Table of Contents

[Design Decisions 1](#_Toc93783788)

[Explanation of Modules 1](#_Toc93783789)

[VPC 1](#_Toc93783790)

[S3 1](#_Toc93783791)

[EC2 2](#_Toc93783792)

[IAM 2](#_Toc93783793)

[ASG 2](#_Toc93783794)

[ALB 2](#_Toc93783795)

[KMS 2](#_Toc93783796)

# Design Decisions

* The decision was made to not include tfvars to simplify deployment and increase portability of code
* Security Groups were moved inline to increase readability of code and reduce complexity
* An additional S3 Bucket was included specifically for logging purposes (VPC and ALB)
* An IAM role was created for SSM Session Manager added for ease of management and testing
* Only the root module contains a main.tf for simplicity of reference
* All config was moved into sub modules, the root module only passes variables and calls on sub modules for configuration

# Explanation of Modules

Below is a brief overview of each module and most of the issues that where ran into with helpful links

## VPC

* VPC Module was created first, this is standard config and additional information can be found here on creation <https://registry.terraform.io/modules/terraform-aws-modules/vpc/aws/latest>
* After basic configuration was created and initial run of Checkov, the code was updated with VPC flow log configuration added this config was straightforward and can be found here <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/flow_log>

## S3

* This was straightforward until I started writing the bucket policy for ALB logging
  + This was fixed by following <https://stackoverflow.com/questions/56751080/terraform-setting-up-logging-from-aws-loadbalancer-to-s3-bucket>
* The S3 Image bucket was created next, this took a fairly short amount of time following <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/s3_bucket>
  + Small caveat, I did have to re-read this multiple times to see how they included multiple lifecycle rules. I believe this is a newer feature as I have a distinct memory of running into this issue in the past.

## EC2

* This was pretty straightforward, the only issue that was ran into was that t2.micro is not supported for RHEL. Decision was made to move to t3.micro.

Unsupported: The requested configuration is currently not supported. Please check the documentation for supported configurations.

## IAM

* The IAM role was created roughly halfway through troubleshooting the EC2 instance.
* <https://cloudanddevopstech.com/2020/11/01/terraform-aws-ec2-with-ssm-agent-installed/> was used which has a great writeup

## ASG

* This module came together with few issues except the userdata script would not be accepted
* Userdata script was resolved by following: <https://github.com/hashicorp/terraform-provider-aws/issues/5530>
* All other documentation used:
  + <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/launch_template>
  + <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/autoscaling_group>
  + <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/autoscaling_attachment>

## ALB

* The only issue I ran into here was with S3 Logging, documentation used was:
  + <https://registry.terraform.io/modules/terraform-aws-modules/alb/aws/latest>
  + <https://github.com/terraform-aws-modules/terraform-aws-alb/blob/master/examples/complete-alb/main.tf>

## KMS

* This was the last module created, at this point I had deployed all resources and confirmed no issues where present. The 2 below links where initially followed for all except the kms policy:
  + <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/kms_key>
  + <https://gist.github.com/sunnyneo/063f7553177a2a75c78657adc7ca23a9>
* I proceeded to manually update the KMS key policy following the answer on <https://serverfault.com/questions/1000686/how-to-auto-scale-ec2-instances-with-an-encrypted-root-volume>
* After confirming the ASG launched successfully, I updated the policy in Terraform