Infrastructure as Code

ow

Pointers

- Code-based infrastructure
- Configuration file
- Reproducibility and Consistency
- Automation and Orchestration
- Version Control
- Scalability and Agility

What is IAC??

- Infrastructure as Code (IaC) is a practice that involves managing and provisioning infrastructure resources using machine-readable configuration files.
- Instead of manually configuring and managing infrastructure components, IaC allows you to define your infrastructure in code, providing several benefits such as version control, automation, and reproducibility.

Code-based Infrastructure

- IaC treats infrastructure components (servers, networks, storage, etc.) as code.
- Instead of using manual processes or GUI-based tools, infrastructure configurations are expressed in code files.Typically using declarative languages like Terraform, CloudFormation, or Ansible.
- IaC uses configuration files to describe the desired state of infrastructure resources. These files specify what resources should be provisioned, their properties, relationships, and any additional configuration details.

Automation/Version Control

- Reproducibility and Consistency: With IaC, infrastructure provisioning becomes repeatable and consistent. The code-based approach allows you to create infrastructure in a predictable manner across different environments, ensuring that the same configurations and resources are deployed consistently each time.
- Automation and Orchestration: IaC tools enable automation and orchestration of infrastructure provisioning. You can define dependencies, order of operations, and complex configurations in code, allowing for automated deployment and management of infrastructure resources. This helps reduce manual errors and saves time.
- Version Control: Infrastructure configurations can be stored in version control systems like Git. This provides the ability to track changes, review code, collaborate with team members, and revert to previous versions if needed

Advantage of IAC

- Consistency and Standardization: IaC enables the creation of infrastructure configurations in a standardized and consistent manner. By defining infrastructure as code, you can ensure that the same configurations are applied consistently across different environments, reducing configuration drift and promoting reliability.
- Reproducibility and Version Control: IaC allows you to version control infrastructure configurations using tools like Git. This brings the benefits of software development practices to infrastructure management, enabling you to track changes, roll back to previous versions, and collaborate effectively with team members.
- Automation and Efficiency: With IaC, infrastructure provisioning and management become automated. By defining your infrastructure in code, you can leverage automation tools to deploy and manage resources. This reduces manual effort, saves time, and minimizes the risk of human errors that can occur during manual configuration.
- Scalability and Agility: IaC facilitates easy scaling of infrastructure. By modifying the code, you can quickly adjust resource quantities, sizes, or add new resources as needed.