**AWS Landing Zone deployment using terraform**

|  |  |
| --- | --- |
| **Work Effort Name** | AWS landing zone deployment using terraform |
| **Prepared By** | KAVYA S |
| **Document Issued Date** | 05-03-2024 |
| **Document Version** | 1 |
| **Reviewed By** | Mahavishnu Govindaraj |
| **SA Certificate Number** |  |

| **Version No.** | **Date** | **Revised by** | **Description of Changes** |
| --- | --- | --- | --- |
| 1. | 11-03-2024 | KAVYA S | * Update the project requirement. * Remove unwanted space and grammatical errors. * Use common fonts across the document. * Revisit the backup and retention. * How AWS control tower is going to be setup. * Is feasibility check done for solution overview in page 9? * How landing zone is going to be setup and mention the architect diagram if it is going to be setup. * Revisit the entire document once and align it with the project. |
|  |  |  |  |
|  |  |  |  |

Contents

[1. Background & Introduction 4](#_Toc90380478)

[1.1 Project requirements 4](#_Toc706589727)

[1.2 Project Scope 4](#_Toc1456981679)

[1.3 AWS Control Tower Overview 6](#_Toc977368325)

[1.4 Account provisioning flow 6](#_Toc96122217)

[1.5 Contacts 7](#_Toc918752495)

[1.6 Abbreviations & Keywords 7](#_Toc1700163018)

[1.7 Solution overview 8](#_Toc1967322466)

[2. AWS Account 10](#_Toc1563631947)

[2.1 IAM Users & Password Policy 10](#_Toc510331282)

[2.1.1 IAM Users 10](#_Toc1363637391)

[2.1.2 Password Policy 10](#_Toc1907000190)

[3 Architecture Diagrams 11](#_Toc780137118)

[3.1 Account factory for terraform - Account provisioning flow 11](#_Toc1320866345)

[3.2 Infra Design Diagram 12](#_Toc1401439837)

[3.3 Accounts under organizational units 13](#_Toc1018950957)

[3.4 Best practices 13](#_Toc755633160)

[4 Network & Design Security 15](#_Toc1090661981)

[4.1 Virtual Private Cloud (VPC) 15](#_Toc359454376)

[4.2 Root account Subnet List 15](#_Toc683605297)

[4.3 Prod Subnet List 15](#_Toc386101577)

[4.4 Non prod Subnet List 16](#_Toc541478898)

[4.5 Network ACL (NACL) 16](#_Toc288745611)

[4.6 Route Table 16](#_Toc60121812)

[4 Governance and Security 17](#_Toc1204054701)

[5.1 Log Management 17](#_Toc1047581728)

[5.2 AWS Config 17](#_Toc1123565839)

[5.3 CloudTrail 17](#_Toc1899605943)

[5.4 KMS 17](#_Toc823512939)

[5.5 Security hub 18](#_Toc1235569558)

[5.6 Guard duty 18](#_Toc1929483528)

[5.7 AWS Cloud watch 18](#_Toc913893417)

# Background & Introduction

## Project requirements

Using AWS Control Tower and Terraform, the project will construct a standardized AWS Landing Zone infrastructure with an emphasis on networking, security, and effective account administration. To demonstrate effective infrastructure provisioning, thorough documentation is essential. This includes Terraform configuration files, network diagrams, IAM Policies, and deployment logs. In order to implement policies, the organizational structure entails the construction of Organizational Units (OUs) such as Log, Security, Prod, and Non-Prod, in addition to Service Control Policies. Best practices for network security should be incorporated into the design of the VPC architecture, and consistency is guaranteed by automatic account provisioning via AWS Control Tower's Account Factory. Important elements include AWS CloudTrail logging configurations, IAM role and permission management, secret management techniques, and compliance monitoring. The project necessitates a careful methodology that makes use of both AWS Control Tower and Terraform to establish a secure and standardized AWS environment.

## 1.2 Project Scope

The scope of the project includes the design and implementation of a standardized AWS landing zone infrastructure using Terraform and AWS Control Tower. The primary focus is on incorporating best practices for security, networking, and account management including considerations such as OU structure ,Service Control policies ,tagging policies enforcement,IAM roles and permission management.

1. **AWS Control Tower Environment Setup -** Implementation of AWS Control Tower to establish a multi-account AWS environment. Creation of organizational units (OUs) for logical segregation of accounts based on business requirements.
2. **Terraform Module Development -** Development of modular Terraform modules for various infrastructure components, promoting code reusability and scalability.
3. **VPC Architecture Design -** Definition of a standardized VPC architecture, including subnets, routing, and considerations for high availability and fault tolerance.
4. **Network Security Measures -**  Configuration of network security measures such as security groups, network access control lists (NACLs), and network flow logs.
5. **Automated Account Provisioning -** Leveraging AWS Control Tower, including the Account Factory, for automated provisioning of AWS accounts.
6. **IAM Role and Permission Management -** Designing IAM roles and policies for different personas and services, adhering to the principle of least privilege.
7. **Service Control Policies (SCPs) -** Establishment of SCPs to enforce fine-grained permissions and control access to AWS services.
8. **Tagging Policies Enforcement -** Implementation of tagging policies to ensure consistent tagging across resources. Utilization of Terraform for enforcing and managing tagging policies.
9. **Secret Management Strategies -** Utilization of AWS Secrets Manager or AWS Key Management Service (KMS) for secure storage of sensitive information.
10. **Auditing and Compliance -** Configuration of AWS CloudTrail for auditing AWS API activity.Implementation of AWS Config Rules for continuous compliance monitoring.
11. **Logging Configuration -** Configuration of AWS CloudWatch Logs for centralized logging.
12. **Organizational Structure -** Definition of the organizational structure based on business requirements.Implementation of AWS Control Tower OUs for logical grouping of accounts.
13. **Terraform Configuration Files -** Creation of organized and maintainable Terraform configuration files.
14. **Account Factory for Terraform -** Integration of the Account Factory feature in AWS Control Tower for automated and consistent provisioning of AWS accounts. Configuration of account baseline settings and policies using the Account Factory.

## 1.3 AWS Control Tower Overview

*AWS Control Tower is a service provided by Amazon Web Services (AWS) that simplifies the process of setting up and governing a secure and multi-account AWS environment. It helps organizations adopt AWS best practices for security and compliance by automating the setup of a well-architected, multi-account AWS environment.*

Automated Account Provisioning: By automating the setup of new AWS accounts, Account Factory for Terraform makes sure that every account is provided reliably and in accordance with organization’s guidelines.

Standardized Environment: To encourage uniformity and best practices, AWS Control Tower creates a standardized landing zone with basic configurations and infrastructure.

Compliance and Security: By restricting access to AWS services and resources, SCPs and guardrails assist in enforcing compliance and security standards throughout the company.

Organizational Structure: OUs offer an organized and controlled environment by allowing you to rationally arrange AWS accounts according to projects, business units, or other factors.

Governance and control: AWS Control Tower offers a centralized hub for governance and control, enabling businesses to successfully establish and implement policies.

## 1.4 Account provisioning flow

Step 1: New Account Request - Users initiate the process by submitting a new account request.

Step 2: Account Request Repository - The account request details are stored in the Account Request Repository for tracking and management.

Step 3: Account Request Pipeline - The request enters an automated pipeline for further processing.

Step 4: Request Table and Trigger Function - The request details are recorded in the request table, triggering the request processing function.

Step 5: Control Tower Update Handling - If there's a Control Tower update, the request is sent to the Account Request Queue. A dedicated Account Request Processor handles this, and the request is forwarded to AWS Service Catalog under the Control Tower Management Account.

Step 6: No Control Tower Update Handling - If there's no Control Tower update, the request proceeds to the Account Provisioning Invoker.

Step 7: Account Provisioning State Machine - The request is directed to the Account Provisioning State Machine, and the pipeline creation is initiated in the CodeBuild stage.

Step 8: Customization Invocation State Machine - For new customization requests, the process is directed to the Customization Invocation State Machine and sent to the invocation pipeline.

Step 9: Customizations Handling - If customizations are specified in the initial account provisioning request, they run only on targeted accounts. For previously provisioned accounts, additional customizations must be initiated manually in the accounts pipeline.

Step 10: AFT Operations in Management Account - The entire Account Factory for Terraform(AFT) operations are conducted within the AFT Management Account.

## 1.5 Contacts

|  |  |  |  |
| --- | --- | --- | --- |
| **Owner** | **Role** | **Name** | **Email ID** |
| 1CH | Project Manager | MAHA VISHNU G | vishnu@1cloudhub.com |

Table-1 : Contacts

## 1.6 Abbreviations & Keywords

The following are the terminology and acronyms are used in the document.

|  |  |  |
| --- | --- | --- |
| S.NO | Abbreviation / Keyword | Definition |
| 1 | AWS | Amazon Web Services |
| 2 | IAM | Identity Access Management |
| 3 | VPC | Virtual Private Cloud |
| 4 | S3 | Simple Storage Service |
| 5 | AFT | Account Factory for Terraform |
| 6 | ACL | Access Control List |
| 7 | IGW | Internet Gateway |
| 8 | NAT | Network Address Translation |
| 9 | OU | Organizational Units |
| 10 | SSO | Single Sign On |
| 11 | SCP | Service Control Policies |

Table-2 : Abbreviations & Keywords

## 1.7 Solution overview

STEP -1 **Prepare AWS Environment -** Set up an AWS account and ensure necessary permissions for AWS Control Tower.

STEP - 2 **Install AWS CLI and Terraform.**

**STEP - 3 Set Up AWS Control Tower.**

**STEP - 4 Organize Organizational Units (OUs) in AWS Control Tower -** Define OUs like "Log," "Security," "Prod," and "Non-Prod" in AWS Control Tower.

STEP - 5 **Create Terraform Code Repository -** Set up a version-controlled repository(GITHUB) to store Terraform code.

**STEP - 6 Design VPC Architecture -** Develop Terraform modules for VPC architecture in the root account considering security and scalability.

STEP - 7 **Define IAM Policies -**  Write Terraform configurations for IAM roles and policies following the principle of least privilege.

**STEP - 8 Implement Service Control Policies (SCPs) -** Configure Terraform modules to enforce Service Control Policies in AWS Control Tower.

**STEP - 9 Set Up Tagging Policies -** Define Terraform configurations to enforce tagging policies for consistent resource tagging.

STEP - 10 **Automate Account Provisioning -** Use Terraform to create modules for the Account Factory within AWS Control Tower for automated AWS account provisioning.

**STEP - 11 Implement Network Security Measures -** Configure Terraform modules for network security measures, including NACLs and Security Groups.

**STEP - 12 Enable Auditing and Compliance -** Set up AWS CloudTrail and AWS Config for auditing and compliance monitoring. Use Terraform to configure logging and centralize logs in Amazon S3.

**STEP - 13 Secret Management Strategies -** Implement Terraform configurations for secret management using AWS Secrets Manager or other strategies.

STEP - 14 **Deploy the Landing Zone -** Execute Terraform commands to deploy the landing zone infrastructure, including VPCs, IAM roles, and network configurations.

# AWS Account

Below are the AWS accounts for Landing zone setup.



|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **Account Number** | **Account Name** | **Description & Purpose** |
| 1 |  | Common service | Common services |
| 2 |  | Management | Master account/Root account |
| 3 |  | Production account | Hosts live customer facing workload |
| 4 |  | Non Production account | Development,testing and staging activities |

Table-3 : AWS Accounts

## 2.1 IAM Users & Password Policy

### 2.1.1 IAM Users

|  |  |  |
| --- | --- | --- |
| **AWS Account Name** | **Username** | **Policies** |
| Prod, Non-Prod and Management,Security,Common service |  | Power users |

Table-4: IAM Users

### 2.1.2 Password Policy

Password policy for AWS accounts

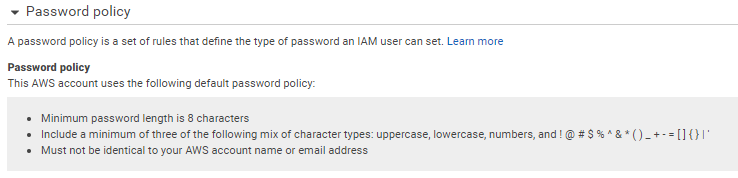
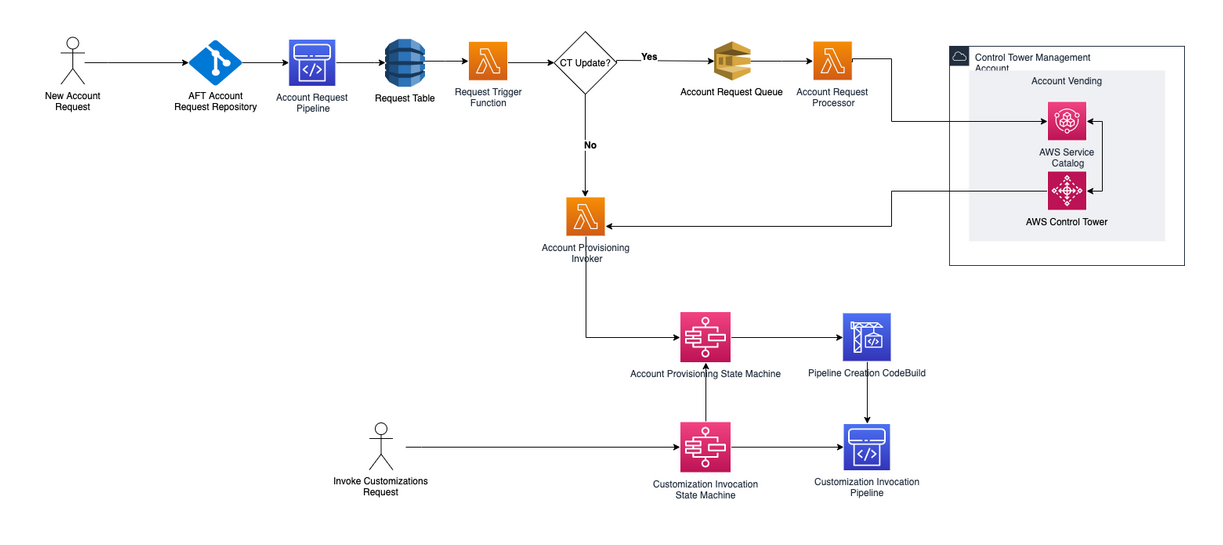


Table-5: Password policy enabled in AWS accounts

# Architecture Diagrams



## 3.1 Account factory for terraform - Account provisioning flow



## Infra Design Diagram

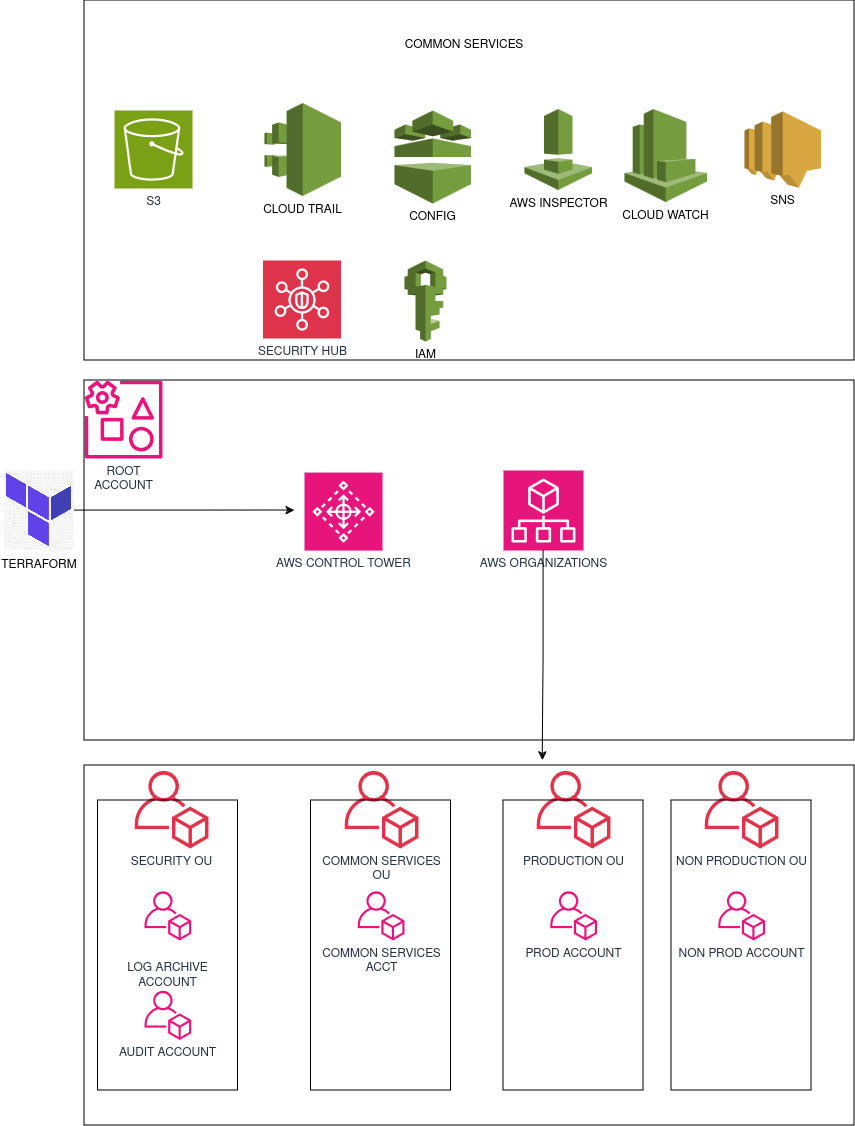


Table-8: Infra Design Diagram

## Accounts under organizational units

|  |  |
| --- | --- |
| **Accounts** | **Functions** |
| Production account | This account is dedicated to hosting production workloads and applications. |
| Non production account | Used for development, testing, and staging environments |
| Security account | Centralized account for security-related services and monitoring. |
| Common services account(Network account) | It serves as a central hub for the network routing between the multi account in landing zone accounts. |
| Master account (Root account) | It is the payer account that manages the entire organization structure. |

## 3.4 Best practices

|  |  |  |
| --- | --- | --- |
| **Category** | **Consideration** | **Services Used** |
| **AWS Control Tower and Account Factory** | * Establishment and enforcement of governance guardrails. * Leverage of AWS Control Tower for a consistent multi-account environment. * Usage Account Factory for automated and standardized account creation. | AWS Control Tower  AWS Organizations  AWS Account Factory |
| Secret Management Strategies | * Use AWS Secrets Manager for secure storage and management of sensitive information. | AWS Secret Manager |
| Automated auditing and compliance | * Implement AWS Config for continuous monitoring of account configurations. * Use AWS Config Rules for custom compliance checks. | AWS Config  AWS Config Rules |
| **Tagging Policies Enforcement** | * Enforce tagging policies through AWS Organizations to standardize resource tagging. | AWS Resource groups  AWS Organizations Tag Policies. |
| **IAM Roles and Permission Management** | Principle of least privilege is followed when defining IAM roles and permissions.Regularly audit and review IAM policies for security. | AWS IAM |

# Network & Design Security



## 4.1 Virtual Private Cloud (VPC)

*Virtual private cloud (VPC) is an on-demand configurable pool of shared computing resources allocated within a public cloud environment, providing a certain level of isolation between the different organizations using the resources.*

|  |  |  |  |
| --- | --- | --- | --- |
| **VPC** | **Region** | **VPC CIDR** | **VPC ID** |
| awsroot\_vpc | N VIRGINA | 192.32.0.0/16 |  |
| prod\_vpc | N VIRGINA | 192.33.0.0/16 |  |
| nonprod\_vpc | N VIRGINA | 192.34.0.0/16 |  |

Table-9: VPC Details

## 4.2 Root account Subnet List

*Subnet is a logical partition of an IP network into multiple, smaller network segments and it is tightly linked to IP addresses, subnet masks, and CIDR notation.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Subnet Tier** | **Subnet Name** | **Subnet CIDR** | **Available IPs** | **Subnet ID** | **Availability zone** |
| Public Subnet | public\_subnet\_1 | 192.32.3.0/24 | 249 |  | us\_east\_1a |
| Public Subnet | public\_subnet\_2 | 192.32.4.0/24 | 249 |  | us\_east\_1b |
| Private Subnet | private\_subnet\_1 | 192.32.1.0/24 | 249 |  | us\_east\_1a |
| Private Subnet | private\_subnet\_2 | 192.32.2.0/24 | 249 |  | us\_east\_1b |

## 4.3 Prod Subnet List

*Subnet is a logical partition of an IP network into multiple, smaller network segments and it is tightly linked to IP addresses, subnet masks, and CIDR notation.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Subnet Tier** | **Subnet Name** | **Subnet CIDR** | **Available IPs** | **Subnet ID** | **Availability zone** |
| Public Subnet | prod\_pubsubnet\_1 | 192.33.3.0/24 | 249 |  | us\_east\_1a |
| Public Subnet | prod\_pubsubnet\_2 | 192.33.4.0/24 | 249 |  | us\_east\_1b |
| Private Subnet | prod\_prisubnet\_1 | 192.33.1.0/24 | 249 |  | us\_east\_1a |
| Private Subnet | prod\_prisubnet\_2 | 192.33.2.0/24 | 249 |  | us\_east\_1b |

## 4.4 Non prod Subnet List

*Subnet is a logical partition of an IP network into multiple, smaller network segments and it is tightly linked to IP addresses, subnet masks, and CIDR notation.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Subnet Tier** | **Subnet Name** | **Subnet CIDR** | **Available IPs** | **Subnet ID** | **Availability zone** |
| Public Subnet | nprod\_pubsubnet\_1 | 192.34.3.0/24 | 249 |  | us\_east\_1a |
| Public Subnet | nprod\_pubsubnet\_2 | 192.34.4.0/24 | 249 |  | us\_east\_1b |
| Private Subnet | nprod\_prisubnet\_1 | 192.34.1.0/24 | 249 |  | us\_east\_1a |
| Private Subnet | nprod\_prisubnet\_2 | 192.34.2.0/24 | 249 |  | us\_east\_1b |

## 4.5 Network ACL (NACL)

*It is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets.*

*Currently we are not blocking any incoming and outgoing traffic using NACL*

## 4.6 Route Table

*A route table contains a set of rules, called routes, that are used to determine where network traffic is directed. Each subnet in your VPC must be associated with a route table.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Route Table Name** | **Destination** | **Target** | **Subnet Association** |
|  |  | local |  |
|  | tgw- |
|  | igw- |

Table-17: Route Table Rules

# Governance and Security

## 5.1 Log Management

Enabled To track the event history of AWS account activity of any API activity in all OUs. Below are the S3 buckets for log management for Audit trail.

**Note:** Since CloudTrail provides only 90 days so log stored in bucket is recommended.

|  |  |  |
| --- | --- | --- |
| **Management S3 Bucket Name** | **Purpose** | **Retention (Days)** |
|  | Config audit | 30 |
|  | VPC flow log audit | 30 |
|  | CloudTrail | 30 |

## 5.2 AWS Config

*AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations.*

## 5.3 CloudTrail

*Enabled To track the event history of AWS account activity of any API activity.*

|  |  |  |
| --- | --- | --- |
| **S.No** | **Cloud Trail status** | **Account details** |
| 1 | Enabled | Prod |
| 2 | Enabled | Non-Prod |
| 3 | Enabled | Common service |
| 4 | Enabled | Management |
| 5 | Enabled | Security |

## 5.4 KMS

*AWS Key Management Service (KMS) is an Amazon Web Services product that allows administrators to create, delete and control keys that encrypt data stored in AWS databases and products. AWS KMS provides a single view of all AWS keys in use, creating centralized encryption control.*

**AWS Managed Keys**

|  |  |  |  |
| --- | --- | --- | --- |
| **Aliases** | **Key ID** | **Status** | **[Key spec](https://docs.aws.amazon.com/kms/latest/developerguide/symm-asymm-choose.html" \l "symm-asymm-choose-key-spec" \t "_blank)** |
| aws/codepipeline |  | Enabled | SYMMETRIC\_DEFAULT |
| aws/S3 |  | Enabled | SYMMETRIC\_DEFAULT |
| aws/backup |  | Enabled | SYMMETRIC\_DEFAULT |

## 5.5 Security hub

*AWS Security HUB used to comprehensive view of your high-priority security alerts and compliance status across AWS accounts.  With Security Hub, you now have a single place that aggregates, organizes, and prioritizes your security alerts, or findings, from multiple AWS services, such as Amazon GuardDuty, Amazon Inspector, and Amazon Macie, as well as from AWS Partner solutions.*

*With Security HUB we have enabled the CIS AWS foundational Benchmark and AWS Foundational Security has enabled in all accounts.*

## 5.6 Guard duty

*Detect and respond to malicious activity and unauthorized behavior and analyze findings for security investigations.*

## 5.7 AWS Cloud watch

*AWS Cloudwatch is a service for monitoring of resources and services.It is a collection of performance data. It gives and alarm or react to any unexpected behaviour.*