IAM Module – Tasks

Task : create an IAM user, Add "S3FullAccess", login as this user and create a bucket. Now navigate to IAM service and try to create an IAM user and verify.!

Task : Create an IAM user, Add "AdministratorAccess" policy. Login and try to access "Billing information". It will Deny. --> Provide Billing access to the above IAM user.??

Task : create an IAM user, Add "EC2FullAccess", login as this user and try to create a bucket..

Task : Create a policy to 'deny' s3.

Task : Create an IAM user, provide him "AdministratorAccess" and associate "task 1 policy" to this user, login and try to create a bucket.!!!

Task : Create a Policy to allow S3FullACcess, but this should work from only one/your Network. You have to add an IP address condition.

Task : Create a custom policy to activate MFA for IAM users, He himself able to activate MFA.

Task :Create an IAM user, provide S3FullAccess. Login as this user and Now try to "Activate MFA". It will give error. Fix this errors and allow him to activate MFA.

Task : Create an IAM user, Want to provide "S3FullAccess" with our custom policy, We need to restrict this user to work only on specific region (mumbai/ap-south-1).

Task : Create a policy allow all s3 actions on all resources, Add a "region" condition. Login as this user, Try to create bucket in "mumbai" it should allow. If you choose any other regions it should not allow.

S3 Module - TASKS

Task : Create an s3 bucket, Upload an object, Access the Object as an IAM user.Make it public and access same object using public URL.

Task : Configure life cycle management rule to transit data from S3Standard to "Delete" after 2 days.

Task : Download a template from Internet. Deliver the webiste using s3 - static webiste feature.

Task : Create a bucket in N.Virginia, Configure Versioning and test it.

Task : Create another bucker in N Virginia region only.. configure same region replication between these buckets.

Task : Configure to receive alerts to your email id when any Object deleted from an S3 bucket.

Task : Enable Encryption on s3 bucket and test it. (SSE-S3/AWS Managed key KMS-DMK)

Task : Create an IAM user with "S3FullAccess". We need to restrict/DENY this user to perform "DeleteObject" opertion from a specific bucket (bucketname)

Task : create a bucket policy, that allow "one spcific IAM USer", to perform anything on this bucket. Remaining all users should not able to perform anything on this bucket.

Task : Create a bucket, Allocate this bucket to only one specific IAM user. Allow him to perform "GetObject", "PutObject" and "DeleteObject". This User should not able to view all the bucket we have in s3 platform.

Task : Make a bucket public for eveyone using "BUCKET POLICY".

EC2 Module – Tasks

Task : Launch windows ec2 instance and using keypair get connect to ec2 instance.

Task : Change the password of "Administrator", Disconnect from the instance. Now try to login to ec2 instance using "keypair pwd", "Custom password".

Task : With in this ec2 instance "Create a new user" in ec2 instance and provide him "Local administrator rights", also provide him "Remote desktop permissions"..

Task : Once Task 3 completed, Try to Take a session with task3 user.

Task : Launch an amazon linux 2, make it as webserver and deliver custom web content. (httpd-apache)

Task : Once task 1, completed. Change apache configuration port from 80 to 8888/8080 and Delive same webpage on 8888/8080 port.

Task 3: Launch an Amazon Linux 2/2023, install nginx and deliver custom web page.

Task :

-> Launch an ec2 instance,

-> Add a 2 gb additional volume, associate it with ec2 instance..

-> Make 2 gb volume available at os level and write some data into it. (Reboot and verify)

-> Reboot instance and enter "df -Th",

-> Now also it should show 2gb volume with data.

Task : Launch another ec2 instance in same AZ as existing ec2 instance.. Detach the volume from instance 1 and attach it to newly launched ec2 instance. Mount it to new directory. You should be able to see all the data.

Task : Launch an ec2 instance in ap-south-1a.. Create a "New volume with 1 gb size" in ap-south-1a and attach this volume to ec2 instance and make it available and write some data. Goto AWS console "Increase the volume size to 2 GB", make this 2gb volume available at OS level. Write some data.

Task : Launch Another ec2 instance in ap-south-1b..

Get the 2gb volume data from instance 1a to newly launch 1b ec2 instance.. Make Same volume available to "1b instance"..!!

Task : Launch Linux ec2 instance, Make it as webserver.. Add custom webpages.. Add a 2gb volume while launching and make it perm mount.. then Create a Golden AMI and Launch instance from GAMI and test output.

Task : Launch a Windows ec2 instance, Connect to it.. CHANGE ADMINISTRATOR PASSWORD, change wallpaper, install "putty" software, Install IIS and deliver custom webpage, Change timezone also..

Now, STOP the INSTANCE and create a GoldenAMi.

-> Launch an instance from GAMI, connect and Verify you got all custom settings or not.???

Task : How to setup password authentication to ec2-user user instead of keypair authentication.

Task : Launch an ec2 instance, while launching pass userdata and deliver webpage.

Task : Create an alarm, when Instance CPU usage is less than 30% for 10 minutes, Stop the instance and get an alart to your email.Now, create an Alarm to monitor your ec2 instance cpu, when cpu usage is 80% or more for 5 min, Reboot your ec2 instance and get an alert to your email.

Task : Create an ALB, and deliver webpage on One TG / Port :80. Test Stickyness.

Launch another ec2 instance and configure apache to run on port 8080.

Task : on existing LB, Add listeners on 8080 and test output on 8080.

Open port 8080 at your ec2 SG and ELb SG.

Task : what is Telnet and how to use telnet, test it with an ec2 instance on port 22.

telnet to google on port 443.

telnet to your ec2 instance on port 22 and test it.

Task : Configure and deliver NLB.

Task : Make sure you implement Pipeline mechanism.

Task : Create an ASG as we discussed (AMI, Load Balancer, launch template and Asg)

Task : launch an ec2 instance, install stress package and add load on your instance, Create an alarm to Automatically stop your ec2 instance when load is more than 70% for 5 min period.

Task : Create an updated version in "launch template" and test the blue and green deployment method.

Task : Create alarms on ASG. Whenever CPU Usage is <= 20% for 5 min, Set your desired capacity to 1.

When CPU usage is More than >=70% for 5 min, Set your desired capacity to 3.

Add load on your ec2 instance using stress package.

Task : Try Step scaling policy using shared script.

Task : Configure Schedule scaling policy.

Task :

1. Launch an ec2 instance, connect to it,

2. Create EFS..

3. Install httpd in your ec2 instance..

4. Mount EFS to /var/www/html/ and create index.html...

5. make it as perm mount. get EFS entry from /etc/mtab and write to to /etc/fstab

6. Test your ec2 instance webpage.

7. Stop above ec2 instance and create a goldenAMI.

8. launch a new ec2 instance from GoldemAMi, test your webpage. (**)

Then Process to below steps

==> Now create a GOLDENAMI.. or use Step 7 GAMI..

==> Create a Load Balancer

==> Configure ASG with this GOLDENAMI.. Connect to any of the ec2 instance, modify the content.. this modification should reflect in all instance..

Task : Launch an ec2 instance using CLI. (ami id, instance-type, subnet-id, security group id, keypair, count) (Example 4: To launch an instance and add tags on creation)

Task : Stop that ec2 instance using CLI comamnds.

Task : Start the same ec2 instance using cli.

Task : Launch an ec2 instance and Configure a role to access s3 data. (don't configure accesskey/secretkey.. use roles)

Task :

1. Create a sample webpages (index.html & status.html) and Upload a webtemplates to an s3 bucket..

2. launch an ec2 instance, while launching the instance, make your instance as webserver using userdata.

3. While using userdata, copy all the webcontent from s3 bucket to DocumentRoot path (/var/www/html/). Perform using "userdata".

4. make sure you attach an IAM role to your ec2 instance that can provide s3 access.

Task : How to recover a Windows instance administraor password if we loss the keypair.

AWSSupport-RunEC2RescueForWindowsTool

Task : Make linux ec2 instance as webserver without logging to OS level.

Task : Install tree package in linux instances without logging into.

Task : Configure Eventbridge to stop an ec2 instance at a particular time using Crontab. (9:15 Am)

Task : Configure to stop an ec2 instance after 5 minutes, trigger an email with custom email format.

Task 3 : Configure to get an alert when someone deleted an s3 bucket. (Enable Data trail)

VPC Module - Tasks

Task : Create a VPC with 2 Subnets. (1 Public SUbnets and 1 Private Subnets)

1 Public SUbnets (ap-south-1a)

1 Private SUbnets (ap-south-1b)

Task : Create a VPC with 4 Subnets. (2 Public SUbnets and 2 Private Subnets)

2 Public SUbnets (ap-south-1a, ap-south-1b)

2 Private SUbnets (ap-south-1a, ap-south-1b)

Task : create a CustomVPC, Launch a JUMP Server in CustomVPC's PUBLIC Subnet (Use WINDOWS AMI), Now launch another EC2 instance (Amazon Linux 2) in CUSTOM VPC's PRIVATE SUBNET. Connect to Linux instance from Jump Server.

Task : create a CustomVPC, Launch a JUMP Server in CustomVPC's PUBLIC Subnet (Use LINUX AMI), Now launch another EC2 instance (Amazon Linux 2) in CUSTOM VPC's PRIVATE SUBNET. Connect to Linux instance from Jump Server.

RDS - Tasks

Task : Launch a Mysql RDS db instance in CUstomVPCs Private SUbnet. Launch a Windows ec2 instance in custom VPC Public Subnet and Install Mysql workbench then connect to the RDS db instance.

Task : Launch a Linux ec2 instance in custom VPC Public Subnet and Install Mysql client then connect to the RDS db instance.

Task : Launch a POSTGRE SQL RDS db instance in CUstomVPCs Private SUbnet. Launch an ec2 instance in custom VPC Public Subnet and Install "PGAdmin" then connect to the RDS db instance.

Task : Launch a MS SQL RDS db (Express) instance in CUstomVPCs Private SUbnet. Launch an ec2 instance in custom VPC Public Subnet and Install "SSMS (SQL Server management Studio)" then connect to the RDS db instance.

Task : Launch a mysql rds db in custom vpc private subnet.

Task : Launch an ec2 instance in "customVPC" public subnet. Make sure your rds and public subnet instance are comm,unicating each other.

Task : Install "mysql workbench" in YOUR LOCAL LAPTOP. From your local laptop, using workbench and jump/public subnet instance, connect to RDS Database.

Route 53 Module - Tasks

Task : In Sandbox, Create a private HostedZone with your own domain name i.e; avinash.com

Launch an ec2 instance, make it as webserver and test the output.

Task : In Route53, Map your ec2 instance IP to Domain name.

LAMBDA Module- Tasks

Task : Stop ec2 instance when file uploaded into s3 bucket

Task : What ever the file names starting with "A/a" should move to another s3 bucket

Task : What ever the file names ending with ".bkp" should move to another s3 bucket

Task : Create a lambda function to perform telnet test. get output as "Telnet success" or "telnet failed".

Task : Get the list of IAM users whoever not logged in to aws account in last 1 hr.Once above test completed, move this with in VPC and test again.

SECURITY SERVICES -Tasks

Task : generate SSL certificvate, apply it to Load Balancer and deliver via Route 53. Add rewrite rules.

Task : Create an ELB, Deliver your ELB hosted website with Cloudfront distribution.

Task : Create an S3 bucket, Place some data and deliver the data using Cloudfront distribution.

Task : Launch an ec2 instance in Mumbai region, configure web server.. Migrate this instance to "mumbai" using MGN Service