https://www.udemy.com/courses/search/?q=aws%20certified%20solutions%20architect&src=sac&kw=aws

kanhu364@gmail.com

welcome1

https://learn.oracle.com/education/downloads/OracleCloudInfrastructurestudyguide.pdf

Sign In details to aws.amazon.com

Root user (having god mode all privileges)::

anigulerais@gmail.com

123@Anilguleria

When MFA (multi factor authentication) will be enabled it will require bar code as well. Download google authenticator from playstore.

A region is a physical location in the world which combines two or more availability zones.e.g London region you have a Sydney region Singapore region

Availability zone is one or more discrete data centers each with redundant power networking connectivity housed in separate facilities.

Edge location is just an endpoint for AWS which is used for caching content.Typically this consists of cloud front which is Amazon's content delivery network.

IAM

---

https://agcloud-2019.signin.aws.amazon.com/console --signin link to be sent to user. customize it.

kanhu364@gmail.com/welcome1

User

Groups

Policy

Roles.

enable MFA(multifactor authentication (username,password,barcodes) for root account. root account is one which we use to create aws account.)

Create user :: Anil.Guleria/123@Anilguleria

**Index ::**

10. Setting up a billing alarm

11. S3 :: simple storage service

12. Create s3 bucket

13. Security and encryption

14. Version control -- select bucket/oject -properties-enable version control

15. Lifecycle management- select bucket - managemnt - add lifecycle rule -transition of current/previosus version and expiration and deletion of objects.

16. Cross region replication - bucket- mgt- add rule - whole bukcet -- create new bucket - select destination region

17.S3- transfer acceleration - utilises cloud front edge network to accelerate your uploads into S3.

18. Cloud front overview -- its not read only but also we can write. Collection of edge locations is called distribution(CDN)

19. Cloud front lab: s- network and content delivery - cloudfront -- create distribution

20.Snowball overveiw :: Peta byte scale data transport solution. can import and export to and from s3

21. Snowball lab -- order a snowball from amazon and copy data to it -- services -- migration and transfers -- snowball

22. Storage Gateway :: a service that connects an on-premise software appliance with cloud based storage like s3.

24. EC2 101 Elastic compute cloud:: a webservice that provides resizable compute capacity in cloud

25. Launch our first EC2 instance and install webserver on it and accessing server thorugh putty :: services -- compute --ec2 -- launch instance

26. Launch our first EC2 instance :: ON EBS backed instance, root ebs volume also deleted on instance termination, and root volume cann't be encryptd directly.

27. Security group basics ( a virtual firewall) :: all inbound traffic blocked and outbound allowd bydefault. An ec2 instnace has multiple SG and vice versa. ONly for allow rules not deny.

28 EBS 101 (elastic block store) :: a virtual hard disk in cloud to run applications and databases (SSD AND HDD)

29. Volumes and snapshots :: volume type can be converted from one type to another, we can migrate an ec2 instance to different region -- ec2-create snaphost-create ami image -select ami-action - launch instance in different region.

30. AMI types ::Storage for root device is of two types::EBS and instance store (ephemeral—short living incase of instance crash/terminate it will be deleted only in reboot it will persist)

31. Encrypted root device volume and snapshots ::create a snaphots of root volume-encrypt snapshot- create ami image from encyrpted snapshot - from ami image create a new ec2 instnace with encrypted root volume.

32. Cloud watch :: a monitoring tool to monitor performance and alarm can be created to trigger notication. While cloud trail is for auditing like cctv. while creating ec2 instance select detailed monitoring to get alert in every 1 minute else its 5 minutes bydefault.

33. Cloud watch lab ::dashboards,alarms,events,logs :: Services – cloud watch – alarm – create – select metric like cpu utilisation – ec2 – per instance - select server.

34. AWS command line :: Iam – create user only give programmatic access, we need to run aws configure in command line and give secret keyid,access key,username than we can run "aws s3 ls"

35. Using iam roles with ec2 :: 34. uses password hard coded so we use iam roles instead. roles are secure and universal can be used in any region.

Create a role – iam – role- create role and then ec2 instance – action – instance setting – attach/replace iam role – select your role.

36. Using bootstrap scripts :: in advance section you can give bootstrap script

37. EC2 instance metadata :: http://169.254.169.254/latest/meta-data // local-ipv4,meta-data/profile or user-data to get history of commands run by user.

38. EFS2 :: Elastic file system :: is a shared storage multiple instances can use, scalable upto petabytes,support thousands of nfs connections, and data stroed in multiple AZ's within a region.

39. Placement group :: a group of instances. types - Cluster and spread placement groups. A spread placement group can span multiple AZ while cluster cann't.

41. :: Databases :: RDS(relation database system). Multi az disaster recovery and read replicas(read replicas used to enhance performance and can be multi AZ or region)

42.:: Lets create an RDS instance :: service -- database - rds - create - select mysql and create an ec2 instance and install wordpress and make it connect to database (in database's secruity group we add an inbound rule and add id of webserver's security group to this rule so that it can access database), rds runs on virtual machine we can't login. amazon do patching iteself.

43::RDS Backups,Multi-AZ and read replicas :: retention 1 to 35 days and stored in s3. DB snapshot is manual bkp, it can resotre db even if db is no longer there.

44::RDS Backups,Multi-AZ and read replicas Lab :: enable backup,convert db to multi az, create read replica, covert a read replica primary db node

45. Dynomo DB :: is fast flexible nosql db service for application which needs consistent, single digit millisecond latency at any scale. Stoed on ssD.

46. Redshift ::DAtawarehousing solution used for business intelligence.ONly available in one AZ. Asynchronously replicate your snapshots to S3 in another region for disaster recovery.

47. Aurora :: Amazon aurora is a mysql-compatible, relational database engine, how to convert mysql to aurora db.

48. Elastic Cache :: a webservice that makes easy to deploy scale in-memory cache in the cloud to increase database and web application performance. types ::reddis and memcahe(cann't be backed up)

49-60 .. route 53.

49. DNS :: DNS IS human friendly name for a ip address.

50. Register a domain - SERVICES - NEWTORING AND CONTENT DELIVERY - ROUTE53 - SELECT GET STARTED - NOW UNDER DOMAIN REGISTRATION -- REGISTER DOMAIN

51. Creating a routing policy lab -- create 3 ec2 webserver instances. services -rout53 -hotsted zones -slect your dns- create record set-name leave blank and value -give 3 ec2 IPs.

52-60 routing policy type ::Simple, weighted,latency,failover,geo location,geoproximity,multivalue routing policies.

61-62 :: VPC Features :: igw,routing tables,Nacl,security groups,subnets(public/private contains ec2 instances)

63 ::Create VPC :: services - networking and context delivery - vpc

create vpc, create two subnets(public/private),create igw and attach to vpc,create route table,edit route-add route-select igw,assocaite subnet-assocaite public subnet.

64. Network address translation instances and NAT GATEWAY LABs:: NAT instance is an ec2 instance in public subnet which acts a internet gateway for ec2 instances in private subnet, not used now.

NAT gateway :: Used for same purgpose but this is not an instance but its gateway in public subnet and 9 out 10 only gateways are used.

65 ::ACL ACCESS CONTROL LIST :: vpc-security-acl-create-add inbound/outbound rules. Assign acl to subnet. ONe subnet has only one ACL but not vice versa.

66. Custom VPCS AND ELS

67. VPC FLOW LOGS LAB:: TO CAPTURE ip traffic going to and from vpc. Allow to store log data in cloudwatch.

create log group in cloudwatch- assign loggroup to vpc- login and check logs in cloudwatch.

68. Bastions :: acts as a proxy server and are outside firewall.

69 Direct connect :: its a amazon cloud service solution to directly connects your data centre to AWS. fast,secure,high throughput better than vpn.

70 VPC endpoint Lab:: Allows to access s3 bucket/dynodb from private subnet ec2 instances without using NAT instance/gateway.

72. Load Balancers Theory :: Application (http/https layer 7),network(tcp traffic layer 4) and classic (old traditional)

73. Load Balancers Lab ::Classic and application

74. Advanced Load balancer theory :: Sticky sessions,cross zone load balancing and path patterns.

75. Autoscaling group lab :: allows us to configure a cluster of sever which can be autoscaled based on load metrics -- EC2 -- IN left panel -- select autoscaling -- launch configuration.

76. HA architecture :: Use mutiple AZ's and multiple regions, read replicas,scaleout (adding more intances) scale up(adding more resourecs to existing instances)

77. HA Word press site -- builing Fault tollerant wordpress site

78-79 Setting up EC2 and adding resilience and autoscaling Lab :: a big lab pratice it up.

80. Clean up :: remove ec2 instance - target group - load balancer cloud distribution --s3 buckets everything created in step 78-79.

81. cloud formation :: is a way of completely scripting your cloud environment. Quick start (on aws.amazon page) is a bunch of cloud formation templates already built by aws solutions architects allowing you to create complex environments very quickly

82. Elastic beanstalk ::you can create quickly deploy and manage applications in the aws cloud without worring about the infrastructure that runs those applications.

services -- compute -- elastic beanstalk-- get started .

84. SQS :: Distributed queue system that enables web service applicatoins to quickly and reliably queue messages that one component in the application generates to be consumed by another component.

types :: Standard (ordering is stict mostly but sometime not) and fifo(stirct order is followed)

85. SWF :: SIMPLE WORKFLOW SERVICE :: SWF is a web service that makes it easy to coordinate work/tasks across distirbuted application components.

SQS is message oriented api while swf is task oriented.

86. SNS -- Simple notification service :: is a web service that makes it easy to set up, operate,and send notifications from the cloud.

Both SNS AND SQS are messaging services but SNS is push based like we get billing alerts.

87. Elastic transcoder :: Its a media transcoder in the cloud convert media files from their original source format into different formats that will play on smartphones,tablets,PCs etc.

88. API Gateway :: API gateway are front door to your aws environment.

89. Kinesis :: Kinesis is a platform on aws to which streaming data is sent and there it can be analyzed. Types - streams(shards to store data),firehose (data need to analyze as it comes) and Analytices(on the fly analysis)

90. Web indentity federation - cognito :: Web identity federation lets you give your users access to aws resources after they have successfully authenticated with a web-based identity provider like amazon,facebook or google. Cognito acts as identity broker between your application and web id providers. User pools --hold user data. Identity pool -- authorise to aws resources.

91. Lambda Concepts :: is a compute service where you can upload your code and create a lamda function. Aws lambda takes care of provisioning and managing the servers that you use to run the code.

92. Lets build a serverless webpage using API and lamda :: Lab

94. Lets build an Alexa skill Lab

95. Important practical flow

10. Setting up a billing alarm

------------------------------

Login with root user :: select username drop down :: billing dashboard -- billing preferences -- tick mark free tier alerts and billing alerts -save.

services -- cloudwatch -- billing --create alarm -- scroll down - enter 2$ and mail\_id. Login to mail\_id and press confirm link. Now you can view your alarm.

11. S3 :: simple storage service

--------------------------------

object based allow to upload files from 0 to 5 TB, unlimited storage.

Files are stored in buckets.

https://s3-eu-west-1.amazonaws.com/ag -- bucket address.

s3 is universal namespace.

Can not be used to install operating system or db as its object based not block based.

Successfull upload to S3 gives HTTP 200 status code.

MFA delete can be turned on to prevent files from deleting.

key funadamentals of S3 are ::

key :: name of object

value :: simply data made up of sequence of bytes.

version id :: important for versioning.

metadata :: data about data which we store.

subresources:: Access control lists and torrent

read after write consistency for puts of new objects.

Eventual consistency for overwrite puts and deletes so need to wait a seocnd else we see old file.

Storage classess or tiers::

S3 standard

S3-ia (infrequenlty accessed.)

s3-one zone - IA

S3- Intelligent tiering

S3 Glaceir

S3 Glacier deep archive (Glaciers are used for data archving).

12. Create s3 bucket -- service - s3 - create bucket

13. Security and encryption

Security ::

ACL -- at object level

encryption ::

sse -- server side encryption

sse- ksm -- amazon and customer managed encryption

sse-c -- customer provided encryption

14. Version control -- select bucket or object -- properties -- enable version control.

15. Lifecycle management -- Automates moving objects between different storage tiers.

Can be used in conjunction with versioning and can be applied to current and previous versions.

select bucket - management -- add lifecycle rule - name -- select trainsing for current and previous version and expiration of ojbects and permanent deletion of old versions.

16. Cross region replication

----------------------------

versioning must be enabled on both source and destination buckets.

Regions must be unique.

Existing files in a bucket are not replicated automatically to destination bucket in different region.

Subsequent updated files will be replicated automatically.

Delete markers are not replicated

Deleting individual versions or delete markers will not be replicated.

//Any file we delete will not be replicated to cross region replication bucket.

Select a bucket -- mgt -- replication -- add rule -- whole bucket -- create new bucket-- name it -- select destination region -- select create new rule -rulename -- save.

17. S3- transfer acceleration :

-----------------------------

it utilises cloud front edge network to accelerate your uploads into S3 and instead of uploading directly to an S3 bucket you can use a distinct U.R.L. to upload directly to an edge location which will then transfer to s3 bucket.

18. Cloud front overview

------------------------

Edge location ::This is the location where content will be cached. This is separate to any AWS region or availability zone.

Origin :: this is the origin of all the files that the CDN(CLOUD DISTRIBUTION NETWORK) will distribute.

This will can either be an S3 bucket or it could be an EC2 instance or an elastic load balance or Route 53.

DISTRIBUTION ::distribution is the name that's given to the CDN which is a collection of edge locations.

Web distribution is typically used for Web sites

RTMP :: RTMP distirbution is used for media streaming.

Edge location are not just read only you can write i.e. put an objet to them.

Object are cached for the life of TTL (time to live)

you can clear cache objects but you will be charged.

19. Cloud front lab ::

--------------------

serivces -- network and content delivery - cloudfront -- create distribution --

web or rtmp (select web) -- get started -- select bucket -- keep default

scrrol down -- come to ttl -- you can give time to live min and max values -- create distribution.

It takes time 15 minutes and its a paid so don't create here.

once its done . copy domain\_name

Now in url write ::: domain\_name/untitled.jpg .. file will open.

Distribution invalidation can be set so that some files of all files in a bucket will not be distributed on edge locations.

20.Snowball overveiw ::

Peta byte scale data transport solution. can import and export to and from s3.

snowball -- 50 and 80 TB.

snowball edge -- 100 TB

snowmobile -- 100 Petabyte of data.

21. Snowball lab -- order a snowball from amazon and copy data to it and sent it to amazon.

services -- migration and transfers -- snowball -

**22. Storage Gateway ::**

AWS storage gateway is a service that connects an on-premise software appliance with cloud based storage to provide seamless secure integration between an ogranisation on-premise IT environment and AWS 's storage infrasturcture.

File gateway - for flat files, stored directly on S3.

Volume gateway - two types ::

Stored volumes :: Entire dataset is stored on site(on-premise) and is asynchronously backed up to s3.

Cached volume - entire dataset is stored on s3 and most frequently access data is cached on site.

Gateway virtual tape library :: durable, cost effective solution to archive data in aws could.

Section :: 4 EC2 Elastic compute cloud

--------------------------------------

24. EC2 101

-----------

EC2 is a webservice that provides resizable compute capacity in cloud.

EC2 Pricing types

On demand :: allows you to pay a fixed rate by hour or second with no commitment.

Reserved :: Provides you with a capacity reservation, offer a significant discount on hourly charge for an instance. Contract terms 1 or 3 years.

Spot :: enables you to bid whatever price you want for instance capacity.

Dedicated hosts :: Physical ec2 server dedicated for your use. It allows us to use existing server bound licenses like we have purchased license for oracle, sap etc. that we can continue with existing license.

25. Launch our first EC2 instance and install webserver on it ::

----------------------------------------------------------------

services -- compute --ec2 -- launch instance -- select which machine to run (window,redhat linux,suse ) -- amazon linux -- select t2 micro as its available in free tier -- next -- selet number of instances =1 and enable terminate protection -- next -- there will be root , you can add another volume but not needed now -- next

add tags --

name -- servername

department -- developers

empoyeeid -- emp\_id of person who provision it.

next:: configure security group (its a virtual firewall)

groupname/description -- webdmz

add rule -- http -- allow to all -- reveiw and launch -- generate a new key pair

download key to your local mahcine -- launch instance.

/\*Accessing server with chrome extension ::

ssh chrome extension -- secure shell extension -- add extension

chrome://apps -- type this in url -- you will get secure shell app.

cmd --cd to folder where keys are copied.

keygen -y -f key-name.pem > key\_name.pub

ren key-name.pem key-name

IN secure shell you can enter username hostname and key file name.

Using putty

-----------

:: putty doesn't recognise pem file so we convert it to ppk file using puttygen.

type puttygen in search button - in puttygen diaglog box -- load -- select your pem file(if its not visible select all files in dropdown) -- choose your pem file -- its imported successfully.

click save private file to same directory --name it converted\_mykey .

Now open putty ::

Hostname :: copy it from ec2 instance -- ec2-3-87-212-130.compute-1.amazonaws.com or 3.87.212.130

connect -data -- enter username -- ec2-user

ssh - auth -- browse -- slect your ppk file

Open

You can also save session. -- type a name and save -- open.

Next time you can direclty open form saved session name.

Install webserver ::

-------------------

Open putty session

sudo su // to change to root user.

yum update -y //it will install all latest os packages.

yum install httpd -y //install apache. it will convert ec2 instance to a webserver.

cd /var/www/html

vi index.html

<html><h1>Hello Anil to cloud</h1></html>

service httpd start

chkconfig on //this will start httpd on system reboot.

Paste 3.87.212.130 or hostname in browser you will see your website.

26. Launch our first EC2 instance

---------------------------------

we can check status checks, monitoring, tages,description and change server parameters, start stop.

On right side we can select to create different type of instances like spot , reserverd etc.

Termination protection is turned off by default.

on Elactic block storage (EBS) backed instance , root ebs volume also deleted on instance termination.

ebs root volumne of default ami's cannot be encrypted. Can be done using third party tool or apis.

additional volumes other than root can be encrypted.

27. Security group basics ( a virtual firewall)

-----------------------------------------------

All inbound traffic is blocked by default.

All outbound traffic is allowed.

Changes to security group take effect immediatly.

YOu can have any nubmer of EC2 instances in a security group.

YOu can have multiple security groups attached to EC2 instances.

security groups are stateful.

If you create an inbound rule allowing traffic in, that traffic is automatically allowd back out again.

Specific IP using security group can not be blocked, this can be done using network access control lists ( in vpc section)

YOu can specify allow rules, but not deny rules using security groups.

Lab :: EC2 -- select your isntance -- left side network and security -- security groups -- select a security group -- edit --now you can add or delete a rule.

//ly you can assign other security groups to your instance.

select instance -- action - networing -- change security group -- assgin other security group

28 : EBS 101

------------

Elastic block store :: its a virtual hard disk in cloud.

5 flaours ::

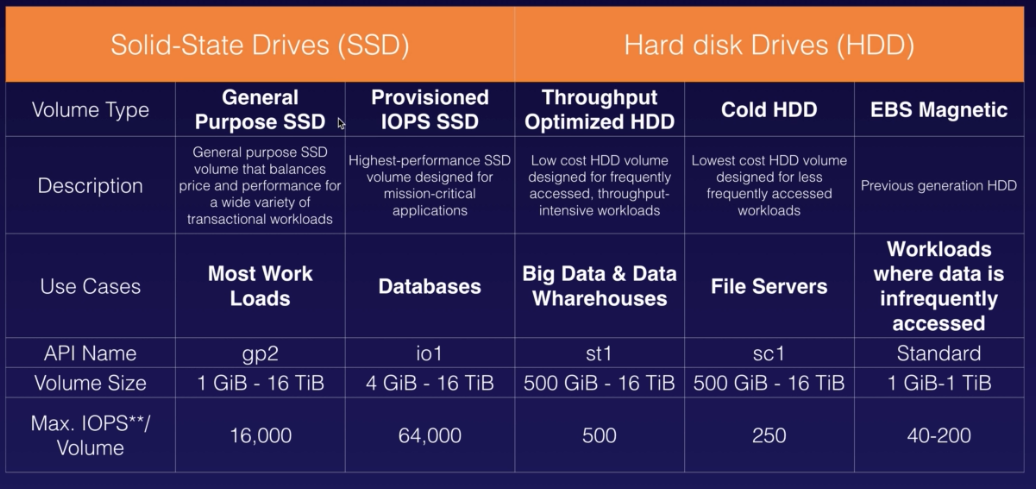
general purpose SSD

provisioned IOPS SSD

THOROUGHPUT OPTIMIZED HARD DISK DRIVE – HDD

cold hdd -- HDD

ebs magnetic HDD



29. Volumes and snapshots ::

You can assign or create an ec2 instance with different volume types. Additional Volume types can be converted from one type to another.

Volume (in left pan) – select desired volume – right click – change size or volume type – save – refresh.

One EC2 instance can be migrated to another AZ or region. For this we create a snapshot and than ami image. With Ami image we will configure new EC2 instance new AZ or region.

Select volume – create snapshot

Snapshot – create image

Under ami – you can see your image – select it and launch – now you can create a new ec2 instance in different availability zone by this.

If you migrate to different region ::

Ami – select ami – action -copy image – select new region to launch new ec2 instance.

Exam tips::

Volume resides on ebs, while snapshot on s3. Snapshots are point in time. They are incremental. First time snapshot creation takes time.

Its good to stop instance while taking snapshot, although we can take snapshot while instance is up.

Ami’s(image) can be taken from snapshot or volume directly.

Volume size can be changed on the fly as well as storage type can be changed too on the fly.

Voulme and EC2 instance should be in the same availability zone.

30. AMI types

Storage for root device is of two types :: EBS and instance store (ephemeral—short living)

EBS we did in lecture 29 till now.:

To create an ”instance store” AMI ::

EC2 – community AMI (in left panel) – instance store -- rest select same as ebs ami. Instance store volume can be added only time of AMI creation while EBS volume can be added later as well.

EBS ec2 instance can be stopped , reboot and terminate, but instance store ec2 instance can be only reboot or terminate we can’t stop them that’s why they are ephemeral. If instance store ec2 instance crashed or underlying host fails we will loose all data.

EBS instance can be stopped and we don’t loose data incase of failure or crash.

We can reboot both, by rebooting we won’t loose data.

On termination root volume will be deleted in both cases, however in EBS instance we can tell aws to to keep root volume.

31. Encrypted root device volume and snapshots ::

Steps ::

Create an ec2 instance.

Create a snapshot of unencrypted root device volume. Volume – select volume – create snapshot

Create a copy of snapshot and select encryption option. Sanpshot – select new snapshot – copy.

Create an ami (image) from encrypted snapshot. – select encrypted snapspot – action – create image.

Use that ami to launch new encrypted instance . Images – Ami – select your encrypted image- launch – rest all steps same to create a ec2 instance. Now new ec2 instance will have an encrypted root volume.

Exam tips :: snapshots and volume create from a encrypted snapshot are always encrypted.

Snahpshot can be shared only if they are not encrypted.

**32. Cloud watch**

It’s a monitoring tool to monitor performance.

User for monitoring performance, it monitor most of aws and application running on aws.

Cloud watch with ec2 monitor event every 5 minutes. Interval is 1 minute if detailed monitoring is turned on.

Cloud watch alarm can be created to trigger notifications

Cloud watch is all about performance monitoring while cloud trail is auditing remember a cctv like whenever we create an instance it will make a api call with amazon.

33. Cloud watch lab ::

Dashboards :: we can create awesome dashboards to see what’s happening with our aws environment.

Alarms :: allow to set alerts which will notify us when threshold limit are hit.

Events :: cloudwatch event helps you to respond to state changes in your aws resources.

Logs :: cloud watch log helps to aggregate, monitor, store logs.

Create an alarm ::

Create an ec2 instance – after launch – select detailed monitoring to get alert in every 1 minute- - rest all same.

Services – cloud watch – alarm – create – select metric – ec2 – per instance – select your server and select metric as CPUUtilisation – create alarm – name it , give 90% as threshold, 1 minutes out 1 or 2 out of 5. Notify me give email\_id or list.

34. AWS command line ::

Iam – create user only give programmatic access –

Login to putty –

aws s3 ls //it will give error

aws configure

enter secret access key id , access key , username (programatic user we created)

output format :: keep blank.

aws s3 ls //it will list all s3 bucket.

aws s3 mb s3:/testbucket\_2019

aws s3 ls

Access key and id is also stored in home directory

cd ~

ls //there will be no files but a hidden file will be there .aws

cd .aws

credentials file contains access key details.

Amazon didn’t recommend to hard code password like these so roles are there.

35. Using iam roles with ec2 ::

Roles are more secure than access key and secret keys on individual ec2 instance.

Roles are easier to manage, roles are universal you can use them in any region.

Create a role – iam – role- create role

Now ec2 – select your instance – action – instance setting – attach/replace iam role – select your role.

Now you can login to putty and run cli without need of access key. No need of .aws file you can delete it.

36. Using bootstrap scripts ::

Boot strap scripts can be used to run some commands at the time when instance boots.

EC2 – launch – IN advance section of first screen you can enters command as below and rest all steps same to create ec2 instance.  
  
#!/bin/bash

yum update -y

yum install httpd -y

service httpd start

chkconfig httpd on

cd /var/www/html

echo "<html><h1>Hello Cloud Gurus Welcome To My Webpage</h1></html>" > index.html

aws s3 mb s3://YOURBUCKETNAMEHERE

aws s3 cp index.html s3://YOURBUCKETNAMEHERE

37. EC2 instance metadata ::

metadata used to get information about an instance(such as public ip,local-ip,profile etc)

Login to putty ::

curl <http://169.254.169.254/latest/meta-data>

It will list all files to see data in file enter ::

curl <http://169.254.169.254/latest/meta-data/local-ipv4>

curl <http://169.254.169.254/latest/meta-data>/profile

to get user data ::

curl http://169.254.169.254/latest/user-data //it will show all user commands like history commands.

38. EFS2 :: Elastic file system

EFS provides a shared storage,means multiple instance can use one file system and its auto scalalble upto petabytes.

Supports network filesystem version 4 (NFSV4) protocol.

We will pay only storage we use (no pre-provisioning required)

can scale up to petabytes.

can support thousand of concurrent nfs connections.

Data is stored across multiple AZ's within a region.

Read after write consistency.

Steps ::

Create EFS filesystem first.

services - storage - efs - create file system -- select mount targets a 2-3 or more availabity zone in select region. -- next - enable encryption if needed - create file system.

Now create 2 ec2 instances ::

ec2 launch -- number of instances select 2 and in advance secion type below bootstrap script-- rest all same, select security group same as of efs file system.

#!/bin/bash

yum update -y

yum install httpd -y

service httpd start

chkconfig httpd on

yum install -y amazon-efs-utils

Servers are created with root volume as ebs.

Now we have to mount shared storage EFS we created with both instances.

To be done by command line :: First assign admin\_role else we have to give access\_key.

Now login to both instances using putty ::

public ip 54.166.49.1 , data-username- ec2-user and ssh-auth - browse key file.

On first server

cd /var/www/html / it will be empty, now we have to mount it shared efs storage.

cd ..

pwd

/var/www

ON aws console -- efs - selct your mount point and click blue link with mounting instruction ::

mount -t efs -o tls fs-bf8d755c:/ efs //replace efs with your directroy

mount -t efs -o tls fs-bf8d755c:/ /var/www/html //this is id of efs disk. crosscheck it before running command.

cd html

echo "<html><h1>Hello</h1></html>" >index.html

Now mount on second server

sudo su //to get in with root user

cd /var/www

mount -t efs -o tls fs-bf8d755c:/ /var/www/html

Now you can see this index.html here too.

Now open any ip of instance in browser it will show page.

**39. Placement group**

Two types ::

Cluster placement groups :: Group of instances within a single availability zone used when we need low latency and high network thorughput.

Spread placement group:: group of instances that are each on distinct underlying hardware. recommended for applications with small number of critical instances that shuould be kept seperate from each other to avoid risk.

Cluster placement group cann't span multiple availablity zones.

A spread placement group can span multiple AZ.

ACCOUNT name for placement group should be unique within our aws account.

Only certain types of instances can be launched in a placement group.(compute,optimized,gpu,memory optimized,storage optimized), aws recommend homogeneous instances in a placement group.

Placement group cann't be merged.

You cann't move exisiting instance to placement group. If we have to move we have to create AMI of existing instance and then luanch new instance from ami into placement group.

40. Summary

41. :: Databases ::

RDS :: relation database service

RDS OLTP supports below databases ::

SQL,ORACLE,MYSQL,POSTGRESQL,AURORA(AMAZON PROPRIETY),MARIADB

Key features ::

Multi az disaster recovery:: If database instance goes down connection will be redirected to another instance automatically.

Read replicas :: there can be 5 read replicas. It’s a read only copy of db, suppose we have a blogging site than most of the people are reading the blogs so read connections can be directed to read replica db’s instead of sending to primary node. IN this case if primary node goes down connection will be manually need to transferred.

Read replicas are created for performance enhancement. They can be multi AZ and a read replica can be created in a different region too.

Automatic backup should be on if we have to use read replicas.

DYNODB (NO SQL)

RDS OLAP :: REDSHIFT (AMAZON DATAWAREHOUSE) Used for business intelligence and ANALYTIC PROCESSING.

Elastic cache :: USED TO SPEED up performance of existing databases (frequently identical queries)

MEMCACHED

REDIS

42.:: Lets create an RDS instance ::

Steps ::

i.)Create mysql database instance.

ii)launch an EC2 instance

iii)add rule to rds security group so that our ec2 instance security group can access database.

iV)Login to ec2 instance and install wordpress and connect to db.

Detail steps ::

service -- database - rds - create - select mysql -only enable option for free tier -- create

databse instance class :: make it minium -- db2.t2.micro--1vcpu,1gb ram

Mulit-az - selct no

db instance identifier - db1

enter username/password

select default vpc

public accessible -- no

turn off backups - 0 days

monitoring -- disable

delection proteciton - disable

create database -- will take 5-10 minutes.

ON EC2 - launch an instance

advance sections enter bootstrap script ::

#!/bin/bash

yum update -y

yum install httpd php php-mysql -y

cd /var/www/html

wget https://wordpress.org/latest.tar.gz

tar -xzf latest.tar.gz

cp -r wordpress/\* /var/www/html/

rm -rf wordpress

rm -rf latest.tar.gz

chmod -R 755 wp-content

chown -R apache:apache wp-content

service httpd start

chkconfig httpd on

Name it wordpress server.

Now EC2 instance cann't talk to rds instance. So we have to enable this by following steps ::

rds-launch-wizard security group is created automatically when db is created. We have to select this security group and add rule to it to acces port 3306 for security group of our ec2 instance.

Security group -- select rds-launch-wizard - inbound -- edit -- add rule -- my sqlserver -- add group\_id for security group used by ec2 wordpress instance in source tab -save

Login to ec2 wordpress instance ::

copy public ip and paste in browser ::

it will show wordpress screen -- click lets go

enter db name, username,password,db host -- db host you will get from -- service - rds - db -click your db instance -- copy value under endpoint and port section, it will be as below ::

agclouddb.cek2ab6prwxd.us-east-1.rds.amazonaws.com

Prefix -- you can keep default or change.

next -- it will prompt to create wp-config.php file. copy context from dialog box and create wp-config.php file in /var/www/html and paste content in that.

Come back to browser -- press run installation on same dialog box -- enter site name , username,password , email\_id -- install wordpress.

Press login -- enter username/password -- now you can write data to our rds instance.

Exam tips ::

RDS runs on virtual mahcine

you cannot login to these operating system however.

Patching of rds operating system and db is amazon responsibility.

rds is not serverless.

Aurora serverless is serverless.

43::RDS Backups,Multi-AZ and read replicas

Automated backups :: Retention period 1 to 35 days. Provided point in time recovery.

Backup data stored in S3, we get free s3 space equal to size of our db.

It takes a full db snapshot and transaction logs backup. Backup is restored to a new instnace with new dns endpoint.

DB Snapshot backup :: these are done by manually. Can even used to restore db even RDS instance has been deleted.

44::RDS Backups,Multi-AZ and read replicas Lab ::

to take db snapshot ::

database - rds - select your db instance -- actions - take snapshot (to take db snapshot)

To change your db to mutli-aZ

database - rds - select your db instance -- modify - multi-az deployment - yes(to turn it on) -- continure -- immediatly -- modify.

click on your db - configuration -- multi-az will be yes now.

Now if we will goto action - reboot - select reboot with failover -- intance will to go to different AZ.

Creating Read replica ::

Turn on automatic backup first.

Click your db - modify -- retention period - 5 days - modify

Now create replica ::

select your db - action - create read replica - select region if you want a diffent region or select a different AZ -- encryption you can enable if you want -- multi-az you can set true/false -- give it name -- create

Turn you read replica db instance to a primary instance ::

select your replica db - action - promote read replica.

Exam tips ::

Read replicas ::

Read replicas can be multi-AZ

used to increase performance.

Must have backups turned on.

Can be in diffent regions.

Can be aurora or mysql.

Can be promoted to master(primary db) thiw will break the replication.

MulitAz::

User for DR.

We can force failover from one AZ to another by rebooting RDS instance.

45. Dynomo DB

Dynomo DB is fast flexible nosql db service for application which needs consistent, single digit millisecond latency at any scale. Supports key value and document. Used in mobile, web,gaming ,ad-tech,IOT.

Sotred on SSD storage

Spread across 3 geographically distinct data centres

Eventual consistent reads (defualt, more than a second)

Storngly consistent reads. (Less than a second)

46. Redshift

DAtawarehousing solution used for business intelligence.

Only available in one AZ.

Backup ::

Enabled bydefault with 1 day retenction, can be extened to 35 days.

Redshift Always attempts to mantain at least three copies of your data(original and replica nodes on compute nodes and backup in S3.

Redshift can also asynchronously replicate your snapshots to S3 in another region for disaster recovery.

47. Aurora

Amazon aurora is a mysql-compatible, relational database engine.

Migrate Mysql db to Aurora::

select your db - action - create aurora read replica -- name db instance idnetifier -- check backup and Az/region parameter -- create

A aurora-cluster will be created. having write and read aurora dbs. If we want we can change reader node to master also.

Exam tips ::

2 copies of your data is contianed in each AZ, with minimum 3 AZ. 6 copies of our data.

We can take snapshots with aurora and share snapshots with other aws accounts.

2 types of replica available. Aurora replicas and mqsql. Automated failover is only available with aurora replicas.

Aurora has automated backups tunred on by default.

48. Elastic Cache ::

Elastic cache is a web service that makes it easy to deploy,operate,and scale in-memory cache in the cloud.It is used to increase database and web application performance.

Two types ::

Memcache and Reddis.

Reddis is multi-AZ, AND you can take backups and restores of reddis.

If you need to scale horizontally, and a quick solution use memcahed but it cann't be backed up.

**49-60 ROUTE 53**

**DNS ::**

DNS is used to convert human friendly domain names like http://acloud.guru to an internet protocol(IP) address such as http://82.345.56.56.

IP address are used by computers to identify each other on network.

.com -- its top level domain name.

.co.uk -- .co is second level domain name and uk is top level domain name.

Domain name are controlled by iana,its a database.

ELB (ELASTIC LOAD BALANCER) do not have pre-defined IPV4 addresses; you resolve them using DNS name.

alway choose alias record over a cname if choice is given.

DNS types :: SOA,NA,A,CNAMES,MX,PTR.

**REgister a domain name ::**

SERVICES - NEWTORING AND CONTENT DELIVERY - ROUTE53 - SELECT “GET STARTED” - NOW UNDER DOMAIN REGISTRATION -- REGISTER DOMAIN -

PICK A NAME -- hellocloudguru2019 give this name -- check (to check wether name is available or its used by some other website)

click add to cart (its paid will show the price)

continue -- enter registrant contact details -- accept terms - complete purchase -- it can take 1 hr to 3 days.

ONce its registered --

In left pan select registered pan -- if its registered it will appear.

Click hosted zones -- domain will appear

hellocloudguru2019.com

Now we will make routing policy ::

Types ::Simple, weighted,latency,failover,geo location,geoproximity,multivalue

Create 3 EC2 instances in three different regions, with below bootstrap scirpt and replace X with region name in bootstrap script.

#!/bin/bash

yum update -y

yum install httpd -y

chkconfig httpd on

service httpd start

cd /var/www/html

echo "<html><h1>Hello Cloud Gurus! This is the X Web Server</h1></html>" > index.html

Simple routing polciy ::

copy IP of all three ec2 instances.

services - rout53 - hotsted zones -- select your dns - create record set --

In right panel --

Name leave blank so it will show hellocloudguru2019.com

value :: paste all three IPs in sepreate rows.

routing policy -- select simple -- create.

Now type hellocloudguru2019.com in browser.

In all other routing policy below we can create health checks.

route53 - hosted zones - select your dns name -- create record set--

Now we will create 3 record for three EC2 instances and a health check for all 3 ec2 instances.

Name -- text box you can keep blank its a naked domain no www with it.

value -- enter IP of first EC2 instance.

routing policy -- weighted,latency,failover,geo location,geoproximity,multivalue

Create for all three EC2 instance.

In left panel -- click health check - create health check

name -- region of first ec2 instance like Sydeny

IP address -- enter IP address of ec2 instance of sydny region.

hostname -- hellocloudguru2019.com

port -- 80

Path -- index.html

advance configuration

interval -- you can set to fast (means every 10 seoncds)

failover threshold -- 2 , it means 2\*10 = 20 seconds if its down health check will fail.

click next - here you can configure alert if you want else create health check.

//ly create health check for all three EC2 INSTANCES.

Now assign health check to record created for all Ec2 instances.

Hosted zone(in left panel) - select frist record - in right panel at last -- associate with health check -- select health check for desired region and assign i.e. For sydny instance select health check of sydny.

network and migration- rout 53

Simple routing ::

In simple rooting policy you can only have one record with multiple IP

addresses and you can't have any health checks. If you specify multiple values in a records set route 53 will return all of the values to the users in a random order.

So here's our user typing in a DNS address.

Route 53 is giving them the first record and then it's giving them the second record but we can't have a health check on that.

Weighted routing policy ::

Weighted rooting policies we've got our user typing in a DNS address and we're

setting weights so we're sending 20 percent of the traffic to US EAST-1 and 80 percent of the traffic to US-WEST-1.

Latency Routing policy :: this policy is based on user location and latency. In latency routing there might be chances that a London based user might get a fast connection from Sweedon region rather than UK region, So in latency routing policy his connection will go to lowest latency that is Sweedon.

Failover routing policy ::

In this we've got an active passive environment our active environment could be US and our passive environment could be AP(Asia pacific), if for some reason our EC2 instances or a region goes down it's gonna detect this using a health check and it's gonna fail over to our passive environment.

Geo location routing ::Y

Geo location routing allows our European customers to be sent to our European servers and it allows US customers to be sent to our US servers.LIke if we have a bill payment than euros will be for europe and $ will be for US.

Geoproximity routing ::

it lets “amazon route 53” to route traffic to your resources based on geographic locations of your users and your resources. we can optionally choose to route more or less traffic to a given resource by sepcifying a value known as bias. To use this policy we must use route 53 traffic flow.

Its above the scope of cloud architect professional certification.

Multi value routing policies ::

this is the same as simple base rooting except you get health checks and so you can have multiple values within your records sets.

So with our record set in here and we've got our first and our second.

And if one fails a health check what's going to happen what's going to remove that from the record set.

Or it's gonna stop serving that record until it passes the health check.

61-63 :: VPC

VPC features ::

Launch instances into a subnet of your choice.

Assign custom IP address ranges in each subnet.

configure routE tables between subnets.

Create internet gateway and attach it our VPC.

MUCH BETTER security control over your aws resources

Instance security gorups

subnet network access control lists(ACL)

Default vpc vs custom vpc ::

defualt vpc is user friendly can, allowing you to immediately deploy instance by clicking launch vpc button.

All subnets in defaut vpc have a route out to the internet.

Each ec2 instance has both a public and private ip address.

Its for developers. It can be recovered if deleted but don't delete.

But in custom -- we create using create vpc button -- Only public subnet has access to interent private doesn't.

VPC peering :: Allows you to connect one VPC with another VPC via a direct network route using private ip address. No transitive peering.

Exam tips ::

Think of VPC as a logical datacenter in AWS.

cONSISTS OF IGWs(or Virtual private gateways), route tables,network access control lists,subnets and security gorups.

1 Subnet = 1 Availability Zone

Secuirty groups are stateful; NACL are stateless.

No transitive peering.

means if a can b and b can c it doesn't mean a can access c, we have to create a new rule so a can access to c.

When you create vpc a defualt route table,NACL and a default security group is created.

Subnet and IGW are not created by VPC.

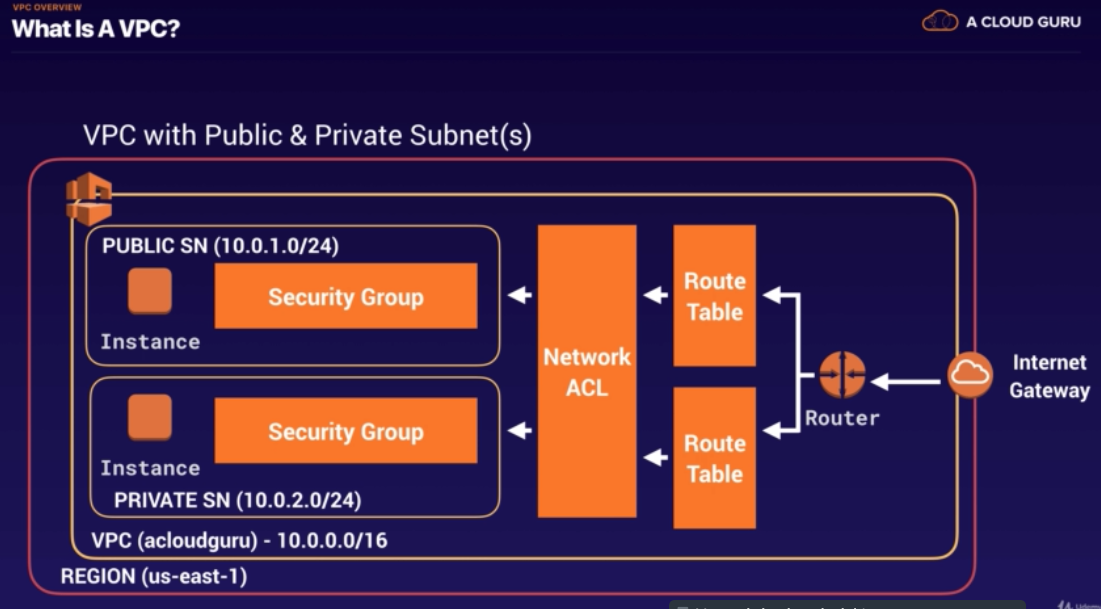
US-EAST-1A for one aws is different from US-EAST-1A for another aws account. They are randomized.

Amazon always reservers 5 IP addresses within your subnets.

YOu can only have 1 internet gateway per VPC.

SECURITY groups cann't span VPCs.

**Create VPC ::**



**Create VPC ::**

-------------

i)Create custom vpc

ii)create two subnets

iii)make one subnet public for web server which is to be accessed over internet.

iv)create IGW and attached to vpc

v)create a new route table for internet and add route for all ipv4 and 6 IPS for our IGW and associate it to public subnet.

vi) Now provision ec2 instances one for public subnet and another for private subnet in your vpc.

vii) How to access ec2 instances.

viii) To access db private subnet instance we have to create a security group which allow access to servers in public subnet.

ix) We cann’t patch private subnet as it has no access to internet so configure NAT instance/gateway.

x.) Configuring ACL for vpc

xi) collect vpc logs using cloudwatch (make cloudwatch group and assign to vpc.)

xii) vpc endpoint :: this enables access to objects outside vpc like “s3 buckets/dynamo db” without the use of NAT instance/gateway by ec2 instnaces in private subnet. Vpc -endpoint-create-s3gateway-vpc-policy-full – create.

**Detailed steps ::**

**i)Create custom vpc**

services - networking and context delivery - vpc--

left panel - your vpc - create vpc -- don't select launch vpc.

name -- cloudguruanilvpc

ipv4 CIDR block -- 10.0.0.0/16 //keep it 16 which is max.

IPV6 CIDR -- AMAZON PROVIDED IPV6 CIDR BLOCK

TENANCY - DEFAULT/DEDICATED (SELECT DEFAULT WHICH IS MULTITANENT MEANS SHARED BY OTHERS AS WELL)

CREATE

this vpc creation creates the below ::

route table -- it will create

network acl -- it will create

security group -- it will create

subnet -- it won't create

internet gateway - not created

**ii)create two subnets**

We will create two subnet

10.0.1.0 --Public subnet (for servers to be directly accessed from internet)

10.0.2.0 -- privat subnet ( for servers which are not connected to internet e.g. a db server which is to be accessed only by applicatoin server)

Publice subnet ::

in left panel - subnet - create subnet

name -- 10.0.1.0-us-east-2a (this is AZ we are going to select below)

vpc -- select cloudguruanilvpc

AZ - us-east-2a

IPV4 CIDR block - 10.0.1.0/24 //it shold be less than 16. Here 16 has maximum IPs and 28 is lowest.

IPV6 -- don't assign ipv6

create -- now we can see it.

Private subnet ::

Name 10.0.2.0-us-east-2b

vpc -- cloudguruanilvpc

aZ -- us-east-2b

ipv4 - 10.0.2.0/24

ipv6 - don't assign

create

**iii)make one subnet public**

NOw here auto-assign-public will be no for both subnets. But 10.0.1.0 we have to make publicly accessible.

So select it -- action - modify auto assign ip setting -- enable auto-assign - save

Now it will be changed to yes if we scroll to right side.

**iv)create IGW and attach to vpc**

click internet gateway in left panel - create IGW --

NAME - MYIGW -- save

Now its not attached to any vpc, so attach it.

select your IGW - action - attach to vpc - select your vpc -- attach.

Per VPC we can have only one IGW. We can't attached second IGW to a gateway.

v)create a new route table add route to igw and associate with public subnet::

create a new route table and add route for all ipv4 and ipv6 IPS for our own IGW and associate it two public subnet.

Main route table routes allows IPs within a VPC to communicate with each other and is created bydefault when vpc is created.

We cann't use main route table route out it for internet.

so create a new rout table for internet ::

routes - create route table --

name -- mypublicroute

vpc -- select our vpc

create - close.

now we have two route table

main - default created with vpc and second is mypublicroute.

Select mypublicroute -- to create a route out for internet

Press button “edit route” - add route

destination --0.0.0.0/0

target -- igw -- it will display our igw select it.

//ly for ipv6

::/0 and select our igw.

save

Now we created routes in our route table added route to our IGW and now we have to associate it with public subnet.

Now we have to do subnet association with these routes.Any subnet which associates to this route will be public.

select your route table just created -- subnet association -- edit subnet association -- select subnet 10.0.1.0 and save.

Now we have public subnet associated with our mypublicroute route table and default route table.

Private subnet associated with default route table only.

vi) Now provision ec2 instances

Now we have to provision ec2 instances for our VPC.

service - compute ec2 -- in third step confiure intance details -- select

network -- select our vpc name

subnet -- select public subnet

auto assign public ip -- already enable bydefault

Tag -- Name - webserver01

create a new security group if no security group in this vpc. Security group can't span across VPC

group name and desc -- webdmz

add rule ::

ssh 0.0.0.0/0 //it will be already there.

http to all. -- review and launch

keypair --create new key pair

download key pair.

launch instance

Launch second ec2 instance ::

network -- select our vpc name

subnet -- select private subnet

auto assign public ip -- already disable bydefault keep it as it is.

Tag -- Name - My-dbserver

security group -- select existing -- assign default security group not the one we created above.

key pair -- use same which is used in first instance.

**vi) how to access vpc instances**

Now go to putty.. import key and we can access ec2 instance in public subnet.

**vii) how to access private instance**

Now how to access private subnet instance ::

It will be accessed through our webserver , public instance.

Create a new security group ::

name my-dbsg

vpc -- select out vpc

add rules to it with source as 10.0.1.0/24 for all::

all icmp 10.0.1.0/24 // this is to test ping command.

http 10.0.1.0/24

https 10.0.1.0/24

mysql/aurora 10.0.1.0/24

ssh 10.0.1.0/24

create

Now select your db-server ec2 instance -- secruity group --

assign my-dbsg to it and remove default secuirty group.

By this security group all servers in subnet 10.0.1.0/24 can access db server on ports of icmp(ping command),http,https,mysql and ssh.

Copy it private ip.

ON public instance

ping paste ip // it will work.

Now inorder to ssh we have copy the key used by private instance and paste it in public instance.

Open downlaoded key file in text editor and copy its contents and paste it in file on public instance.

ON public instance ::

vi key\_file.pem

chmod 400 mykey.pem // if permissions are more than this than login fails.

ssh ec2-user@private\_ip -i key\_file.pem

**64. Network address translation (NAT instnaces and NAT gateway)**

::

Now your private subnet instance is not available to connect to internet. Supoose we have to apply updates, it will fail.

run yum update -y

Here comes NAT into picture.

NAT instance is a single ec2 instnace created in public subnet which acts as a internet gateway for instances in private subnet to access internet.

NAT gateway is a highly available gateway which allows private subnets instances communicate to internet without becoming public.

9 out of 10 times in real world we are using NAT gateway.

Exam tips ::

Nat instances are out of date but still comes in exam.

When creating a NAT instance, disable source/destination check on the instance.

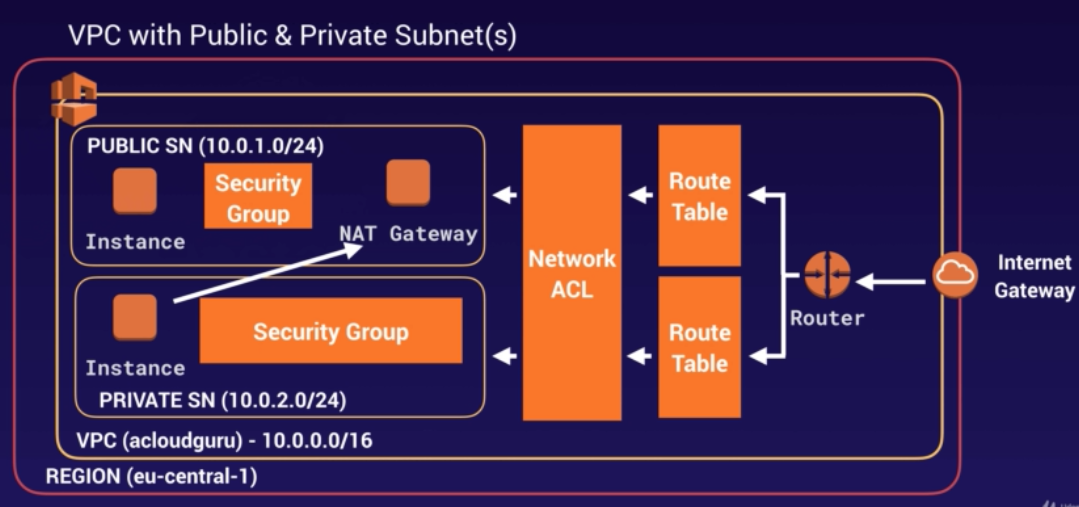
NAT instances must be in public subnet.

There must be a route out of private subnet to the NAT instance, in order for this to work.

The amount of traffic that NAT instances can support depends on the instance size. If you are bottlenecking increase instance size.

You can create high availability using autoscaling groups, multiple subnets in different AZs, and a script to automate failover.

NAT GATEWAYS ::



redundant inside the availability zone.

Preferred by the enterprises.

Starts a 5GBps and scales currently to 45gbps.

no need to patch.

Not associated with security groups.

Automatically assigned a public ip address.

remember to update your route tables.

No need to disable source/destination checks.

If there are multiple AZs sharing a NAT gateway and if the AZ which hosts NAT gateway gets down than all private instances in other AZ will also loose internet access. So configure multiple NAT gateway across multiple AZs.

Lab ::

NAT instance ::

Launch ec2 instance

step 1 - click community AMIs in left panel -- type NAT -- select first instance --

network - select your vpc

select it in public subnet

it will get a IP

SELECT a security group -- that webserver has

make a general purpose ssd -- select second option

next -- launch - select existing key --launch

As NAT instance doesn't have traiffic in or out for itself but it do this for a private instance so disable source/destination checks for NAT instance.

select your nat instance -- actions - networking - change source/desitnation check - disable

vpc - route tables - select main route tables(as all private subnet instnaces are associated with main route table) - add route

0.0.0.0/0 -- target -- select NAT instance.

Now login it your public instance -- ssh to private -- run

yum update -y //it will run.

Now terminate this instance as we have to configure our NAT gateway.

Also remove this entry from route table.

select route - main route - edit - select above route and delete.

NAT gateway ::

service - netwokring -- VPC -- SLEECT NAT GATEWAY IN LEFT PANEL - CREATE --

SUBNET -- slect your public subnet //it dispalys id only check subnet ID for public subnet and select that ID here.

elastic ip allocation id -- press create new ip -- create -- after this don't close -- select Edit route table link -- select your master route table for your VPC--route -add

0.0.0.0/0 -- target -- click NAT gateway -- select your NAT GATEWAY -- close. //it will take some time to setup.

Now go to private EC2 instance and you can install package or run yum udpate -y.

65 ::ACL ACCESS CONTROL LIST ::

Exam TIPS::

VPC automatically comes with an ACL and this ACL allows all inbound/outbound traffic.

We can create custom ACLs, and custom acl has all inbound/outbound traffic denied by default.

Each subnet in your VPC must associated with an ACL, by default its associated to default ACL, but we can change it explicitly to custom ACLS.

Individual IP address can be blocked using ACL not with security groups.

One ACL can be associated to multiple subnets but one subnet can be associated to one ACL only.

Network ACL contains a numbered list of rules which are evaluated in order starting with lowest number.

ACL are stateless and have seperate inbound and outbound rules not like security groups which are statefull.

ACL lab ::

vpc - security - acl - create acl -

name - myacl

vpc - select your vpc

EC2 instance - get ip of public instance – login to putty.

sudo su

yum install httpd -y

chkconfig httpd on

service httpd start

cd /var/www/html

vi index.html

<html><h1>Welcome to cloud</h><html>

Paste public webserver IP in browser it will work.

Now our public subnet is associated to default ACL which allows all traffic.

We will change our subnet to new ACL and assign allow/deny rules.

VPC -- ACL - select new acl - subnet associations -- select public subnet -- add it.

Now paste public IP in browser it wont' work as ACL denys all bydefault.

select your acl - inbound rules -- edit inbound rules

add rule : give rule no. and port.

100 -- port 80

200 -- 443 https port.

300 -- 22 (ssh port)

//ly outbound rules

100 -- 80

200 -- 443

300 -- 1024-65535 /ephemeral ports. (these are ports assigned temporarily to a process which was previously using main port like 80 or 443. Once process is done they are reassigned to another processs.

Now you can access your page again.

Now add a deny rule ::

ACL rules are accessed in chronological order means in number order. Here if I deny my IP in ACL than this rule number should come less than 100 as rule number 100 allow access to all so it will be read first.

Inbound - edit - add rule -- rule number -- 99 , port -- 80 ,sourc -- my\_ip/32(in goodle type what is my ip), allow/deny -- deny.

save.

Now check you wont' be able to access site.

66. Custom VPCS AND Elastic Beanstalk

ec2 - load balancer - create -- http -- IN detail in next chapeter.

Elastic Beanstalk -- atleast two public subnets are required to create Elastic Beanstalk.

Elastic Beanstalk cann't be created for private subnets as no IGW associated with them.

67. VPC FLOW LOGS ::

TO CAPTURE ip traffic going to and from vpc. Allow to store

log data in cloudwatch.

Flow logs can be created at three levels ::

vpc

subnet

network interface level.

Exam tips ::

You can't enable flow logs for VPCs that are peered with your VPC unless the peer VPC is in your account.

You can't tag a flow log.

After you have created a flow log, you can’t change its configuration: for example you can't associate a different IAM role with the flow log.

Below traffic is not monitored by flow logs ::

Traffic generated by instances when they contact amazon DNS SERVER. if we use own DNS than traffic is logged.

Traffic generated by windows instance for amazon windows license activation.

Traffic to and from 169.254.169.254 for instance metadata.

dhcp traffic.

Traffic to the reserved IP address for the default vpc router.

Lab ::

i) create a log group in cloud watch.

ii) assign log group to your vpc.

iii) login to http server and check cloud watch logs for log data.

i)service - mgt and goverence - cloud watch - logs --lets get started or actions - create log group -- name -- vpc-first-log-group --create log group.

ii)vpc - select custom vpc - action - create flow log

filter -- all

destination -- send to cloudwath logs

destination log gorup -- select the log group create above --vpc-first-log-group

IAM role -- as there is no role for this yet so click blue link "Set up Permissions" --a new tab will open --

You can keep all default that is role name as "flowlogsRole" -- allow.

Come back to old tab ::

refresh it

filter -- all

destination -- send to cloudwath logs

destination log gorup -- vpc-first-log-group

iam role -- select flwlogsRole

create

iii)Now try to access your website and check logs ::

cloudwatch - logs -- click you log .

68. Bastions ::

speical purpose computer on a network specifically designed and configured to withstand attacks. Host a single application like proxy server and is hardened in this manner primarily due to its location and purpose which is either outside firewall or in a demilitarized zone.

A bastion is used to securely administer EC2 instances(using ssh or rdp). Bastions are called jump boxes in Australia.

You cann't use a NAT gateway as a bastion hots.

A NAT gateway or NAT instance is used to provide internet traffic to EC2 INSTANCE in a private subnets.

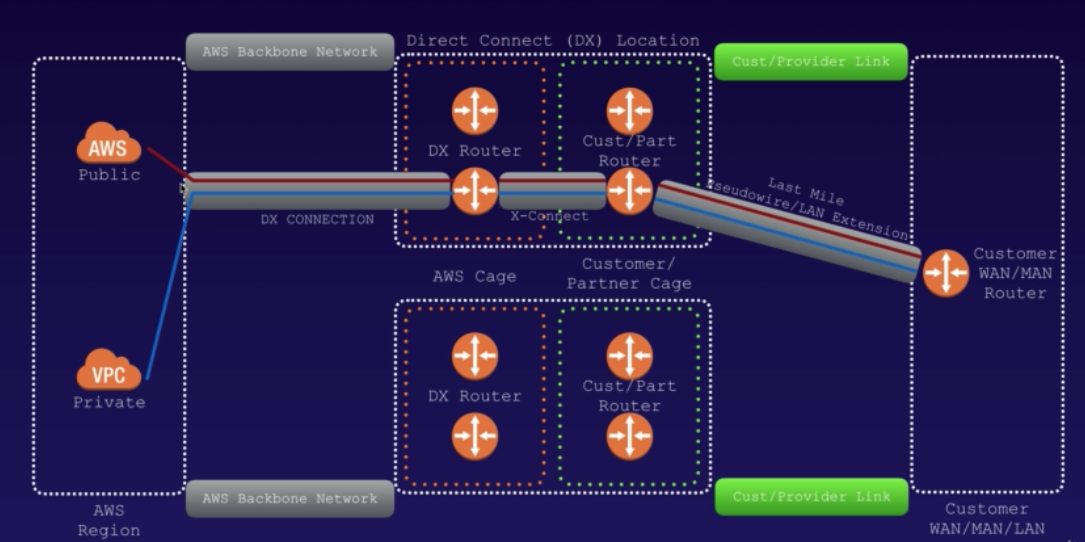
69 Direct connect ::

Direct connect is a amazon cloud service solution to directly connects your data centre to AWS.

iF YOU vpn CONNECTIONS drops out frequently use direct connect.

Useful for high throughput workloads(ie lots of network traffic)

Or used if you need a stable and reliable secure connection.



70 : VPC endpoint ::

VPC ENDPOINT enables you to privately connect to your vpc to supported AWS services and VPC endpoints services prwered by PrivateLink without requiring an IGW,NAT device,vpn connection or aws direct connection. Instances in your VPC DO NOT REUQIRE public IP addresses to communicate with resrouces in the service. Traffic between your VPC and other service does not leave the amazon network.

two types of VPC endpoints ::

Interface endpoints

Gateway endpoints.

Here we will do Gateway endpoints.

Interface endpoint is an elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported service.

Gateway endpoint are line IGW currently supported for amazon s3 dynamodb.

Lab ::

Currently if our private instnace has to download a package like httpd it has to connect to NAT gateway in public subnet which routes connection to outside internet.

Now with Gateway endpoint we will create VPC gateway in private subnet and our private instance will connect to vpc gateway and download/upload files, no need to go outside amazon network but this is for only S3 and dyanamodb for rest internet we will use the same NAT gateway.

services - IAM - CREATE ROLE - EC2 SERVICE CHOOSE IT – NEXT PERMISSIONS -- SEARCH -- type s3 -- SELECT AMAZON S3 FULL ACCEESS.

NAME -- S3-ADMIN-ACCESS - CREATE ROLE.

Now assign this to your db server.

ec2 -- select your db server -- action -- attach iam role -- assign role created above --apply.

Now if you public subnet is assgined to custom acl you might be not able to access it using ssh , so change it to defautl ACL so there will be full access to it.

Now login it to public ip and ssh to private ip .

sudo su // to get into root user.

aws s3 ls

echo "test" >test.txt

copy it to s3

aws s3 cp test.txt s3://bucket\_name -- get bucket name from ls output.

Goto s3 -- you can see your file.

creating and using endpoint ::

Now delete your NAT GATEWAY or route for nat gateway in route table. ::

vpc -- routes tables -- select main route table for your vpc -- edit routs -- remove -- route associated to NAT GATEWAY TO ACCESS INTERNET.

GO BACK TO ec2 instance

aws s3 ls //it will fail as there is no route.

Now create endpoint ::

vpc - endpoint in left panel -- create endpoint -- go down

select service as amazon-s3 gateway as com.amazonaws.us-east-1.s3

vpc -- select vpc

rout table -- select master route table

policy -- full access

create end point

Now click route tables -- new route will be there but it will be visible after some time i.e. 4-5 minutes.

Once visible you can test ::

ON ec2 instance

aws s3 ls

it won't work becuase with endpoint we have to give region name.

aws s3 ls --region us-east-1 //region name you will get while you create endpoint and select service com.amazonaws.us-east-1.s3.

**8. HA Architecture**

**72. Load Balancers Theory**

**Load Balancer ::**

3 different types of load balnacer ::

Application load balancer -- intelligent load balancer works on layer 7 , used for http and https.

Network load balancer -- Network load balance are used for extreme tcp traffic works on layer 4.

Classic Load balancer -- These are traditional old load balancers, not intelligent simply use round robin.

504 error means the gateway has timed out. This means that the application not responding within the idle timeout period. Troubleshott the application, is it web server or db server.

If you need the IPv4 address of your end user, look for the X-Forwarded-For header.

**73. Load balancers and health checks Lab :**

Exam tips ::

Instances monitored by ELB are reported as Inservice, or outservice

Health checks check the instance health by talking to it.

Load balancers have their own dns name. YOu are never given an IP address.

Read the ELB faq for classic load balancers.

**Classic load balancer ::**

i) create two ec2 intances and install webserver on them.

ii) creaete load balancer and add healthchecks and add ec2 instances.

iii) Check instances in load balancer - instances -- they should be in service.

iv) Now test load balancer by copying its dns name from description.

**Application load balancer ::**

i) Create target group ( elb send requests to a target in target group)

ii) Create application load balancer select target group we created.

iii) Select your target group - target -- select werservrs - add to register -- save.

iv) on server status is healthy on target page then go to load balnacer -- copy dns and paste in browser and check.

Load balancer is a paid service every time we use it so delete it after lab ::

Create two EC2 instances

vpc -- select defualt vpc

subnet -- select different subnets for both instances so there availability zones will be diffrent.

check -- auto assign public ip should be enabled.

advanced -- write bootstrap script as below ::

yum update -y

yum install httpd -y

chkconfig httpd on

service httpd start

cd /var/www/html

echo "<html><h1>Welcome to cloud webserver - 01</h><html>" > index.html

Change this line for webserver 2.

echo "<html><h1>Welcome to cloud webserver - 02</h><html>" > index.html

Paste both public ip in browser it should work.

Create load balancer ::

EC2- in left panel scroll down:: network & security -- load balancer -- create -- select classic

name - myclassicelb

VPC -- select your default vpc

create an internal load balancer -- it means we can create load balancer for private subnet -- but don't select it here.

Select subnet – press + sign for public subnet

enable advance vpc configuration -- you can select multiple subnets in this -- keep it uncheck here.

Port -- 80 // you can also select 443 if you want but leave it here.

assign security group -- assign webdmz -- the same for ec2 instance.

configure healthcheck

in advance detials ::

timeout -- 2

interval -- 5

rest default.

next -- select your both load ec2 instances

tags -- you can skip

close

Now select your load balancer – instances – instances should be in service. One they are in service you can tese, it will take time.

Now load balancer is created now click your load balancer ::

In description copy DNS name paste it in browser now check it will show content of your EC2 instances randomly.

NO IP is associated with load balancer so use DNS names.

If we shut down one ec2 instance, click your load balancer and down you will see health check failure on ELB screen.

Delete you load balancer and ec2 instances.

**Now configure intelelligent laod balancer that is application load balancer ::**

Target group :: Load balancer routes reqeust to the targets in a target group using target group settings. Targets can be europe, us etc.

EC2 --In left panel scroll down—load balancers -- select Target group -- create target group

Target group name -webgroup1

type - instance (as we are using ec2 instances so select it)

- IP if we using servers outside AWS than we can select IP

- Lambda function -- will be covered in future.

Protocol -- http

Port -- 80

VPC -- select defualt

Health check settings ::

protocol - http

path -- /index.html

Advance health check

2, 3, 5, 6, 200

create

Create load balancer ::

Load balancer -- in left panel - create load balancer -- select application load balancer this time --

Name - MyALB

schecme - internet facing

availability zone -- select all - configure network settings

it wil give a warning as we are not using https -- but ignore it click next -- select security group -- webdmz

Configure routing --

target group - select existing

name -- select webgroup1 // we created above

type -- instance -- next -- register target (no action) -next

review - create.

Traget groups -- select your target group -- target - there will be no targets -- edit -- select you instance - add to registered --save.

Refresh page -- status will be health for instances.

Copy dns names of load balaner and paste it in browser.

**74. Advanced Load balancer theory ::**

Sticky sessions enable your users to stick to the same EC2 instance. Can be useful if your storing information locally to that instance.

Cross Zone Load balancing enables you to load balance across multiple availability zones.

Path patterns allow you to direct traffic to different EC2 instances based on the URL contained in the request.

**75. Autoscaling group lab ::**

**i) Create launch configuration -ec2-left panel -autoscaling-launch configuration**

**ii) on last page -- select create an autoscaling group using this launch configuration**

**iii) name,select minimum instances to start with, vpc,subnet , use scaling policy**

**iv) give cpu as 80% to add one more instance and tag name and create**

**v) Terminate one instance it will start a new one. And if load crosses 80% it will add new instance.**

Autoscaling allows us to configure a cluster of sever which can be autoscaled based on load metrics.

EC2 -- IN left panel -- select autoscaling -- launch configuration -- create launch configuraiton -- select ami --

name -- my-asg-launch-configuration

iam role -- admin\_role or any other

Advanced details --

Bootstrap script for http server user data section ::

#!/bin/bash

yum update -y

yum install httpd -y

service httpd start

chkconfig httpd on

cd /var/www/html

echo "<html><h1>Welcome to cloud webserver autoscaling</h><html>" > index.html

IP address type -- select first.

next -- add stroage -- default -- add security group - webdmz

-- create launch configruation

// this will not create any ec2 instnace till now.

ON same page once launch configuraiton is created -- click create an auto scaling group using this launch configuration

group name -- my-asg

starts with -- 3 //means there will be always 3 instances avaiable online. If one crash or terminate it will start a new one to keep 3 instances up.

select subnet -- you can select multiple subnets so that instances will be in different Aavalibility zones.

Advanced details :: you can select laod balancer and health check configuraiotn -- but leave it here as default.

configure scaling policy -- use scaling policies to adjust capacity of this group --

scale between -- 3 to 6 //means there will be minimum 3 and maximum 6 instances.

metric type -- select average cpu utilisation

traget value -- 80 //means if cpu utilisation 80% it will add one more instance.

Instances need = 60 seconds. //means instances will start to take load in 60 seconds.

-- next -- next -- tag -name - myasg -reveiw -- create

Now if you terminate one ec2 instance -- itw will automatically starts a new one. and if cpu utilisation goes above 80% it will add a new instance.

**76. HA architecture**

Always design for failure.

Use mutiple AZ's and multiple regions where ever you can.

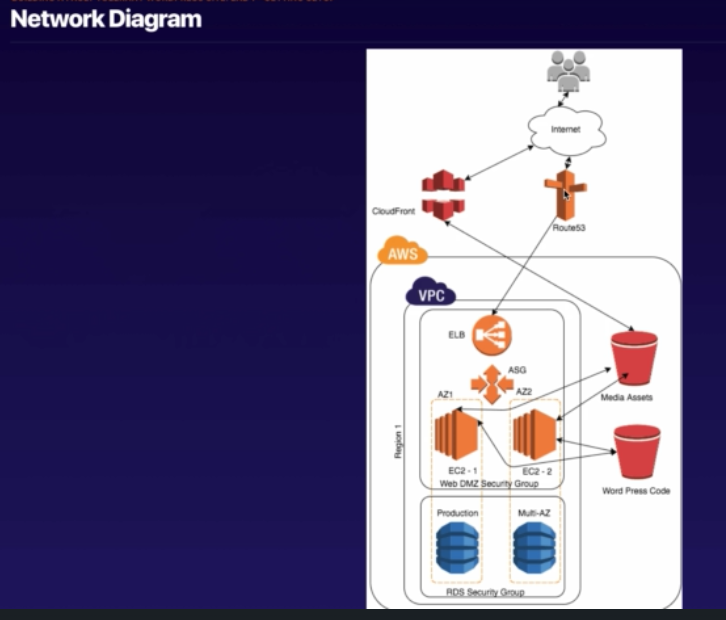
Know the difference between Multi-AZ(DR- DIsaster recovery) and read replicas(FOR IMPROVING PERFORMANCE) for RDS.

know the difference between scaling out (adding more instnaces) and scaling up(increase resources in ec2 instance)

Read the question carefully and always consider the cost element.

Know the different S3 storage classes.

**77. HA Word press site -- building Fault tollerant wordpress site**



**78-79 Setting up EC2 and adding resilience and autoscaling.**

i) Create S3 bucket one for code and one for media files and create a role to access s3 bucket.

ii) Create a cloud front distribution --networy & content-cloud front. select media bucket as original domain.

iii) Create secuirty group or use existing group which we have to use to access db server and open mqsql/aurora port for webdmz secuirty group in it.

iv) Provision database instance rds, type mysql --dev/test -- db.t2micro-1gb ram and secrity group give the above one. delete rest.

v)Provison EC2 instance install wordpress, using bootstrap script.

vi) Paste you public ip of ec2 instance in browser and get started to install wordpress.

vii) create a post , title and uplaod pictures and view them by "view post button". Or in wp-content direcotry on server images file can be seen.

vii)Add redudency by copying images loaded to wordpress to media bucket and code directory(var/www/html) to code bucket.

viii)Enable URL rewrite (to use cloud front)

ix))Edit bucket policy make it public

x))Now create a load balancer register wordpress server to it and create a domian and create a recordset and assign load balancer to this recordset.

We registered wordpress server to load balancer, registered load balancer to a domain name here.

Adding Resilience steps ::

In this phase we will made our word press main server as writer node which only admin will access and load data and posts., and create readernodes which end users will access. buckets will be synced with writer node and reader nodes pick data from synced buckets. Steps as below ::

xi)Enable auto sync of s3 buckets to /var/www/html as reader node will read data from bucketes.

xii)Create a machine image AMI to create new instances for resilience and scalebility

xiii)Make WP ec2 instance write server //on which code and media changes will take effect and sync buckets with this servers directory

xiv)Create a auto scaling group and select load balancing and target group for our load balance. this will add our all reader instances created in auto scaling group to be added to load balancer.

xv) Bring writer node out of target group so that it won't be accessible using domain name.

paste domain\_name you can see reader node and if we paste domain\_name/wp-admin -- writer node it will ask for username/password.

xv) Test availablity by failing one node, you can see activity history in auto scaling group.

i) S3 buckets

s3 - create -- name -- acloudguru-bucket-code-2019

region -- select desried resion.

create

//ly create second bucket --name -- acloudguru-bucket-media-2019

create a role to access S3 BUCKETS ::

services -- IAM -- CREATE ROLE --

CHOOSE A Service --EC2 -- next -- type s3 in search box

select admins3full access -- next

role name - S3FORWORDPRESS

NEXT

CREATE role

ii)Create a cloud front distribution

services -- Networking and content delivery -- cloud front

create cloud front -- get started --

original domain name -- select media bukcet

rest all default

create distribution

iii) Open mysql port for my webdmz secuirty group.

servcies -- vpc -- secutiry groups --

we have to open our rds group for webdmz security group.

Select one rds group -- inbound - edit - add rule

type --mqsql/aurora

source – custom – in next filled of source – type sg (than it will display all security groups) select your webdmz security group which you will assign to your webserver.

iv) Provision rds instances

Services –rds --databases -- mysql - next -- dev/test-mysql //rest are paid.

Database instance class -- db.t2.micro - 1 vCPU, 1 GB RAM

Multi-az deployement --select create replica in different zone

Storage type -- general purpose SSD

dbname -- acloudguru

username -- acloudguru

password -- acloudguru

next

vpc - defualt

public accessiblity -- no

choose exisiging vpc security gorup -- select rds group to which we added rule in last step and remove other group.

Database options

db name -- acloudguru

rest all default

create database

v.) Provison EC2 instance install wordpress, uplaod images to wordpress

EC2 --LAUNCH -- IAM ROLE -- S3FORWORDPRESS //THE role we created in step 1.

In bootstrap paste wordpress script

#!/bin/bash

yum update -y

yum install httpd php php-mysql -y

cd /var/www/html

echo "healthy" > healthy.html

wget https://wordpress.org/latest.tar.gz

tar -xzf latest.tar.gz

cp -r wordpress/\* /var/www/html/

rm -rf wordpress

rm -rf latest.tar.gz

chmod -R 755 wp-content

chown -R apache:apache wp-content

wget https://s3.amazonaws.com/bucketforwordpresslab-donotdelete/htaccess.txt

mv htaccess.txt .htaccess

chkconfig httpd on

service httpd start

add storage -- add tags

name -- mygoldenimage

secuirty group -- webdmz

review and launch -- launch -- assgin exisiting key.

Grab public ip address of and login to putty

cd /var/www/html

ls //there will few files for wordpress.

cat .htaccess // to check htaccess has been installed.

service httpd start

service httpd status

Now paste your IP in webbrowser.

Wordpress screen will appear -- press lets go

database name,username,password -- acloudguru

database host -- endpoint of rds instance server // you can get it by below navigation

rds -- select your rds database-- connectivity -- endpoint

submit -- next screen -- copy contents of wp-config.php file

On putty server in html directory create below file

vi wp-config.php

paste contents from above screen.

Now come back to web browser -- click run installation

Site title -- Hello cloud guru

username -- acloudguru

password -- acloudguru

email -- wirte your mail or try to keep blank

install wordpress

login button

username/passowrd -- acloudguru

We will create our first post by in wordpress server

click post in left panel -- add new --

Hello cloud Gurus

hello cloud gurus welcome to my wordpress site.

click images -- upload picture

add one more photo.

Publish

clcik veiw post :: you will be able to see your post as an website.

Now in putty --

cd /var/www/html/wp-content/uploads/2019/02

ls //we can see our files we uploaded.

**vi) Add resilence by copying content to buckets and enable URL rewrite (to use cloud front)::**

Login to putty for your wordpress server

aws s3 ls

aws s3 cp --recursive /var/www/html/wp-content/uploads s3://acloudguru-bucket-media-2019

//ly copy the code to code bucket

aws s3 cp --recursive /var/www/html/ s3://acloudguru-bucket-code-2019 //by this if ec2 instance goes down than in autoscaling group other ec2 instance will be created and can pull down this code.

aws s3 ls s3://acloudguru-bucket-code-2019

**Rewrite URL for cloud front distribution::**

cat .htaccess file //this will store the cloud front url to enable url rewrite.

rewriterule ^wp-content/uploads/(.\*)$ http://paste\_your\_cloud\_front\_url\_domain\_name/$1 [r=301,nc]

services -- network & content delivery -- cloud front -- select your cloud front -- domain name copy it

//this will tell the webserver that conents of wp-content/upload directory will be view by cloud front distribution url.

aws s3 sync /var/www/html s3://acloudguru-bucket-code-2019

It will sync file .htaccess which is changed.

Now enable URL rewrite for http server

cd /etc/httpd/conf

cp httpd.conf httpd.conf\_bkp

vi httpd.conf

Under AllowOverride controls in .htaccess

AllowOverride All

save and exit

service httpd restart

vii). Edit bucket policy make it public and check cloud front is wokring

s3 - select your bucket --permissions -- bucket policy

Paste the content as in video.

Login to wordpress -- posts -- hello cloud gurus click this

Now image will come -- right click on image -- copy image and in another web browser paste it -- you can see the image – which means cloud front is working.

viii) Now create a load balancer register wordpress server to it and create a domian and create a recordset in domain and assign load balancer to this recordset.

services - ec2 - load balancer -- create load balancer -- application load balancer

name -- MyALBWP

internet facing

select all AZ

next -- security gorup -- webdmz

next --target group

new

name -- mywppressinstances

type -- instance

health checks

path -- /healthy.html

advance health check

2 3 5 6 200

next -- register target -- select target - create.

Give domain name:

service -- rout53 -- select hosted zones under dns management -- create record host -- in right panel

name -- text box blank let it be a naked domain after text box hellocloudguru2019.com

Alias - yes

alias target -- select your laod balancer

rest keep as it it.

create

Register your web instance to target group.

ec2 - target groups -- tragets -- edit -- select server mygoldnimae -- add to registered – save //its already done above. If target group is not showing your server than do it again.

Paste hellocloudguru2019.com in browser it will show images from your updload -- and copy image and paste it in a different browser image will open.

ix). Enable auto sync of s3 buckets to /var/www/html and create AMI to create autoscaling server groups reader nodes ::

This is reverse because here we have to create AMI to be used to create autoscaling group of servers. These servers will access data from buckets and send to client.

Later we will change this line in main WP server which is already created as that will be the write server rest are read server.

root user on your wordpress server ::

crontab -e and save below line.

\*/1 \* \* \* \* root aws s3 sync --delete s://acloudguru-bucket-code-2019 /var/www/html

Now open bucket -- and uplaod a file to bucket.

ON server cd /var/www/html and ls // you can see your file you uplaoded. //ly if a file will be deleted from s3 .. it will also be deleted from server as per crontab command.

x)Create a machine image AMI to create new instances for resilience and scalebility::

ec2 - running instance -- select instance for wordpress -- action - create image

name -- MyWPReadNode

Image description -- This is the defualt read node for WP

Create.

//it will reboot your machine so putty session will terminate.

AMI will be created you can see it as below

images (in left panel) -- AMI

xi) Make you wordpress a write server

Write server is one on which code changes and media file uploads will be done.

Login to WP server instance and remove existing crontab entry :: \*/1 \* \* \* \* root aws s3 sync --delete s://acloudguru-bucket-code-2019 /var/www/html

add below lines to crontab ::

root> crontab -e

\*/1 \* \* \* \* root aws s3 sync --delete /var/www/html s://acloudguru-bucket-code-2019

\*/1 \* \* \* \* root aws s3 sync --delete /var/www/html/wp-content/uploads s://acloudguru-bucket-media-2019

Now you can uplaod a file and check that in bucket.

xxii) Create a auto scaling group and create reader nodes and login to reader node and writer node ::

ec2 -- auto scaling -- auto scaling group -- create auto scaling gorup -launch configuraion --

create new -- don't select server here but select MY AMI IN LEFT panel -- select ami you created -- next - t2micro -- next

name -- MywordpressLC

IAM role -- S3forWP

advance details ::

boot strap script

#!/bin/bash

yum update -y

aws s3 sync --delete /var/www/html s://acloudguru-bucket-code-2019

nex add storage

security group --webdmz

create and launch

create -- slect key

create auto scaling group

name -- my\_WP\_RN (READER NODE)

GROUP SIZE -- 2

NETWORK -- VPC (SELECT DEFUALT OR YOUR VPC)

SUBNET -- SELECT ALL SUBNETS

ADVANCE DETAILS ::

lOAD BALANCING -- TICK IT

TARGET GROUP – MYWPINSTANCES //this target group is associated with load balancer which is associated with domain name.

HEALTH CHECK -- ELB

GRACE PERIOD = 60 SEC

NEXT -- YOU CAN SELECT KEEP INITAIL SIZE TO SELECT AUTOSCALING.

KEEP INITIAL SIZE HERE.

next -- name -- myreadernodes

create auto scaling group

close.

Bring you writer node out of target group ::

select your target group

ec2 - traget group - select your target group -- targets

edit -- select wirter node wp -- remove - save.

Now till this time your new instances will be up with name myreadernodes.

Once they are up they will be in target group you can check.

Now login to your site route53 address ::

hellocloudgurus2019.com -- it will present reader nodes.

If we havn’t taken domain\_name than wirter dns of load balancer it will show reader-nodes.

if you wirte

hellocloudgurus2019.com/wp-admin // it will go to writer node and prompt for login details too.

ON writer node -- posts -- add a new post

This is a test cloud gurus.

update a picture.

//image might take few mintues to propagate due to cloud frontbut your proceed to publish.

click publish button on right side.

After 10 minutes refresh page -- you can see image.

Now login to reader nodes.

you can see image and copy image and open in new browser.

xiii). Test availablity by failing one node ::

ec2 - select one of reader node -- action -- terminate

Now you can test site is still up.

Auto scaling group -- event history -- you can see one instance down-- and another new instance it will bring up.

xiv). Testing db failover to another AZ ::

rds -- select db -- actoins -- reboot -- clcik reboot with failover -- this options starts your db on a different Availablity zone.

If this db is serving a website than that website will be down for couple of minutes once the db is up in another AZ website will start working again.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

80. Clean up ::

cleanup :: remove ec2 instance - target group - load balancer cloud distribution --s3 buckets everything.

81. cloud formation ::

Cloud formation is a way of completely scripting your cloud environment.

Quick start (on aws.amazon page) is a bunch of cloud formation templates already built by aws solutions architects allowing you to create complex environments very quickly.

services - mgt. and goverance -- cloud formation -- create stack in left panel -- use a template -- wordpress blog -- next --

stack name -- mywpblog

dbanme,password,rootpassword, dbuser -- acloudguru //for all

keyname - select your key

next -- name -- mywpCFT (CLOUD FORMAITON TEMPLATE)

NEXT -- create

wait for some time -- your stack will be create\_complete

select output tab -- it will give URL to our wp site.

paste that in browser and you can install wordpress but don't do it here.

Now in resources tab ::

your webserver

securty group.

//ly we can create complex cloud formations by using create stack button.

Now delete your cloud formation ::

cloud formation -- select - action - delete

82. Elastic beanstalk ::

With elastic beanstalk, you can create quickly deploy and manage applications in the aws cloud without worring about the infrastructure that runs those applications. YOu simply upload your application, and elastic beanstalk automatically handles the details of capacity provisioning, load balancing, scaling and application health mornitoring.

services -- compute -- elastic beanstalk-- get started --

application name -- hellocloudguru

platform -- PHP

applciaiotn code -- sample application

create application

its created a

security group

bucket

Once created at top there will be a url clcik , it will open you website.

Now click configuration in left tab :: you can change your configuration like instances, laod balancer,capacity etc.

Delete your beanstalk.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Section 9 Applications**

84. SQS

Amazon SQS is a distributed queue system that enables web service applicatoins to quickly and reliably queue messages that one component in the application generates to be consumed by another component. A queue is a temporary repository for messages that are awaiting processing.

There are two types of queue ::

Standard queues (default) -- it also process in order but sometimes due to high throughput one of message can go out of order.

Fifo queues -- strict ordering and no duplicate.

Exam tips ::

SQS is pull based, not pushed based.

Messages are 256 KB in size.

Messages can be kept in queue from 1 minute to 14 days, default retention period is 4 days.

Visibility time out :: This is the amount of time for which a message is invisible in SQS queue after a reader picks up that message. If the job is processed before visibility time out expires the messaged is deleted from SQS, if its not processed message is visible again and is picked by another readeer for processing. This could result in the same message being delivered twice. Visibility timeout maximum is 12 hours.

SQS gurantees that your messages will be processed at least once.

Amazon SQS long polling is a way to retrieve messages from your amazon SQS queues. While the regular short polling returns immediately(even if the message queue being polled is empty), long polling doesn't return a response until a message arrives in the message queue, or the long poll times out.

85. SWF :: SIMPLE WORKFLOW SERVICE

Amazon simple workflow service is a web service that makes it easy to coordinate work/tasks across distirbuted application components.

Task repsent invocations of vairous processing steps in an application which can be performed by executable code, web service calls, human actions and scripts.

e.g. order a book online -- make payment -- a person will pack the order -- delivery will be given.

SQS offers a message-oreinted API while SWF presents a task-oreinted API.

86. SNS -- Simple notification service

Amazon simple notification service is a web service that makes it easy to set up, operate,and send notifications from the cloud.

Like we get billing alerts.

Instananeous, push based delivery(no polling)

Simle APIs and easy integration with applications

Flexible message delivery over multiple transport protocols.

Inexpensive, pay as you go model with no up-front costs.

Web based aws management console offers simplicity of a point and click interface.

SNS VS SQS ::

bOTH MESSAGing services in aws.

SNS - Push

SQS -- Polls (pull)

87. Elastic transcoder

Its a media transcoder in the cloud.

convert media files from their original source format into different formats that will play on smartphones,tablets,PCs etc.

88. API Gateway

API gateway are front door to your aws environment.

YOu can enable API caching in amazon API gateway to cache your endpoint's response. API gateway then responds to the request by looking up the endpoint response from the cache instead of making a request to your endpoint.

Same origin policy -- One page in a website can access another page in website only if they are of same origin/domain.

One page in one domain is able to talk to page in another domain if CORS -- Cross origin resource sharing is enabled in browser.

Exam tips ::

Remember what api gateway is at high level. Ita a door to your aws environment.

API gateway has caching capabilities to increase performace.

API gateway is low cost and scales automatically.

YOU can throttle API GATEWay TO PREVENT ATTACKS.

YOu can log results to cloudwatch.

If you are using javascript/ajax that uses multiple domains with API, ensure that you have enabled CORS on API gateway.

CORS is enforced by the client i.e browser.

89. Kinesis

Streaming data is the data that is generated continuously by thousnads of data sources.

Kinesis is a platform on aws to send your streaming data to.

Kinesis has three types ::

Kinesis streams

kinesis firehose

kinesis Analytices.

Streams :: consists of shards. Data is stored in individaul shards and is persistent with retention of 1 to 7 days. Data stored in shards is picked up by consumers ec2 instances and analyzed resultts are stored in s3 or redshit.

Firhose :: No data persistent. Data comes and need to be analysed as soon as it comes into firehose kinesis. A lamdba function(set of code) is triggered as data comes and analyzed results are stored on s3, redshit etc.

Analytices :: It analyze data on the fly inside kinesis and store analzyed results on s3, redshoft or elastic search cluster.

Exam tips ::

Know difference in types of kinesis. Based on scenerios which kinesis we have to use.

90. Web indentity federation - cognito

Web identity federation lets you give your users access to aws resources after they have successfully authenticated with a web-based identity provider like amazon,facebook or google.

Amazon cognito provides web identity federation with the following features ::

sign-up and sign in to your apps.

Access for guest users.

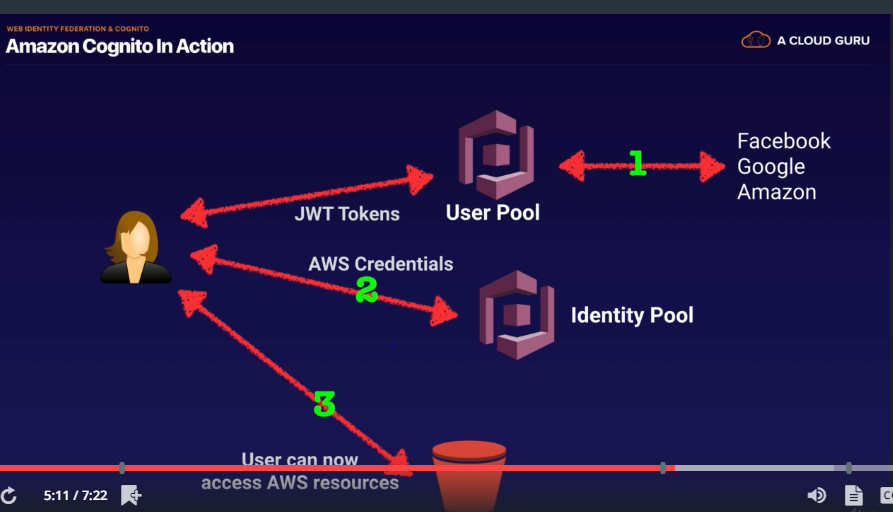
acts as identity broker between your application and web id providers,so you don't need to write any addition code.

synchornizez user data for multiple devices.

recommended for all mobile applicatins aws services.

User pools :: User pool is user based. It handles things like user registration,authentication and account recovery.

Identity pools :: Identity pools authorise access to your aws resouces.



Exam tips ::

Federation allows users to authenticate with a web identity providoer (goodle,facebook,amazon)

The user authenticates first with the web id provider and receives an authentication token, which is exchanged for temporary aws credentials allowing them to assume an IAM role.

Cognito is an identity broker which handles interaction between your applications and the web id provider (your don't need to wirte your own code for this.)

**SERVERLESS**

**91. Lambda Concepts**

92. Lambda concepts

Lambda is ultimate extraction layer::

Data centres

hardware

assembly code/protocols

high level languages

operating systems

application systems

application layer/aws apis

then it comes

aws lambda.

Aws lambda is a compute service where you can upload your code and create a lamda function. Aws lambda takes care of provisioning and managing the servers that you use to run the code. YOu don't have to worry about OS,pathing and scaling etc.

we can use lambda in fowlling ways ::

i) event driven compute service where lamda runs your code in response to events.

ii) As a compute service to run your code in response to http requests using amazon.

Why lambda is cool ::

No servers.

Continuous scaling.

super super super cheap.

Exam tips ::

Lambda scales out (not scales up) automatically. Lets suppose we have five reqeusts there will be 5 invocations or executions of lambda function.

Lambda functions are independent, 1 event = 1 function.

Lambda is serverless.

Know what serivces are serverless.

Lambda functions can trigger other lambda functions, 1 event can=x functions if funtions trigger other functions.

Architecture can get extremely complicated, AWS x-ray allows you to debug what is happening.

Lambda can do things globally, you can use it to backup S3 buckets to other s3 buckets.

Know you triggers.

**92. Lets build a serverless webpage using API and lamda**

i) We created a lamda function

ii) changed code of lambda function

iii) Create a trigger for lambda function , add API trigger.

iv) Give permission to API to invoke your lambda function.

v) Deploy your lambda code.

vi) Create a website which will use your lambda function ::

a.) Create a page index.html which will use URL to invoke your lambda function on a button click.

b.) Create a bucket change it to static website and upload index.html pages to it.

c.) Create a route 53 to access you static website.

i)Services -- lambda -- create function -- select author from scratch

- name -- myfirstlamda-function

Runtime -- Python 3.6

role -- create new role from one or more templates

role name -- mylambdaexecutionrole

Policy templates -- Simple mircoservice permission (select this from drop down)

create

ii) ON next screen -- scroll down -- change defualt code with your code ::

def lambda\_handler(event, context):

print("In lambda handler")

resp = {

"statusCode": 200,

"headers": {

"Access-Control-Allow-Origin": "\*",

},

"body": "Anil Guleria"

}

return resp

So if anyone will invoke this lambda function he will get my name in return.

scroll down

Description :: MY first lambda function

memory you can allocate here or timeout parameter we can give.

Select -- vpc, subnet and iam role , security group etc, but select here no vpc.

save // this saved our lamda function.

iii) Now configure trigger ::

IN left under add trigger

click API gateway

Under configure triggers ::

API -- CREATE A NEW API

SECURITY -- AWS IAM

ADD //THIS WILL ADD TIRGGER

SAVE

nOW scrroll dwon we will get a API gateway end point.

This end point Url IF YOU Click it won't work as we need to give permission to our API gateway to invoke labmda function as explained below ::

### iv) Select API gateway – scroll down – click link “[my-first-lambda-function--2](https://us-east-1.console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/fzuzlpyhm0/resources/6iyrwf/methods/ANY)”

It will show a UI (process flow) from any request.

We will delete it and create process flow for get request.

IN left panel

select on any – action -- delete method-- confirm.

actions -- create a method -- select get -- press right tick mark

INtegration type -- lambda function

user lambda proxy integration -- tick mark

lambda function -- start typing anything it will display lambda function select that

save -- it will ask you are going to give permission to api to invoke your lambda function -- confirm.

v.) Deploy your code

action -- deploy --

stage -- default

description -- myproddeployment

click on get -- click on invoke url -- it will return your name as per code.

vi) Create a website which will use your lambda function ::

a.) Create a page index.html which will use URL to invoke your lambda function on a button click.

b.) Create a bucket change it to static website and upload index.html pages to it.

c.) Create a route 53 to access you static website.

a.)Now we have to invoke this thorugh index.html page.

in index.html downlaoded from resource section add the above invoke url.

open index.html in editor

After GET="pASTE INVOKE url"

and save.

b.)Create a s3 bucket -- make it public

name -- hellocloudguru2019.com

click your bucket -- proerties -- static website hosting

use this bucket to host a website --

In first two text boxes ::

index.html

error.html

save

Upload index.html and error.html to your bucket.

select both files -- make public.

Now select your index.html -- at last click link --

you will see a clickme button on top -- click it it will return your name.

c.)Hookup your route53 to your bucket

services -- route53 -- create a recordset for your bucket you create above.

Now you an open your website thorugh browser.

94. Lets build an Alexa skill

i) create a bucket and make it public.

ii) Create a amazon polling and synthesize with your bucket.

iii) Deploy alexa skill to a lambda function

iv) Poll your bucket text to lamda function to be played by alexa

i) create a bucket

services -- s3 -- name -- clouegurupolyassets2019 -- make it public -- select bucket -- edit public access settings button on top -- uncheck all -- confirm

Now make everything inside bucket public

click bucket -- permissions -- bucket policy

paste bucket policy from resources session

and change ARN in resouce tag in policy.

//ARM is under bucket policy button.

save.

ii) Create a amazon polling and synthesize with your bucket.

Polling will create a mp3 file for the text you paste in polling.

services - machine learning -- amazon polling

Paste any text -- a mp3 file will created which will read your text.

language region -- you can select US,UK india anything.

synthesize to S3

type your bucket name in first text box -- clouegurupolyassets2019

click synthesize

Now go back to s3 -- open your bucket -- you will get a mp3 file there , if you will play it will read your text.

iii) Deploy alexa skill to a lambda function

services -- compute --lambda -- create a lambda function --

select -- aws serverless application repository -- alexa skill-kit-nodejs-factskill -- scroll down -- deploy

go back to previous page -- refresh -- you will get your lambda function

scroll down -- it will show your code at last in data tag

write your name -- same format as to other lines.

save

at top of this page there will be ARN -- copy the ARN.

Now login to developer.amazon.com -- alexa skill -- create a new skill --

skill name -- mystudybuddy

default language -- english us // same to your alexa.

create skill

choose template -- fact skill -- choose

IN left panel -- select invocation --

skill invocation name -- cloud facts //any name you can give

click endpoint in left panel

default region -- paste your lambda url here.

save endpoints

IN left panel -- INtents -- getnewfactintent

same utterences

a cloud fact click + sign

build model (click this button on top)

Test

skill testing is enabled in :: developement

open cloud fact -- it will play your text in lambda function

iv) Polly your bucket text to lamda function to be played by alexa

Open your bucket -- select mp3 file -- on right side -- copy the URL FOR THIS FILE.

compute -- lambda -- select your function -- in code -- in data tag -- delete everything and paste your mp3 url in below format and save.

"audio src =\'url\'/"

YOu can test it from deveoper.amazon.com

Test

skill testing is enabled in :: developement

open cloud fact -- it will play your text in lambda function

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Practical flow ::**

1. EFS2 :: Elastic file store is shared storage

Storage – efs – create

Create two instances and attached admin-iam role to instances.

Efs – select your storage – click blue link it will give mount command.

On instance one ::

Cd /var/www mount html directly using above command. And create index.html inside html directory.

Now mount on instance 2, index.html will be visible there.

1. Creating VPC ::

Nework – vpc – create-create subnets(public and private) – Create instances in public and private subnet – Create IGW give your vpc name – creating routing table other then default – add rule – add igw in this rule—now you can access public subnet instancs thourgh igw.

Now public insances can access internet. To make private instances to access interent ::

NAT INSTANCE/NAT GATEWAY

CREAT NAT INSTANCE OR GATEAY – DEFAULT ROUTE TABLE(AS private subnet uses detault rout table by default) – ADD RULE AND SELECT YOUR NAT INSTANCE/GATEWAY – Now you can access internet from private subnet instances.

Use VPC endpoint :: to access internet resources like S3 bucket/dynamo db from private subnet without using NAT INSTANCE/GATEAY.

VPC ENDPOINT – CREATE – AND IN DEFUALT ROUTING TABLE – REMOVE RULE FOR NAT GATEWAY/INSTANCE – ADD RULE AND SELECT YOUR VPC ENDPOINT – NOW YOU CAN SELECT S3 bucket from your private instance.

VPC -- ACL - create new acl - subnet associations -- select public subnet -- add it

edit inbound/outbound rules.

Logging VPC ::

service - mgt and goverence - cloud watch - logs --lets get started or actions - create log group -- name -- vpc-first-log-group --create log group

vpc - select custom vpc - action - create flow log - filter all - destination - send to cloudwatch - select log group - create a iam role for floglogs and assing it here.

Now check your logs -- cloudwatch - select your log group -- see logs.

3.create wordpress site and db -- Create a read only ami - creating autscaling group - assigning it to load balancer target group -- assign load balancer to a hosted zone.

Create an EC2 instance install wordpress bootstrap script -- create a db server mysql using rds create -- create a secruity group assign it to db and and add rule to db security group for wordpress security group webdmz -- now login to wordpress public ip -- lets get started -- install -- database details you can get -- rds - your db -- select database end point.

Now wordpress is ready with db connection.

Create a bucket and create a script on wordpress server to sync html directory with s3 bucket.

Now create AMI to create read only servers.

1. select create autoscalging group -- launch configuration -- select my amis -- select ami created above - select minimum and maximum servers.

Now create application load balancer -- add register all read only servers to its target group.

Now you access your read only server using load balancer url or further you can assign a website name using rout53.

route 53 - register yoru domain -- create a recordset -- assing load balancer url in recordset.

Now on main write server ::

In crontab reverse the script :: to sync s3 bucket with html directory of write server.

4. Lambda function

compute - lambda - create -- code -- write your code.

Select a trigger for your lambda function:: - in left pancel - select api -- give parameters for api -- at last click api link - UI will display -- click any and delete -- action -- create method--select get -- select lambda function -- your lambda function name -- action - deploy.

Now click invoke url you will get output of your code of lambda function.

Create a html page index.html and code it use your invoke url in that.

Create a bucket and make it public - properties -- host a website -- in first two text boxes write index.html and error.html -- create -- upload index.html and error.html to bucket.

Now you can access using index.html url or you can create a route 53 record set an hostname for your website which will use your lambda function.