the Red Hat Services Portfolio Management team, where he leads the definition, development, and management of automation services products for Red Hat's Global Services organization. With more than 30 years in IT software development, systems administration, and professional services, he brings a breadth of experience and insight to the practice.

Conclusion

How to get started with IT automation

- 45 Embracing IT automation: Why it's good for your career—and how you can get started
- 48 Get started



Embracing IT automation: Why it's good for your career— and how you can get started

Red Hat can make the path to automation more efficient with comprehensive training options that translate into immediate business impacts.

By James Mighion, Manager, Red Hat Global Learning Services automation practice, and Steven Bonneville, Principal Technical Architect, Red Hat Training

In 2019, [Comcast](#) discovered a need to create, share, and reuse automation capabilities across projects and teams—with governance and control. They also required in-house expertise to manage these efforts, so Comcast engaged Red Hat for a five-day onsite training engagement. Comcast Voice Engineers took Red Hat Training courses on Red Hat Ansible Automation Platform, providing IT staff with the opportunity to implement, manage, and troubleshoot automation with hands-on, expert-guided curriculum.

However, the benefits of Red Hat Training and Certification also apply to individuals. Many IT professionals seek automation training on their own and find that it enhances their value to their company and their personal marketability. Take, for example, Christian Sandrini, an IT professional who was named the [2021 Red Hat Certified Professional of the Year](#). He has earned nearly a dozen Red Hat certifications, including the Red Hat Certified Specialist in Ansible Best Practices and the Red Hat Certified Specialist in Ansible Automation Platform.

Sandrini participates heavily in Red Hat Training as a Red Hat Learning Subscription subscriber. He said, “The Red Hat Learning Subscription has helped me because it suits my learning style. I enjoy having the self-paced learning platform where I can go back to rewatch certain videos or do the labs, which are very hands-on. It helped me get into the technology very fast, so even if I didn’t know about a certain product, I could sign up for a class and learn very quickly what the product is all about. Another thing I really enjoy is the early access section of Red Hat Learning Subscription, because it gives you a sneak peek into the technologies that are coming next.”

Sandrini’s first step toward implementing his knowledge was introducing his organization to Red Hat Ansible Automation Platform. His goal was to automate functions and modernize how the team provisions and configures servers with an infrastructure-as-code approach. After receiving overwhelmingly positive reviews of his work from the team, he deployed Ansible Automation Platform. Sandrini said “that was really where the change happened.



It allowed us to have role-based access control, and this is when other teams started to become interested. It shows that Ansible cannot only be used for Linux, but for other components around it as well.”

Sandrini has become an invaluable asset to his own team, implementing and maintaining IT automation to streamline processes and cut costs using the knowledge obtained through Red Hat Training and Certification. He advocates making training available across teams to acquire and improve skills, as well as to distribute knowledge across team members.

For more than 20 years, Red Hat has provided training and certification options for technology professionals to keep up with current and emerging trends.

As the demand for professionals skilled in IT automation continues to grow, so does the importance of staying abreast of new strategies and technologies. In the ever-evolving field of IT, training and professional development is key to remaining competitive on an individual and organizational level. Red Hat Training and Certification is constantly updating course content to keep pace with the industry, ensuring that trained and certified professionals who want to enhance their careers in IT automation have the tools they need for success.

For more than 20 years, Red Hat has provided training and certification options for technology professionals to keep up with current and emerging trends. Red Hat Training courses have options to suit various needs and learning styles, from traditional, classroom-based training to onsite training to diverse virtual learning options. With an average of 20 years in IT and 11 years using Red Hat products, Red Hat certified instructors strive to create an immediate impact on business goals with their expertise.

Red Hat Learning Subscription allows learners to complete real-time virtual training from their own device and instantly access the complete catalog of self-paced courses, videos, and labs. All of this content is consistently updated to keep pace with the industry. In fact, 41% of courses available in the Red Hat Learning Subscription catalog have been added or updated within the past year. Some Red Hat Learning Subscription tiers also include the cost of sitting for a Red Hat certification exam, simplifying the journey from obtaining knowledge to proving it.

While Red Hat Training helps students keep pace with the latest in automation and digital transformation, Red Hat Certification validates IT professionals as skilled and prepared to take on the most ambitious projects in the face of evolving industry challenges. Red Hat currently offers two certifications with a focus on IT automation. Red Hat Enterprise Linux Automation with Ansible (RH294) and the Red Hat Certified Engineer (RHCE®) exam lay the groundwork for automating workflows, employing DevOps practices, and using Ansible Automation Platform for more efficient development. To take this journey further, Advanced Automation: Red Hat Ansible Best Practices (DO447) and the Red Hat Certified Specialist



in Advanced Automation: Ansible Best Practices exam develop and prove the skills needed to use and extend existing Ansible infrastructure across business units in large enterprise environments. Red Hat certified professionals help their organizations achieve optimal efficiency and cost savings through expertise in automation. In addition, organizations are drawn to recruit and retain certified professionals due to decreased time to onboard and increased cost savings.



Steven Bonneville

Steven Bonneville is a Principal Technical Architect at Red Hat in its training organization. With more than 20 years working at Red Hat, Steven provides deep expertise with Red Hat technologies to design courses for Red Hat’s training curriculum, mentor and direct curriculum developers, and recommend directions for future training development. For many years, he has been responsible for the Red Hat Enterprise Linux system administration curriculum, including the RHCE training track, and advanced courses on system administration, virtualization, and storage. He wrote the first version of many of those courses. His current focus is in developing cross-platform automation training using Red Hat Ansible Automation Platform.



James Mighion

James Mighion is the manager of the automation practice within Red Hat’s Global Learning Services organization. He’s been with Red Hat since 2011 and has had many different roles along the way. James is passionate about automation and contributing to open source projects.



Get started



Start your trial of Red Hat Ansible Automation Platform

Get a single, self-supported 60-day subscription for Red Hat Ansible Automation Platform for Red Hat Enterprise Linux.

→ [Try Ansible Automation Platform for 60 days](#)



Learn the basics of Red Hat Ansible Automation Platform

Try our series of on-demand online videos. Ansible Essentials: Simplicity in Automation Technical Overview (DO007) introduces you to Ansible Automation Platform, including configuration management, provisioning, deploying, and managing compute infrastructure across cloud, virtual, and physical environments.

→ [Learn about Ansible Automation Platform with free training](#)



Enable Sysadmin—A community for system administrators

Enable Sysadmin is a community publication that publishes guides, how-tos, tutorials, explainers, tips and tricks, and more. The publication covers existing and emerging technology concepts, software and applications, scripting and programming languages frequently used by system administrators, and common challenges and solutions.

→ [Sign up for the newsletter](#) | [Join the community](#)



Lead with automation and connect your teams

Digital leaders aren't just automating their existing workflows—they're learning to create shared value with automation. Your organization may be automating some aspects of IT, but are you in an advanced or beginning stage? Take the online assessment to find out your current automation maturity, identify next steps, and get resources to support your progress.

→ [Take the assessment](#)