

The diagram illustrates the workflow for provisioning an AWS environment using Terraform and Ansible. It shows the following components and their interactions:

- Control Node:** The central orchestrator, represented by a laptop icon.
- Main.tf:** A file icon representing the Terraform configuration file.
- Terraform:** The tool used for provisioning infrastructure, represented by a blue cube icon.
- Cloud Provider (aws):** The AWS cloud provider, represented by the AWS logo.
- Ansible Playbooks:** Represented by a document icon with a blue square, used for configuration management.
- ANSIBLE:** The configuration management tool, represented by a black circle with a white 'A'.
- Customer/Clients:** Represented by a yellow box containing icons for a laptop, a monitor, and a server.
- Infrastructure Components (AWS Environment):**
 - EC2 Instance:** Labeled "EC2 Instance Running To-Do Flask Application", represented by an orange square icon.
 - Target Node:** A component within the EC2 instance, represented by a blue square icon.
 - Public Subnet:** A network component, represented by a light blue box.
 - US-EAST-1A:** An Availability Zone, represented by a light blue box.
 - VPC:** A Virtual Private Cloud, represented by a light blue box.
 - IGW (Internet Gateway):** Represented by a green diamond icon.
 - RDS-Postgres:** A database instance, represented by a blue cylinder icon.
 - US-EAST-1:** A Region, represented by a light blue box.

The workflow is as follows:

- The **Control Node** interacts with **Main.tf** and **Ansible Playbooks**.
- Main.tf** is used by **Terraform** to provision the **Cloud Provider (aws)**.
- Ansible Playbooks** are used by **ANSIBLE** to configure the **EC2 Instance** and **RDS-Postgres**.
- The **Cloud Provider (aws)** provisions the **EC2 Instance**, **Target Node**, **Public Subnet**, **US-EAST-1A**, **VPC**, and **IGW**.
- The **EC2 Instance** is connected to the **Public Subnet** and the **IGW**.
- The **EC2 Instance** is connected to the **RDS-Postgres** database.
- The **EC2 Instance** is connected to the **Customer/Clients**.

The diagram illustrates the Level 3 architecture, showing the interaction between various components:

- Level 3** (Overall Architecture):
 - EC2 Flask App Container**: Contains the **Web Server** and **Flask Application**.
 - SSH Configuration**: Manages **Packet Management (Python, Docker etc.)** and **Flask Setup**.
 - Database Configuration to RDS**: Manages **Database Connection to RDS** and **MySQL Database Client**.
 - EC2 Configuration**: Manages **RDS Configuration**, **Security Group**, and **VPC Configuration**.
- Tools and Services**:
 - Ansible**: Used for **SSH Configuration** and **Flask Setup**.
 - Terraform**: Used for **EC2 Configuration** and **VPC Configuration**.
 - RDS-Postgres**: A database instance that interacts with the **MySQL Database Client** and the **Flask Application**.