Documentation

Question 1

Created a Github repo named under heksx, my rendition of HTX EKS as to make it more discrete.

Region chosen in ap-southeast-1 (Singapore) for high availability since it is in same country as this project is for HTX, choosing a data center in Singapore allows better control for our data. But I do see a downside where if something were to happen like regional outage or natural disasters (though slim), this will cause all operations to stop.

Module “eks” is used to simplify the creation and for the versioning, read the documentation for this release and find news to ensure that v20 is okay to use. Decided to choose v20 instead of 1 version down as to reduce upgrade headache in the future (ref 1).

Same goes for cluster\_version v1.31 (ref 2).

Variable cluster\_endpoint\_public\_access  = false and cluster\_endpoint\_private\_access = true

added to ensure that no public access into cluster endpoint. Cluster endpoint is only accessible via VPC.

eks\_managed\_node\_groups managed the node group with their specifications and tags. Min size of 2 to have at least 2 nodes as stated in requirements. For the instance type, m6i.large was chosen by me as even though I may not understand what algorithm is being used here, I am taking it as an early implementation. So I am using m6i.large for a lightweight low cost instance. This instance has 8GB memory so it should be able to run basic pre processing and ML tuning. If the algorithm becomes larger and require a higher compute or memory intense instance, I will upgrade them to g5 series. G5 is said to give good balanced power with good cost efficiency. Even higher performance will require choice of p3dn or p4d (ref 3).

For 2 availability zone, I will include it in Question 2’s VPC.

Question 2

VPC will handle high availability aspect in question 1. The availability zones chosen is both Singapore as I would not want to take the risk of having data out of Singapore. The cidr I chose is from 10.0.xxx.xxx range, this is normally used for production work (not always the case). private\_subnets and public\_subnets are defined to allow different entry and access points for internal communications and internet facing interactions. enable\_nat\_gateway was added in case private subnets requires access to internet. Could change to false if deemed not needed. enable\_vpn\_gateway was considered to be added as I believe all internet access require VPN will protect the nodes better but thinking against, it will be a double edged sword if the VPN is not set up correctly. It could be part of a future enhancement. Tags added to point to correct project (ref 4).

Question 3

Following the best practices, least privilege principle. First, creating a EKS cluster role for management of EKS cluster. Attached AssumeRole, added SID for debugging in the future and Principal to point to EKS service (ref 5, ref 6). Permission given is using Amazon Managed Policy, AmazonEKSClusterPolicy (ref 7). Considered writing my own policy by copy pasting from AWS website but I feel if there were to have any changes in permission required to manage EKS on Amazon side, our system would not break.

Next, creating a role for worker nodes to communicate with the EKS services. It is another AssumeRole, added SID for debugging and Principal to point to EC2 service. 3 policy is attached into this role; AmazonEKSWorkerNodePolicy (worker node), AmazonEC2ContainerRegistryReadOnly (ECR policy), and AmazonEKS\_CNI\_Policy (CNI policy). Worker node policy allows all permissions in EC2 with addition of eks:DescribeCluster (ref 8) because for worker node to communicate with EKS this permission is needed. ECR policy is granted so that the node can pull container images from ECR for EKS. Lastly, in order to manage network layer, CNI policy is added. This policy will be given to worker role to handle networking issues. Things like assigning IP address and modifying Elastic Network Interfaces (ref 9).

Reference

ref 1: https://github.com/terraform-aws-modules/terraform-aws-eks/releases

ref 2: <https://docs.aws.amazon.com/eks/latest/userguide/kubernetes-versions-standard.html>

ref 3: <https://www.reddit.com/r/MachineLearning/comments/11f0zs6/d_which_ec2_instance_types_do_you_use_for/>

ref 4: <https://github.com/terraform-aws-modules/terraform-aws-vpc>

ref 5: <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/iam_role>

ref 6: <https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/eks_cluster>

ref 7: <https://docs.aws.amazon.com/aws-managed-policy/latest/reference/AmazonEKSClusterPolicy.html>

ref 8: <https://docs.aws.amazon.com/aws-managed-policy/latest/reference/AmazonEKSWorkerNodePolicy.html>

ref 9: <https://docs.aws.amazon.com/aws-managed-policy/latest/reference/AmazonEKS_CNI_Policy.html>