**Creating AWS Elastic Kubernetes Service.**

1. Create Terraform Providers

* Terraform Settings Block
* Terraform AWS Provider
* Terraform Kubernetes Provider
* Terraform Kubectl Provider
* Terraform Random Provider
* Terraform Null Provider
* Terraform Helm Provider
* Terraform HTTP Provider

1. Create Terraform Backend for Terraform State File

* Create S3 Bucket
* Create DynamoDB Table with LockID
* Create Terraform Backend Block

1. Create VPC

* Create Public and Private Subnets
* Create Route Tables, Routes, and Route table Associations
* Create NAT Gateway with Elastic IP. EKS Worker Nodes in private subnet will connect to EKS Cluster API Server Endpoint via NAT Gateway without leaving AWS platform.
* Create Internet Gateway and attaching the IGW to the VPC

1. Create Bastion Host in Public Subnet

* Create Dynamic Terraform AWS Data latest AMI resource
* Create Bastion Host EC2 Instance
* Create Bastion Host Elastic IP
* Create Bastion Host Security Group
* Create SSH-key and terraform exec-provisioner and file resources to copy ssh-keys to remote /tmp/ folder
* SSH to remote node-group and verify kubelet-config.json in /etc/Kubernetes/kubelet/kubelet-config.json and kubeconfig in /var/lib/kubelet/kubeconfig. Get EKS Cluster API server endpoint url and verify the connection using nslookup or wget <api endpoint url>. Also check connection to ECR using following command **ps -ef | grep kube**, and look for --pod-infra-container-image=…etc.

1. Create Dockerfile and ECR

* Create Amazon Container Registry (ECR)
* Create appropriate IAM Roles and Policies for EKS Worker Nodes to be able to pull the Docker Image
* Create Dockerfile and push the Docker Image to ECR

1. Create AWS EKS Cluster

* EKS Cluster – EKS Cluster will be created in AWS managed account and EKS VPC. It is called EKS Control Plane
* EKS Cluster IAM Roles and Policies
* EKS Cluster Network Interfaces – are created in public subnets automatically when EKS Cluster is provisioned with Cluster endpoint access private ENABLED
* If Cluster endpoint access private only ENABLED, EKS Cluster Control Plane will create Network Interfaces in our AWS account in public or in private subnet which will allow node-groups to connect to EKS Cluster API server through Network Interfaces
* EKS Cluster Security Groups – are created in public subnets automatically when EKS Cluster is provisioned and associated to EKS Cluster Network Interfaces

1. Create AWS EKS Node Group

* EKS Node Group
* EKS Node Group IAM Roles and Policies
* EKS Node Group Security Groups – are created automatically when EKS Node Group is provisioned and associated to EKS Node Group Network Interfaces
* EKS Node Group Network Interfaces – are created automatically when EKS Node Group is provisioned with Cluster endpoint access public and private ENABLED, and it will communicate with EKS Cluster Control Plane API Server Endpoint

Graphical user interface

Description automatically generated

* If Cluster endpoint access private only ENABLED, EKS Cluster Control Plane will create Network Interfaces in our AWS account in public or in private subnet which will allow node-groups to connect to EKS Cluster API server through Network Interfaces

1. Install Kubectl CLI

* Install Kubectl CLI on local computer
* Configure Kubeconfig which will update **/Users/ali/.kube/config** file:

aws eks --region <region> update-kubeconfig --name <cluster\_name>

aws eks --region us-east-1 updatekubeconfig --name my-eks-cluster

1. Infrastructure as Code (IaC)
2. Terraform Install Tools, Command and Language Basics
3. Terraform Settings, Providers and Resources
4. Terraform Input Variables, Output Values, Datasources
5. Terraform Loops, MetaArguments, Splat Operator and Functions
6. AWS VPC 3-Tier Architecture Design using Terraform
7. Bastion Host AWS EC2 Instances, Security Groups, TF Provisioners with Terraform
8. AWS EKS Cluster, Public and Private Node Groups using Terraform
9. Kubernetes Fundamentals
10. Kubernetes Deployment and Service using YAML
11. Terraform Kubernetes Provider - Kubernetes Deployment & Service
12. Terraform Remote State Storage - AWS S3 & DynamoDB
13. AWS EKS IAM Roles for Service Accounts (IRSA) using Terraform
14. AWS EKS EBS CSI Driver Install with Self-Managed AddOn Option using Terraform
15. AWS EKS EBS Demo using k8s YAML (UserMgmt WebApp with MySQL DB)
16. AWS EKS EBS Demo using k8s Terraform (UserMgmt WebApp with MySQL DB)
17. AWS EKS EBS Volumes Retain and Resize Settings
18. AWS EBS CSI EKS Add-On
19. Provision AWS IAM Admin User as EKS Admin
20. Provision AWS IAM Basic User as EKS Admin
21. Provision of AWS Users (Admin & Basic) as EKS Admins using Terraform
22. Provision EKS Admins using IAM Roles & IAM Groups
23. Provision EKS Admins using IAM Roles & IAM Groups using Terraform
24. Provision EKS ReadOnly User using IAM Roles, Groups & k8s CR, CRB
25. Provision EKS Developer Users using IAM Roles, Groups & k8s R, RB
26. AWS Load Balancer Controller Install using Terraform Helm Provider
27. Ingress Basics - Automate with Terraform
28. Ingress Context Path based Routing - Automate with Terraform
29. Ingress SSL and SSL Redirect - Automate with Terraform
30. Install ExternalDNS using Terraform Helm Provider
31. Ingress with ExternalDNS - Automate with Terraform
32. Kubernetes LB Service with ExternalDNS - Automate with Terraform
33. Ingress Name based Virtual Host Routing- Automate with Terraform
34. Ingress SSL Discovery Host
35. Ingress SSL Discovery TLS
36. Ingress Groups - Automate with Terraform
37. Ingress Target Type IP - Automate with Terraform
38. Ingress Internal Load Balancer - Automate with Terraform
39. Ingress Cross Namespaces - Automate with Terraform
40. AWS Network Load Balancer with AWS Load Balancer Controller
41. AWS NLB TLS, External DNS with AWS LBC - Automate with Terraform
42. AWS NLB Internal LB with AWS LBC - Automate with Terraform
43. AWS EKS Fargate Profiles using Terraform
44. Run EKS Workloads on AWS Fargate - Automate with Terraform
45. AWS Fargate Only EKS Cluster using Terraform
46. AWS EFS CSI Controller Install using Terraform Helm Provider
47. AWS EFS Static Provisioning - Automate with Terraform
48. AWS EFS Dynamic Provisioning - Automate with Terraform
49. AWS EFS File System Mount for Fargate Workloads
50. Kubernetes Cluster Autoscaler Controller Install
51. Kubernetes Cluster Autoscaler Controller Test
52. Kubernetes Horizontal Pod Autoscaling with Terraform
53. Kubernetes Vertical Pod Autoscaling with Terraform
54. AWS EKS Monitoring and Logging with kubectl
55. AWS EKS Monitoring and Logging with Terraform

**Kubernetes Concepts Covered**

1. Kubernetes Deployments
2. Kubernetes Pods
3. Kubernetes Service of Type LoadBalancer
4. Kubernetes Service of Type ClusterIP
5. Kubernetes Ingress Service
6. Kubernetes Ingress Class
7. Kubernetes Storage Class
8. Kubernetes Storage Persistent Volume
9. Kubernetes Storage Persistent Volume Claim
10. Kubernetes RBAC
11. Kubernetes Role
12. Kubernetes Role Binding
13. Kubernetes Cluster Role
14. Kubernetes Cluster Role Binding
15. Kubernetes Cluster Autoscaler
16. Kubernetes Vertical Pod Autoscaler
17. Kubernetes Horizontal Pod Autoscaler
18. Kubernetes DaemonSets
19. Kubernetes Namespaces
20. Kubernetes Service Accounts
21. Kubernetes Groups
22. Kubernetes ConfigMaps
23. Kubernetes Requests and Limits
24. Kubernetes Worker Nodes

**Terraform Concepts covered**

1. Settings Block
2. Providers Block
3. Multiple Providers usage
4. Dependency Lock File Importance
5. Resources Syntax and Behavior
6. Resources Meta-Argument - depends\_on
7. Resources Meta-Argument - count
8. Resources Meta-Argument - for\_each
9. Resources Meta-Argument - lifecycle
10. Input Variables - Basics
11. Input Variables - Assign When Prompted
12. Input Variables - Assign with terraform.tfvars
13. Input Variables - Assign with auto tfvars
14. Input Variables - Lists
15. Input Variables - Maps
16. File Function
17. Output Values
18. Local Values
19. Datasources
20. Backends - Remote State Storage
21. File Provisioner
22. remote-exec Provisioner
23. local-exec Provisioner
24. Null Resource
25. Modules from Public Registry
26. element function
27. Remote State Datasource
28. Terraform Datasources

**Terraform Providers used**

1. AWS Terraform Provider
2. Kubernetes Terraform Provider
3. Kubectl Terraform Provider
4. HTTP Terraform Provider
5. Null Terraform Provider
6. Helm Terraform Provider

**What will students learn in your course?**

* You will build a AWS VPC 3-Tier network with Terraform
* You will build a AWS EKS Cluster with Public and Private Node Groups with Terraform
* You will implement a simple kubernetes deployment and service using Terraform Kubernetes Provider
* Understand and Implement Terraform Remote State Datasource
* You will learn AWS EKS IRSA (IAM Roles for Service Accounts Concept) in detail and implement it with Terraform
* You will master Kubernetes Storage concepts with AWS EBS CSI Controller by automating the whole install process with Terraform
* You will master AWS EKS IAM Concepts with 7 detailed Demo
* You will learn to implement AWS Load Balancer Controller Install on AWS EKS Cluster with Terraform
* You will learn to implement 14 Ingress Service Demos (AWS Application Load Balancer) using Terraform Kubernetes Provider
* You will learn to implement 3 Kubernetes Service Demos for AWS Load Balancer Controller
* You will master the AWS Fargate Concepts with 3 demos including running all workloads of EKS Cluster on AWS Fargate (Fargate Only EKS Cluster)
* You will install Kubernetes Cluster Autoscaler on AWS EKS Cluster with Terraform and Test it
* You will implement Horizontal and Vertical Pod Autoscaler Concepts with Terraform
* You will learn to implement AWS EKS Monitoring and Logging using kubectl and Terraform

**Github Repositories used for this course**

* [Terraform on AWS EKS Kubernetes IaC SRE- 50 Real-World Demos](https://github.com/stacksimplify/terraform-on-aws-eks)
* [Course Presentation](https://github.com/stacksimplify/terraform-on-aws-eks/tree/main/course-presentation)
* [Kubernetes Fundamentals](https://github.com/stacksimplify/kubernetes-fundamentals)
* **Important Note:** Please go to these repositories and FORK these repositories and make use of them during the course.