**List Of Labs Topic wise**

**EC2 Lab List**

1. **Create a simple EC2 instance using the AWS Management console and assign new security group.**
2. **Create a simple EC2 instance using the AWS Management console. Connect to the instance using SSH(Linux) .**
3. **Create a simple EC2 instance using the AWS Management console. Connect to the instance using RDP (windows).**
4. **Connect Ec2 instance using Mobaxterm or putty and host sample html application.**
5. **Create a custom Amazon Machine Image (AMI) from an existing EC2 instance. Launch new instances from your custom AMI.**
6. **Create EBS volume and attach to any one instance.**
7. **Create EBS volume in AZ and attach to AZ b instance.**
8. **Create launch template with customized AMI.**
9. **Create Two instances from Launch template and ensure same applications for both Ec2's.**
10. **Migrate Ec2 from one region to another region.**
11. **Take the backup of Snapshot and AMI for every day 2pm IST for two days only (use Life cycle manger)**
12. **Clear all resources.**

**Optional 1. Create one Ec2 machine and host HTML content after that detach root volume and attach to other new Ec2 instance do let me know the result. ( other ec2 instance root volume should be detached before attaching)**

1. **How to access your Ec2 instances even if your pem file is lost.**
2. **Create launch template with customized AMI and launch instance from the LT.**
3. **Migrate Ec2 from one region to another region.**
4. **Swap the root volumes of two instances.**
5. **Create ec2 Image builder.**
6. **Crate capacity reservation for ec2**
7. **Create spot request for ec2 instance.**
8. **Create Reserved ec2 instance.**
9. **Create elastic Ip and assign to ec2 instance and stop ec2 instance and start it again observe the change in public IP.**
10. **Create ENI and assign two privates Ip’s for ec2 instance using ENI.**
11. **Create Target group and register Ec2 instance as a target. (try for both public and private instances)**
12. **Create multiple target group for different AZ’s and register to the public Ec2 instance as a target. (try for both public and private instances)**
13. **Create ALB and configure to the target groups.**
14. **Create NLB and configure to the target group.**
15. **Do lab for ALB , ELB configuration for multiple target groups in different AZ’s**
16. **Create stickiness of load balancer and check target traffic.**
17. **Set up path-based routing on your ALB to route traffic to different target groups. Test the routing by creating multiple paths and associating them with different target groups.**
18. **Create manual scaling for targets and increase and decrease instances.**
19. **Create an Auto Scaling group and configure it to work with your ALB.**
20. **Set up scaling policies to automatically adjust the number of instances by using target tracking policy (CPU)**
21. **Create a scheduled base auto scaling group and check weather it is launching or not instances as per min and max policy.**
22. **Integrate AWS Web Application Firewall (WAF) with your ALB to protect against common web application attacks. Create and test WAF rules to block or allow specific traffic patterns.**
23. **Modify the health check settings for your target group to see how it affects the behaviour of the ALB. Simulate a target instance failure and observe how the ALB reacts.**
24. **Create warmup instance and allocate to Auto scaling policy check the changes.**
25. **Create a Dynamic policy scaling for Ec2 instances (Target tracking, Step scaling and simple scaling policy)**
26. **Create an Ec2 instance using IAM user and connect the instance using SSH.**
27. **Create an IAM role with permissions.**
28. **Attach the IAM role to an EC2 instance using an instance profile and access S3 bucket.**
29. **Set up CloudWatch alarms to monitor EC2 instance metrics.**
30. **Create custom CloudWatch metrics and dashboards.**
31. **Host a sample application in private instance and client should be connected using ELB and application should run without zero down time.**
32. **Create a Network Load Balancer, configure listeners and associate target groups and route traffic based on port numbers.**
33. **Create EFS file system and mount the volume and compare with EBS volume.**

**IAM**

1. **Create IAM user and check both console and CLI based access.**
2. **Create a IAM user and give permissions to check both console and CLI based able to access or not as for granted policy.**
3. **Attach existing default policies and access a IAM by both console and CLI based able to access or not as for granted policy.**
4. **Create inline policy and see the difference.**
5. **Create custom policy and give more controls like object wise bucket wise try with different permissions see the difference.**
6. **Create custom policy to access only one bucket and give more control inside obj like get, put...etc**
7. **Create user and give full admin access and login as a user to check whether all permissions are access or not?**
8. **Create boundary set with permission s3 full access or ec2 full access and check access as a user level**
9. **Create a IAM role with required policy permissions, connect ec2 to S3 by using role.**
10. **Using IAM role with required policy permissions and using endpoint connect from private EC2-S3**
11. **Create a IAM group give permissions to group and add user to group and check user level access or not.**
12. **Create IAM user and add MFA authentication.**
13. **Create user and try to access IAM switch role.**
14. **Create IAM role for cross account access.**
15. **Create role for aws lambda and attach to sns policy to get notification when lambda runs.**

**AWS S3 (Simple Storage Service)**

1. **Create AWS S3 bucket with default options.**
2. **Create a folder and subfolder for the bucket.**
3. **Upload any sample image into above folder structure and give image to**

**public access. (Preassigned Object URL)**

1. **Download the image which is in the aws cloud bucket.**
2. **Host a Static website by enabling static website hosting for a bucket.**
3. **Enable the versioning of s3 bucket and check upload file and delete file.**
4. **Check the variation before and after enabling S3 version.**
5. **Configure object life cycle policies to automatically transition objects to**

**different storage classes**

1. **Create two buckets and copy the files from one bucket to other bucket (same region, other region and other account).**
2. **Set up cross-region replication to replicate objects from one s3 bucket to**

**another in a different aws region.**

1. **Create s3 bucket access point.**
2. **Perform the lab on S3 inventory configuration.**
3. **Create and apply bucket policies to control access to your s3 resources.**
4. **Experiment with IAM roles and permissions for fine-grained control.**
5. **Set up S3 event notifications and trigger when objects are created, updated,**

**and deleted.**

1. **Create SNS topic and subscribe to your mail ID or phone SMS.**
2. **Enable S3 bucket access logs and check logs into given bucket.**
3. **Enable server – side encryption to protect data at rest. Experiment with**

**client-side encryption for data protection.**

1. **Implement security practices, such as bucket policies, access control list.**

**and VPC endpoints for enhanced security and compliance.**

1. **Configure CloudTrail data events to log Amazon S3 object –level API operations in the CloudTrail console.**
2. **Monitor your S3 using CloudWatch.**

**KMS (Key Management System)**

1. **Create KMS customer managed keys for IAM user and keys should be in symmetric (encryption & description)**
2. **Create KMS customer managed keys for IAM user, Role and keys should be in symmetric (encryption & description). Enable KMS encryption when creating the EBS volume.**
3. **Enable of KMS encryption for existing EBS volume be help of Snapshot.**

**Enable KMS encryption to S3 bucket.**

1. **Create KMS for one S3 bucket and check it by upload, delete file, check in another User without KMS access.**
2. **Create two IAM users A, B then you can enable to KMS encryption for user A.**

**You have to test both the users A & B able to upload the file or not in S3 bucket.**

**WAF (Web Application Firewall)**

1. **Create IP set in WAF to block IP using application firewall to block ALB.**
2. **Create WAF for ALB using IP set and test the application data.**
3. **Block your laptop ip using WAF.**

**CloudFront**

1. **Create CloudFront using ALB then test App.**
2. **Create CloudFront using Origin base then test App.**
3. **Create CloudFront using two Origin base then test S3.**
4. **Create CloudFront using ALB and enable WAF, and you can test the App.**
5. **Create CloudFront and configure with EC2 and S3**
6. **Create CloudFront with cache policy and origin then test App!**
7. **CloudFront configure with route53 redirect to ALB then test the App.**

**CloudTrail**

1. **Create CloudTrail with S3 New bucket then test the Event logs.**
2. **Create CloudTrail with S3 New bucket enable KMS, SNS services then test the Event logs.**
3. **Create CloudTrail and check event history details.**
4. **Create cloud trail to S3 bucket and check after every 15 minutes log update’s in s3 bucket.**

**CloudWatch**

**# Alaram**

1. **Create CloudWatch alarms to log metrics, CPU utilization, CPU credits ...etc add SNS topic for alarms**
2. **Create CloudWatch Alarm for ec2 metrics on CPU utilization. You can test able to get a notification from alarm.**
3. **Create CloudWatch Alarm for ALB metrics. You can test able to get a notification from alarm.**
4. **Create CloudWatch Alarm for EBS metrics. You can test able to get a notification from alarm.**
5. **Create CloudWatch Dashboard for all ec2, ALB, S3**
6. **Enable detail monitoring from ec2.**
7. **Create log group to check the logs to know application or service status.**
8. **create rule to automate detailed data points like when volume will delete immediately snapshot should create.**
9. **Create scheduled rule to run a service for particular time interval using CRON job.**

**VPC Endpoint**

1. **Create VPC endpoint and attach to VPC and private route table and access s3 default full access without internet.**
2. **Create VPC endpoint and attach to VPC and private route table and try to access s3 custom policy for bucket without internet.**

**VPC (Virtual Private Cloud)**

1. **Create custom VPC and allow CIDR range.**
2. **Create Internet gate way and attached to VPC.**
3. **Create public and private subnets to custom CIDR range.**
4. **Create route tables to allow public and private access.**
5. **Create Nat gateway and allow internet from private subnet.**
6. **Create Flow Log in VPC &amp; send it to Amazon S3 Bucket (destination)**

**(You can attach IAM Role also)**

1. **Create NACL and enable inbound and outbound rules and check the traffic accordingly.**
2. **Create deny rule for NACL and check the changes also compare with Default NACL.**
3. **Compare NACL and SG with lab.**
4. **Create VPC peering for two VPC’s for same account and same or other regions and try to access for one service to other service between two VPC’s like SSH or Ping.**
5. **Create VPC peering for two VPC’s for other account and same or other regions and try to access for one service to other service between two VPC’s like SSH or Ping.**
6. **Create Customer gateway and Virtual private gateway.**
7. **Create Site-site VPN connection to establish on prem to Cloud connection.**
8. **Create client VPN endpoint to establish connection between On Prem (work from home) to Cloud**
9. **Create Transit gateway to establish multi peering for VPC’s like one to multi(Optional explore and complete by own way)**

**Amazon Route 53**

1. **Create Public hosted zone for Route 53**
2. **Create Private hosted zone for Route 53**
3. **Copy the public hosted zone name server records and paste it into domain name server records (where created domain name registry website)**
4. **Start the creating of records and map into public IP of ec2 instance and access with domain name check weather getting or not.**
5. **Create record for ALB to map ALB URLs by using alias records.**
6. **Create CNAME record and configure to ALB or NLB and access with domain name.**
7. **Create a simple routing policy for Ec2 instance and Load balancer.**
8. **Create Latency based routing policy for ALB or NLB or Ec2 instance IP observe latency status.**
9. **Create weighted based routing policy for ALB or NLB or Ec2 instance IP observe weighted policy status.**
10. **Create geolocation-based routing policy for ALB or NLB or Ec2 instance IP observe geolocation policy status.**
11. **Create health checks and configure to failover routing policy and configure to two different region ALBs or NLB’s or EC2’s and check two regions health status.**
12. **and stop the one ALB or Instance check weather traffic is rerouting or not like High available achieving or not for cross region.**
13. **Compare with Route 53 and Load balancer with High availability labs.**
14. **Use your domain in Route 53 and configure to static website hosting for S3**
15. **Configure CloudFront in Route 53 and access with domain name.**
16. **Create record in private hosted Zone also (optional –do it by own way)**
17. **Add your domain name into AWS certificate manger and now access https protocol with your domain name.**

**AWS Lambda**

1. **Create Lambda function with default options.**
2. **Change configurations like time (setpoint to 15minutes), memory and role.**
3. **Run the lambda and check monitoring like logs, triggers, errors, invocations etc.**
4. **Add trigger as s3 buckets once file uploaded into s3 bucket lambda should trigger.**
5. **Add Destination as SNS once lambda task done mail will trigger.**
6. **Using boto3 create s3 bucket using lambda function.**
7. **Using Boto3 create Ec2 instance using lambda function.**

**AWS RDS**

1. **Create RDS MySQL cluster free tier and connect through SQL work bench create few tables.**
2. **After creating read replica other region check weather created tables there or not**
3. **Check all options in “Actions” after that deleted cluster and take a snapshot.**
4. **Create cluster from snapshot and check data is there or not by using SQL work bench**
5. **Create Ubuntu Linux server and install MySQL and create database and tables.**
6. **Compare MySQL on ec2 vs RDS MySQL after both labs.**
7. **Create Elastic cache for RDS and check for all options.**
8. **Create Aurora DB and observe the difference AWS provided DB and AWS manged DB. (chargeable but same process as MySQL)**
9. **note:1 all DBS creation same steps as per the requirement we can select the cluster.**
10. **Configure SNS notification to when MySQL cluster will delete.**
11. **Connect MySQL RDS from instances and check the tables and DB details.**
12. **Load the data tables from s3 to RDS by using s3 upload option.**

**AWS Elastic Beanstalk**

1. **Create sample application by using Elastic beanstalk and access by EBS URL check it**
2. **Upload sample application into Elastic beanstalk and enable high availability for that application and access it.**

**AWS Glue**

1. **Create simple glue job and check all options.**
2. **Create glue job to move data from one s3 bucket to other s3 bucket.**
3. **Create the crawler to load the schema from s3 to data catalog**

**AWS Redshift**

1. **Create RedShift cluster with free tier and check all options.**
2. **Redshift offers query editor so create tables and data bases directly, no need to connect cluster using any third-party tool like SQL work bench.**

**AWS CLI**

1. **Install and Configure AWS CLI**
2. **Create Ec2 instance and start, stop, and terminate by using CLI.**
3. **Create S3 bucket using CLI.**
4. **Perform the operations like list buckets, objects, delete etc.**
5. **Copy file from Ec2 to S3 and S3 to Ec2**
6. **Copy file from local-Ec2-S3 and vice versa**
7. **Create other multiple services by using CLI like, IAM, RDS, lambda**

**AWS Python BOTO3**

1. **Install Python into local machine and configure CLI also install boto3.**
2. **Write a script to create a S3 bucket by using boto3 document and run a script and check weather bucket is created or not.**
3. **Write a script to create a Ec2 instance by using boto3 document and run a script and check weather instance is created or not.**
4. **Also perform multi operation like start, stop and terminate.**
5. **Same task perform in AWS ec2, install python and boto3 in ec2 and run a script to create s3 bucket and instance from Ec2.**
6. **Perform same task in lambda using python boto3 create s3 and ec2 instances.**
7. **finally observe the changes same script runs in Laptop vs Ec2 vs Lambda.**

**Terraform:**

1. **Create S3 bucket using Terraform script.**
2. **Create other resources also like Ec2, Lambda, RDS etc.**

**SNS**

**186. Create SNS topic and check all options.**

**187. Subscribe your email and confirm it**

**SQS**

**188. Create SQS and check all options.**

**189. Configure SQS with AWS lambda.**

**190. Check send and receive messages with SQS queue**