Infrastructure as Code (IaC) w/ Amazon Web Services (AWS)

By
Reetesh Dooleea & **Ming Xiao Yuan**

7th November 2023

01

Introduction

Software Release

- Final or new version of the software released to the end users.
- Deploying software into environments (staging, production).
- Includes new features, bug fixes, improvements and so on.
- Marks an important milestone in the development cycle.
- But how to adapt this process?

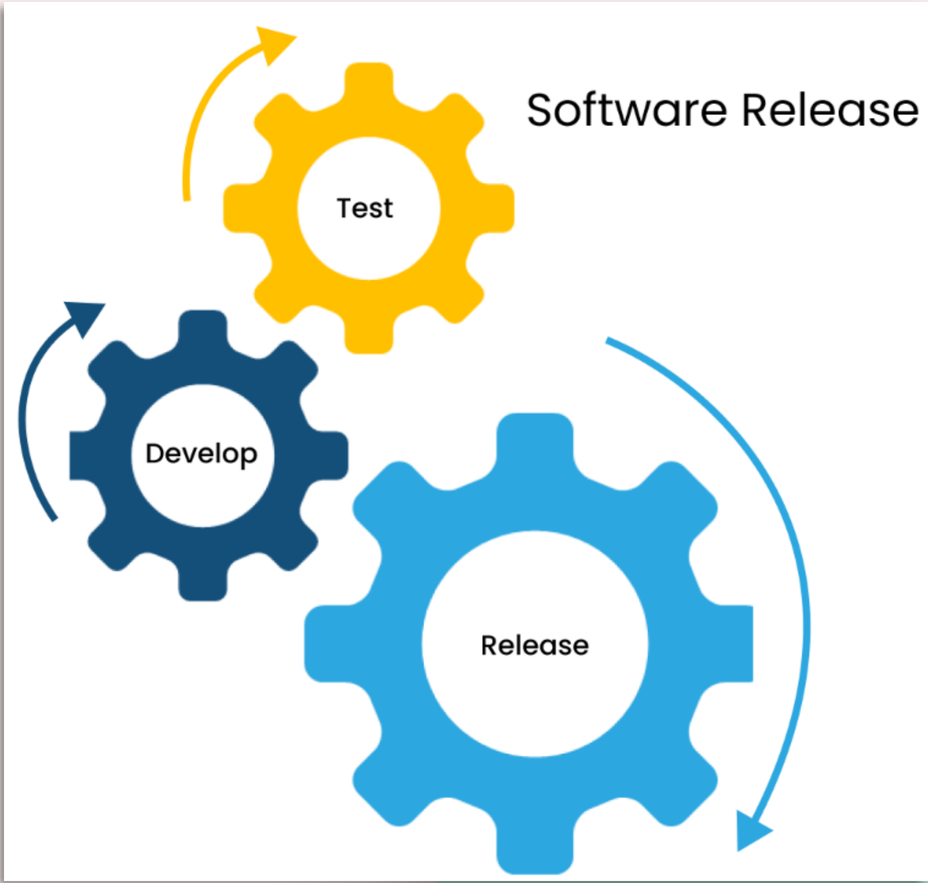


Figure 1: *Software Release*. Retrieved from Quentelli [1].

CI/CD Approach

- Continuous Integration: Integration of code changes in a shared repository.

- Continuous Delivery: Automation of deployment process.

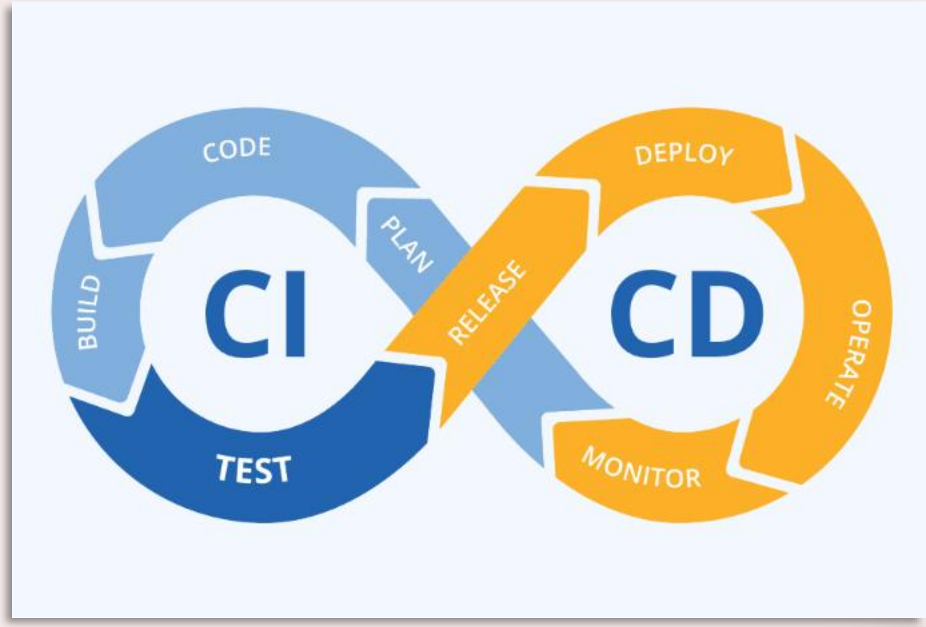


Figure 2: *CI/CD steps*.
Retrieved from TechAhead [2].

DevOps

- Enables Software Development (Dev) and IT Operations (Ops) teams to accelerate delivery through collaboration, communication and automation.

- Goal: Perform Software delivery more efficiently and quickly.

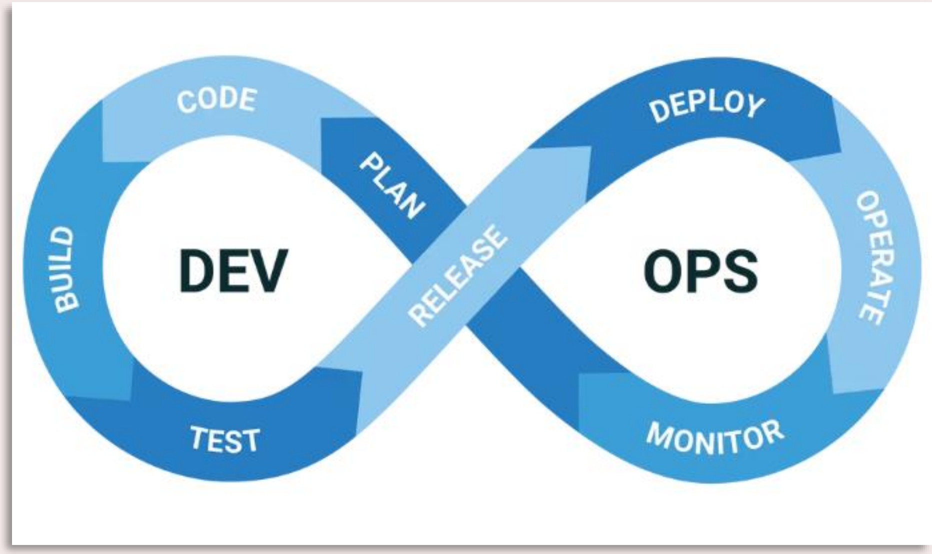


Figure 3: *DevOps philosophy*. Retrieved from Octopus [3].

Relationship between Software Release, CI/CD & DevOps

Tooling

DevOps: framework for fostering collaboration and automation.

CI/CD: technical tools to implement DevOps principles.

Guidance

DevOps encourages the adoption of CI/CD practices to release software efficiently and quickly.

DevOps Practices

Continuous Integration

Continuous Delivery

Microservices

Infrastructure as Code

Monitoring & Logging

Communication & Collaboration

02

Infrastructure as Code (IaC)

Infrastructure as Code

What does Infrastructure mean?

Resources and components required to:

- Support hardware
- Execute software applications or services

Examples of resources & components

- Physical Hardware
- Operating Systems
- Networking
- Servers
- Storage
- Security settings
- Monitoring and Logging
- Load Balancing
- Recovery

Infrastructure as Code

On-Premise

Resources and systems located and ran on a company's own data center.

Cloud Providers

Third-party companies that offer infrastructures hosted in their own data centers which are accessible through an interface.

Microsoft Azure,
Amazon Web Services,
Google Cloud Platform

The difference is mainly the **location**.

Features of IaC

- Practice of managing, configuring and provisioning infrastructure using code (files) and automation.
- Enables the automated creation, modification and destruction of infrastructure resources.

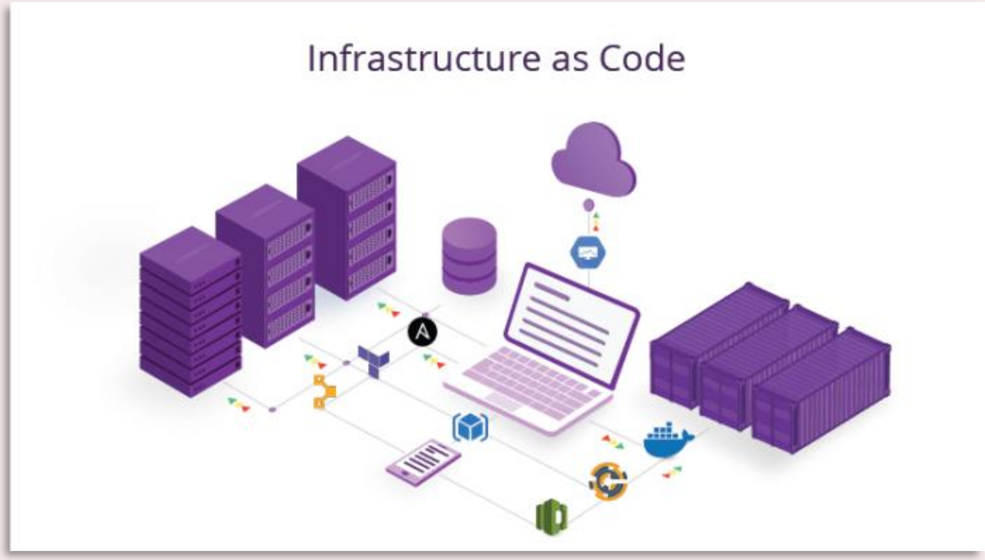


Figure 4: *Infrastructure as Code*. Retrieved from Urolime [4].

Benefits of IaC practices

Consistency

- All infrastructure components are configured identically.
- Reduced risk of errors and inconsistencies.

Scalability

- Easy scaling of infrastructure resources to meet changing demands.
- No manual intervention.

Repeatability

- Deployment process can be replicated and repeated.
- Reduced risk of deployment-related issues and errors.

Version Control

- Easier to track changes to infrastructure code.

Cons of IaC

Complexity

Huge IaC configurations can become complex.

Learning Curve

Time-consuming to learn new skills and adapt to different way of managing infrastructures.

Version Control

Handling changes and conflicts made over IaC files can be challenging and dangerous.

Cons of IaC

Complexity

Huge IaC configurations can become complex.

Learning Curve

Time-consuming to learn new skills and adapt to different way of managing infrastructures.

Version Control

Handling changes and conflicts made over IaC files can be challenging and dangerous.

But..

These challenges can be addressed with proper planning, training and right tooling!
Benefits of IaC >>> Cons of IaC

Relationship between IaC & DevOps

Automation is key.

Usage of software to continuously manage, automate and deliver resources.

03

IaC Tools

Choice of IaC Tools

Puppet
Puppet (2005)

Ansible
Red Hat (2012)

Terraform
HashiCorp (2014)

Chef
Chef (2009)

04

Terraform

What is Terraform?

- Provides a consistent workflow to create resources as well as provision and manage the lifecycle of an infrastructure.

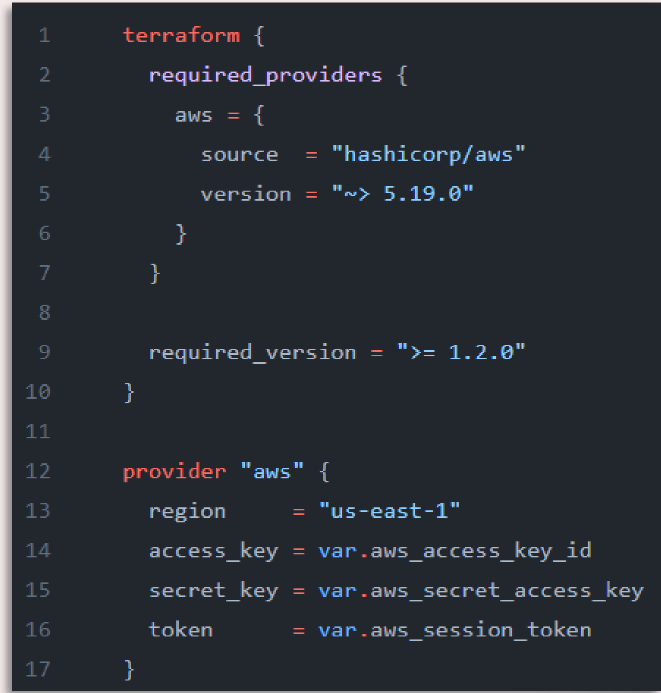
- Compatible with thousands of providers:

  Microsoft Azure,
  Amazon Web Service,
  Google Cloud Platform,
  GitHub, ..

Figure 5: *Terraform logo*. Retrieved from Terraform [6].

How to use Terraform?

- Files that define both cloud and on-premise resources in human-readable configuration files.
- Files can be reused, versioned and shared.
- Usage of HCL language.
  (HashiCorp Configuration Language)



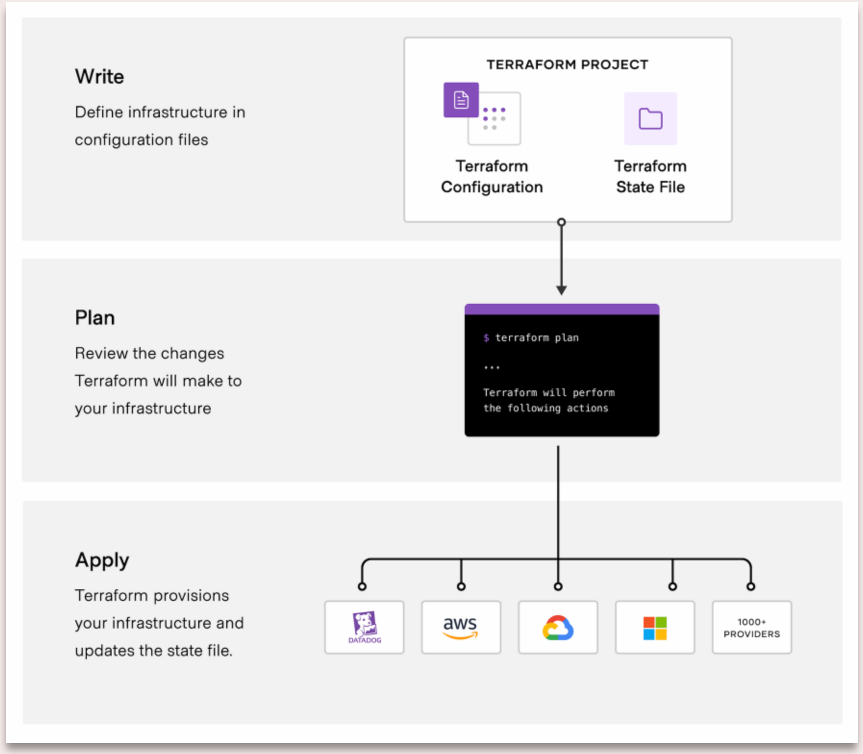
```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 5.19.0"
    }
  }

  required_version = ">= 1.2.0"
}

provider "aws" {
  region     = "us-east-1"
  access_key = var.aws_access_key_id
  secret_key = var.aws_secret_access_key
  token      = var.aws_session_token
}
```

Figure 6: *Example of Terraform HCL*. Retrieved from our GitHub code [7].

Terraform Workflow


Write
Define infrastructure in configuration files
TERRAFORM PROJECT
Terraform Configuration
Terraform State File
Plan
Review the changes Terraform will make to your infrastructure
Apply
Terraform provisions your infrastructure and updates the state file.
DATADOG
aws
1000+ PROVIDERS

```
$ terraform plan

...

Terraform will perform
the following actions
```

Figure 7: *Terraform Workflow*.
Retrieved from Terraform [6].

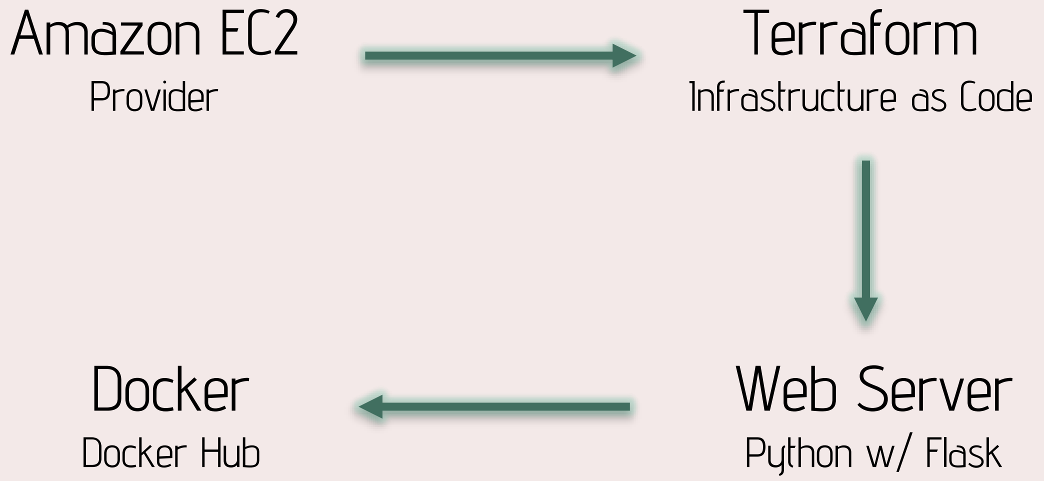
05

Motivation

06

Tool Demonstration

Methodology



07

Conclusion

Summary

DevOps

CI/CD approach facilitates the process of Software Release.

Infrastructure as Code

Managing and provisioning infrastructure with code and automation.

Terraform

IaC tool that defines, manages and provisions infrastructure resources using code.

Tool Demonstration

Creation of an infrastructure on AWS EC2 via Terraform to serve a Web Server with Docker.

References

[1] *Guide to Release Management.* Retrieved from Qentelli. https://www.qentelli.com/thought-leadership/insights/guide-release-management.

[2] *How CI/CD can save app development time and create robust apps.* Retrieved from TechAhead. https://www.techaheadcorp.com/blog/how-ci-cd-save-app-development-time.

[3] *What is DevOps?.* Retrieved from Octopus. https://octopus.com/devops.

[4] *Infrastructure as Code (IaC).* Retrieved from Urolime. https://www.urolime.com/blogs/infrastructure-as-code.

[5] *Infrastructure as Code*. Retrieved from Wikipedia. https://en.wikipedia.org/wiki/Infrastructure_as_code.

[6] *Terraform.* Retrieved from HashiCorp Developer. https://developer.hashicorp.com/terraform/intro.

[7] *Tool demo GitHub code*. Available on GitHub. https://github.com/keshavDooleea/LOG6307E__Tool_Demo.