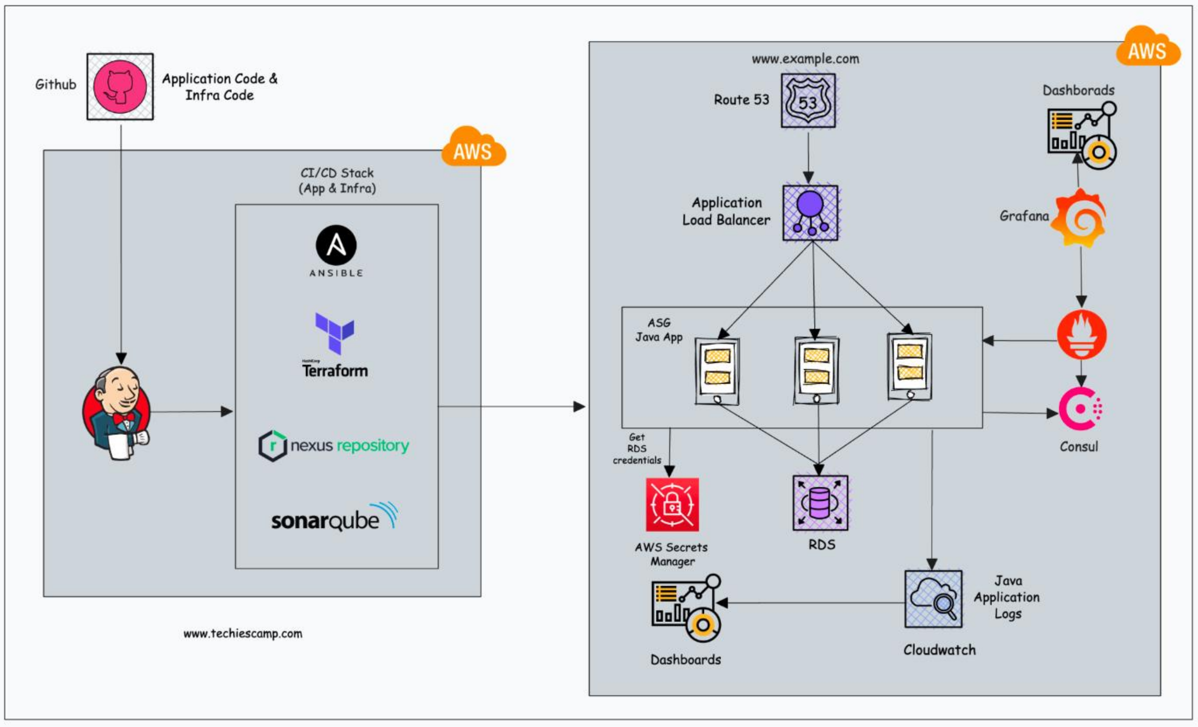
Automating AWS VPC Creation and Management for Scalable Application Deployment



Context: In modern organizations, especially those adopting cloud infrastructure, efficient network design is crucial for application performance, security, and scalability. AWS Virtual Private Cloud (VPC) allows organizations to build isolated networks in the cloud, where they can deploy applications securely. However, as organizations scale, manually creating and managing VPCs becomes error-prone, time-consuming, and difficult to replicate across environments.

Challenge: For an application architecture that includes a Java-based web application, CI/CD automation tools, platform monitoring tools (e.g., Prometheus, Grafana), and managed services (e.g., AWS RDS, S3, Secrets Manager), designing an AWS VPC that adheres to best practices and ensuring its reproducibility is a major challenge. The application requires:

- A VPC with 15 subnets distributed across three availability zones (public, app, database, management, and platform subnets).
- Proper routing via an Internet Gateway for public subnets and a NAT Gateway for private subnets.
- Secure access to AWS managed services using VPC Endpoints.
- Dedicated Network ACLs for segregated security policies across subnet groups.

Objective: The goal is to design and automate the creation of a highly available, scalable, and secure AWS VPC infrastructure using Terraform. This automation will ensure that the network architecture can be replicated consistently across development, testing, and production environments, while minimizing manual effort and reducing the risk of misconfigurations.

Requirements:

1. **VPC Design**:
    - CIDR Block: 10.0.0.0/16.
    - 15 subnets distributed across us-west-2a, us-west-2b, us-west-2c.
    - Internet Gateway for public subnets and NAT Gateway for private subnets.
    - VPC Endpoints for services such as S3, CloudWatch, and Secrets Manager.
    - Dedicated Network ACLs for public, app, database, management, and platform subnets.
2. **Automation with Terraform**:

   - Automate the creation of the VPC and all associated components (subnets, gateways, routing tables, endpoints, etc.).
   - Ensure modularity in Terraform code for reusability across multiple environments.
3. **Outcome**:
   - A Terraform solution that can be used by DevOps engineers or developers to create and manage AWS VPCs for scalable application deployments with a single command.

Key Deliverables:

1. VPC design documentation.
2. Terraform scripts to automate VPC creation.
3. Documentation on how to use the Terraform scripts for infrastructure provisioning.
4. A test plan to validate that the network architecture meets application requirements (e.g., reachability, security).

Significance: Automating VPC creation not only speeds up the process of infrastructure provisioning but also ensures consistency, reduces human errors, and allows for scalability in cloud-based application deployment.