STEP BY STEP EXECUTION OF TERRAFORM TEMPLATE FOR AWS_EKS_ARCHITECTURE:

1)Provider Configuration:

Defines the AWS provider and specifies the region where the resources will be created.

2)Terraform Backend Configuration:

Configures the Terraform backend to store the state file in an S3 bucket. The state file keeps track of the deployed resources and their configuration.

3)VPC and Subnets:

Creates a VPC (Virtual Private Cloud) with a specified CIDR block.
Creates public and private subnets within the VPC, each with its own CIDR block and availability zone.

4)Security Group for Bastion Host:

Creates a security group to control inbound and outbound traffic for the bastion host (typically used for secure access to resources).

5)Bastion Host EC2 Instance:

Launches an EC2 instance (bastion host) within one of the public subnets, allowing secure access to other resources within the private subnets.

6)NAT Gateways and Elastic IPs:

Creates NAT gateways and associates them with the public subnets to allow outbound internet access for resources within the private subnets.
Allocates Elastic IPs for the NAT gateways.

7)Autoscaling Group for Bastion Host:

Sets up an autoscaling group to manage the bastion host EC2 instances, ensuring a desired capacity of 1 and launching instances across multiple availability zones.

8)Elastic Load Balancer:

Creates an Elastic Load Balancer (ELB) with a listener to distribute traffic to backend instances.
Associates the ELB with the private subnets and a security group.

9)EKS Cluster:

Creates an Amazon EKS (Elastic Kubernetes Service) cluster with a specified name and VPC configuration.

10)EKS Worker Nodes:

Launches EC2 instances as worker nodes for the EKS cluster.Configures a launch configuration and autoscaling group for the worker nodes.

11)Kubernetes Control Plane Components:

Defines a Kubernetes deployment for the control plane components.

12)Amazon RDS Instance:

Creates an Amazon RDS (Relational Database Service) instance with specified configurations, such as the database engine, instance class, and security groups.

13)Security Groups:

Creates security groups for worker nodes and the RDS instance to control inbound and outbound traffic.

**Please note that the code provided is a template and may require customization to match our specific requirements, such as replacing placeholder values with actual resource IDs, AMI IDs, key pairs, etc. Additionally, it's important to review and understand the implications of the defined resources before deploying them in your environment.