**What is IaC?**  
Infrastructure as code (IaC) means to manage your IT infrastructure using configuration  
files.

**Why IaC?**  
Historically, managing IT infrastructure was a manual process. People would physically  
put servers in place and configure them. Only after the machines were configured to the  
correct settings required by the OS and dependencies would those people deploy the  
application.

Businesses are making a transition where traditionally managed infrastructure can no  
longer meet the demands of today’s businesses. IT organizations are quickly adopting  
the public cloud, which is predominantly API-driven.

To meet customer demands and save costs, application teams are architecting their  
applications to support a much higher level of elasticity, supporting technology like  
containers and public cloud resources. These resources may only live for a matter of  
hours; therefore, the traditional method of raising a ticket to request resources is no  
longer a viable option.

**Benefits of IaC**

**Consistency and Standardization:** IaC allows infrastructure to be defined and managed using code, ensuring consistent configurations across all environments, including development, testing, staging, and production. This consistency reduces the risk of configuration drift and minimizes errors caused by manual intervention.

**Scalability**: With IaC, infrastructure can be easily scaled up or down to meet changing demands. By defining infrastructure as code, organizations can automate the process of provisioning and scaling resources, enabling rapid and efficient deployment of new infrastructure components.

**Speed and Agility:** IaC enables rapid deployment of infrastructure resources through automation, reducing the time and effort required for manual provisioning. This agility allows organizations to respond quickly to changing business requirements, deliver new features faster, and accelerate time-to-market for applications and services.

**Reproducibility and Version Control:** Infrastructure code can be version-controlled using tools like Git, enabling teams to track changes, collaborate effectively, and roll back to previous versions if necessary. This reproducibility ensures that infrastructure deployments are predictable and reliable, even across distributed teams and environments.

**Cost Efficiency:** By automating infrastructure provisioning and management, IaC helps optimize resource utilization and reduce operational costs. Organizations can provision resources on-demand, scale them as needed, and decommission unused resources, leading to cost savings and improved ROI.

**Enhanced Security and Compliance:** IaC enables security best practices to be codified and enforced throughout the infrastructure lifecycle. By incorporating security controls into infrastructure code, organizations can ensure that security policies are consistently applied, vulnerabilities are identified and remediated proactively, and compliance requirements are met more effectively.

**Collaboration and Knowledge Sharing:** Infrastructure code serves as a single source of truth for defining and documenting the infrastructure architecture. This fosters collaboration among development, operations, and security teams, promotes knowledge sharing, and facilitates cross-functional alignment towards common goals.