**2. Overview of IaC Tools**

1. Create a document summarizing the main IaC tools, focusing on

**1. Terraform**

**Features:**

* Open-source tool developed by HashiCorp.
* Uses a declarative language (HashiCorp Configuration Language - HCL).
* Supports multiple cloud providers (AWS, Azure, GCP, etc.).
* State management for tracking infrastructure changes.
* Module-based architecture for reusable configurations.

**Use Cases:**

* Multi-cloud deployments.
* Infrastructure provisioning and lifecycle management.
* Managing cloud services, networking, and application resources.

**Advantages:**

* Cloud-agnostic, making it versatile for hybrid and multi-cloud environments.
* Provides a structured and modular approach to infrastructure management.
* Supports plan and apply workflow, allowing a preview of changes before deployment.

**2. AWS CloudFormation**

**Features:**

* Native Infrastructure as Code tool for AWS.
* Uses JSON or YAML templates to define AWS resources.
* Stack-based management for grouping resources.
* Integrates with AWS services for automation and governance.

**When to Use:**

* Ideal for AWS-centric environments.
* When managing large-scale AWS infrastructure in a repeatable manner.
* Ensuring compliance and governance with AWS-native security features.

**3. Ansible**

**Role in Configuration Management & IaC:**

* Agentless automation tool developed by Red Hat.
* Uses YAML-based playbooks to define configurations and tasks.
* Supports provisioning, configuration management, and application deployment.
* Can manage both cloud infrastructure and on-premises servers.

**How It Relates to IaC:**

* While primarily a configuration management tool, Ansible can provision infrastructure through modules for AWS, Azure, and other providers.
* Works well alongside other IaC tools like Terraform for post-provisioning configuration.

**4. Azure Resource Manager (ARM)**

**Features:**

* Native IaC tool for Microsoft Azure.
* Uses JSON or Bicep templates to define and deploy Azure resources.
* Provides role-based access control (RBAC) and policy enforcement.
* Enables dependency management between resources.

**Usage in Azure Environments:**

* Ideal for managing and automating Azure infrastructure.
* Ensuring consistent deployment of Azure services.
* Leveraging built-in Azure security and governance policies.