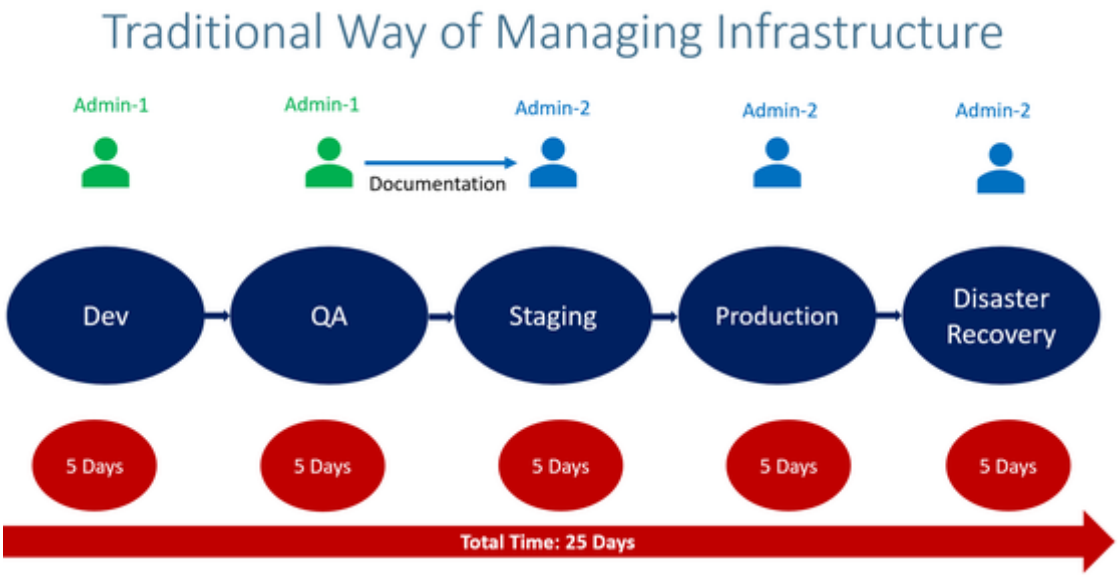
01- What is Infrastructure as Code (IaC)?

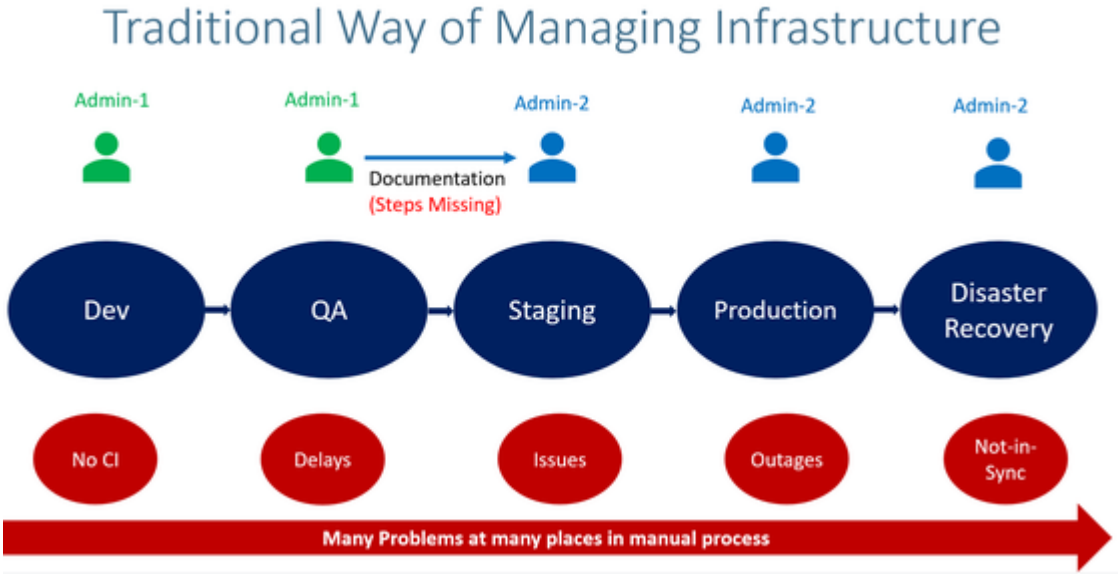
Overview

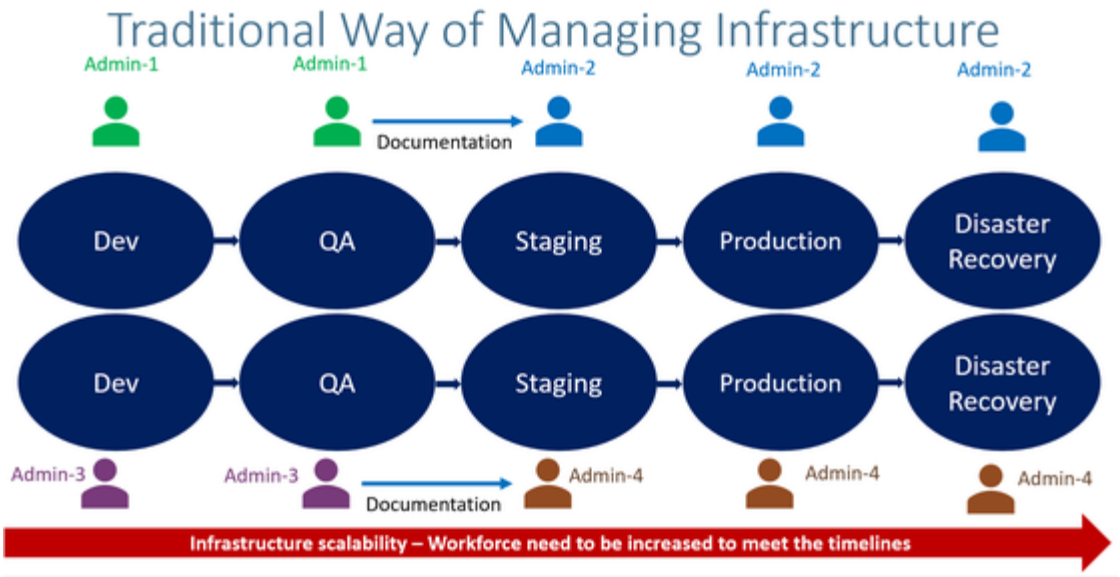
In the past, managing IT infrastructure was a hard job. System administrators had to manually manage and configure all of the hardware and software that was needed for the applications to run.

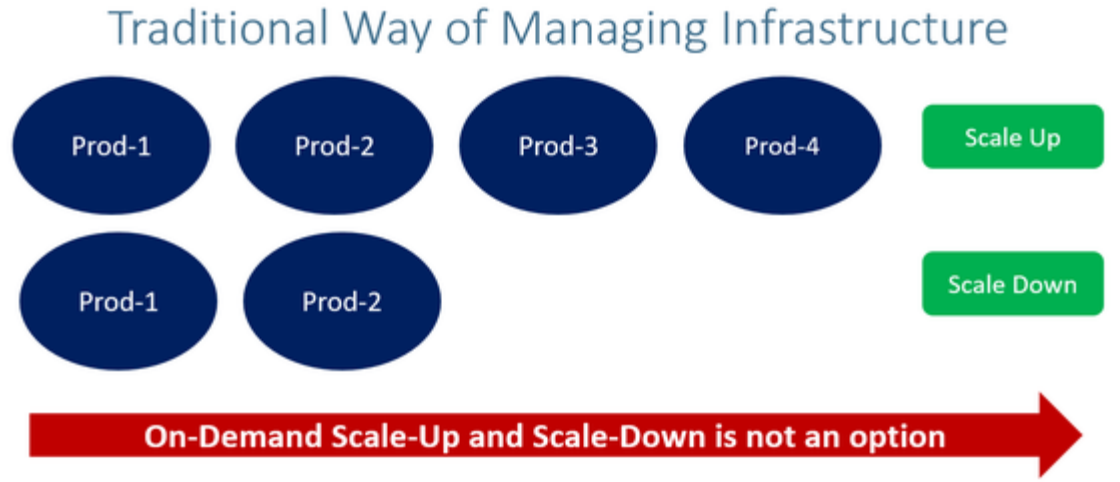
However, in recent years, things have changed dramatically. Trends like cloud computing revolutionized—and improved—the way organizations design, develop, and maintain their IT infrastructure.

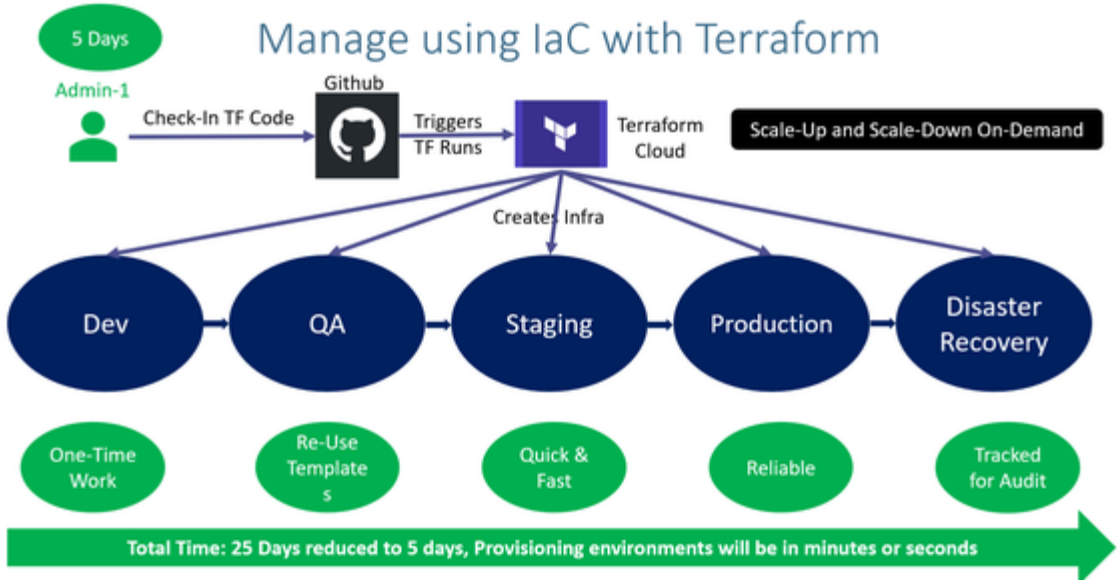
One of the critical components of this trend is called **“infrastructure as code,”**

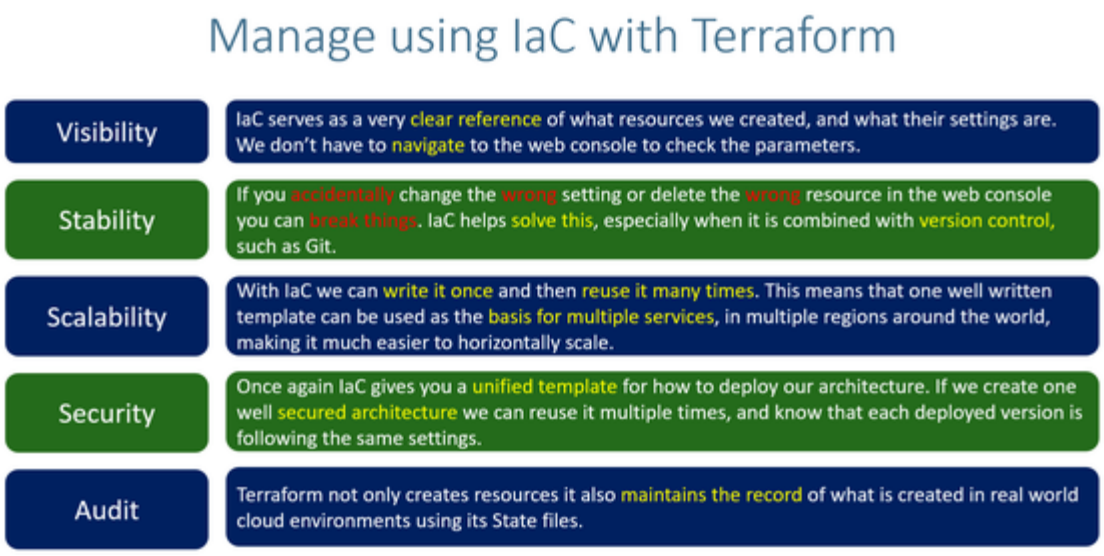





Traditional Way of Managing Infrastructure
Admin-1
Admin-1
Admin-2
Admin-2
Admin-2
Documentation
Dev
QA
Staging
Production
Disaster Recovery
Dev
QA
Staging
Production
Disaster Recovery
Admin-3
Admin-3
Admin-4
Admin-4
Admin-4
Documentation
Infrastructure scalability – Workforce need to be increased to meet the timelines


Traditional Way of Managing Infrastructure
Prod-1
Prod-2
Prod-3
Prod-4
Scale Up
Prod-1
Prod-2
Scale Down
On-Demand Scale-Up and Scale-Down is not an option


5 Days
Manage using IaC with Terraform
Admin-1
Check-In TF Code
Github
Triggers TF Runs
Terraform Cloud
Scale-Up and Scale-Down On-Demand
Creates Infra
Dev
QA
Staging
Production
Disaster Recovery
One-Time Work
Re-Use Templates
Quick & Fast
Reliable
Tracked for Audit
Total Time: 25 Days reduced to 5 days, Provisioning environments will be in minutes or seconds



Benefits of infrastructure as code (IaC)

- Create separate but identical (or near-identical) environments (staging, production, test, etc.)
- **Perform disaster recovery** by quickly creating a new, identical environment from the IaC scripts
- **Versioning**: We can use any VCS to store and manage the infrastructure code
- **Improved productivity.** Administrators and operators no longer must perform manual configuration steps for data center infrastructure changes
- **Speed:** IaC transforms developers' time-consuming provisioning work into a simple script execution to have their infrastructure ready. As a result, application deployments no longer have to wait for the infrastructure, and new software can be released much more quickly.
- **Efficiency:** Codifying your infrastructure gives you a template for provisioning, simplifying system configuration, maintenance, and management. It creates an elastic infrastructure that is repeatable and scalable. This means that DevOps can accelerate every step of software development, leading to more applications being released on a daily basis.
- **Reduced cost:** IaC allows virtual machines to be managed programmatically, eliminating the need for manual hardware configuration and updates. One operator can deploy and manage one machine or 1000 machines using the same set of codes. That means fewer staff are needed and purchasing new hardware is no longer necessary, reducing costs significantly.
- **Consistency:** IaC can improve consistency and reduce errors that often happen during manual configuration. It also eliminates any configuration drift that might occur during a manual process. By codifying and documenting your configuration specifications, IaC helps you avoid undocumented, ad-hoc configuration changes.

Infrastructure as Code (IaC) is the managing and provisioning of infrastructure through code instead of through manual processes.

With IaC, configuration files are created that contain your infrastructure specifications, which makes it easier to edit and distribute configurations. It also ensures that you provide the same environment every time.

Version control is an important part of IaC, and your configuration files should be under source control just like any other software source code file. Deploying your infrastructure as code also means that you can divide your infrastructure into modular components that can then be combined in different ways through automation.

Infrastructure as Code tools you can use to automate your deployments

- Terraform
- AWS CloudFormation
- Ansible
- Chef
- Puppet
- Dockerfile
- Jenkinsfile