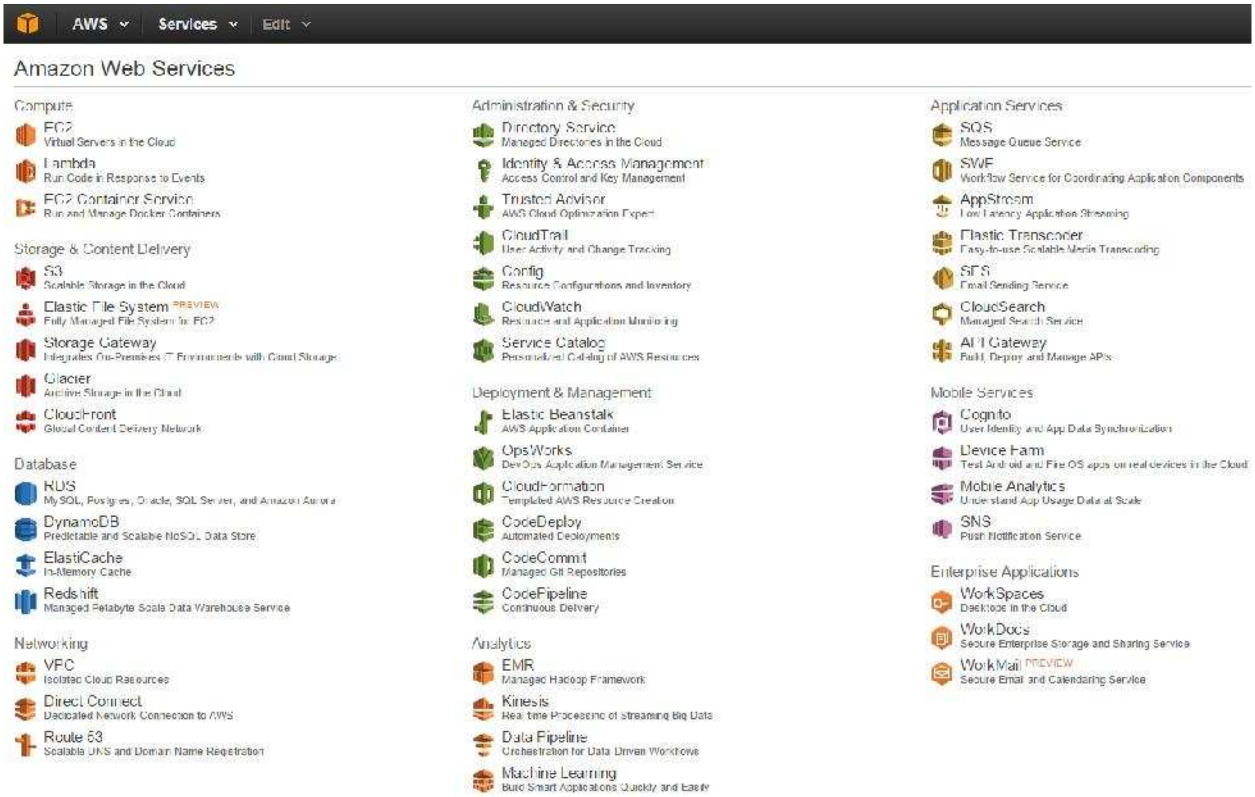
Overview



AWS Services Edit

Amazon Web Services

Compute
- EC2 – Virtual Servers in the Cloud
- Lambda – Run Code in Response to Events
- EC2 Container Service – Run and Manage Docker Containers

Storage & Content Delivery
- S3 – Scalable Storage in the Cloud
- Elastic File System PREVIEW – Fully Managed File System for EC2
- Storage Gateway – Integrates On-Premises IT Environments with Cloud Storage
- Glacier – Archive Storage in the Cloud
- CloudFront – Global Content Delivery Network

Database
- RDS – MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
- DynamoDB – Predictable and Scalable NoSQL Data Store
- ElastiCache – In-Memory Cache
- Redshift – Managed Petabyte Scale Data Warehouse Service

Networking
- VPC – Isolated Cloud Resources
- Direct Connect – Dedicated Network Connection to AWS
- Route 53 – Scalable DNS and Domain Name Registration

Administration & Security
- Directory Service – Managed Directories in the Cloud
- Identity & Access Management – Access Control and Key Management
- Trusted Advisor – AWS Cloud Optimization Expert
- CloudTrail – User Activity and Change Tracking
- Config – Resource Configurations and Inventory
- CloudWatch – Resource and Application Monitoring
- Service Catalog – Personalized Catalog of AWS Resources

Deployment & Management
- Elastic Beanstalk – AWS Application Container
- OpsWorks – DevOps Application Management Service
- CloudFormation – Templated AWS Resource Creation
- CodeDeploy – Automated Deployments
- CodeCommit – Managed Git Repositories
- CodePipeline – Continuous Delivery

Analytics
- EMR – Managed Hadoop Framework
- Kinesis – Real-time Processing of Streaming Big Data
- Data Pipeline – Orchestration for Data Driven Workflows
- Machine Learning – Build Smart Applications Quickly and Easily

Application Services
- SQS – Message Queue Service
- SWF – Workflow Service for Coordinating Application Components
- AppStream – Low Latency Application Streaming
- Elastic Transcoder – Easy-to-use Scalable Media Transcoding
- SES – Email Sending Service
- CloudSearch – Managed Search Service
- API Gateway – Build, Deploy and Manage APIs

Mobile Services
- Cognito – User Identity and App Data Synchronization
- Device Farm – Test Android and Fire OS apps on real devices in the Cloud
- Mobile Analytics – Understand App Usage Data at Scale
- SNS – Push Notification Service

Enterprise Applications
- WorkSpaces – Desktops in the Cloud
- WorkDocs – Secure Enterprise Storage and Sharing Service
- WorkMail PREVIEW – Secure Email and Calendaring Service

-
- 11 Regions with multiple availability zones
  - Places
- Availability Zones (AZ)
  - Separate Data Centers. Each region has 2 more AZ
- 52 Edge locations
  - Used by CloudFront for caching

Identity Access Management (IAM) – User and level of access (chapters 10, 11)

- Features
  - Centralized control
  - Integrated with active directory account
  - User, Group and Roles
  - Multifactor Authentication
  - Password policy
- Roles
  - Allow to assign roles to users and AWS resources. Roles helps avoid storing credentials on resources
- Configure IAM
  - Customer access link 'IAM users sign-in, link'
  - New users
    - Security credential is used to access AWS resources like - Command line, SDK kit, API calls

        - If one forgets the key, it needs to be deleted and another key will need to be created
        - In addition to security credentials, users need a password. Password is required to access the AWS console
    - Roles
        - When you assign a S3 role to any EC2 instance, it has full access to an S3 bucket
        - Roles cannot be assigned to an EC2 instance after that instance has been created. You cannot add or change roles to an existing EC2 instance but you can change permissions of a role already assigned to an EC2 instance

AWS Access via Active Directory (chapter 13)

- Process
    - Go to link
    - Sign in using ADFS credentials
    - Get a SAML assertion in the form of authentication response from ADFS – A cookies is stored saying you are signed into ADFS (Secure Assertive Markup Language)
    - SAML pass to AWS for temporary credentials
    - User is signed in
- Questions
    - Can you authenticate with AD?
        - Yes. Using SAML
    - Are you signed in ADFS first and then granted temporary credentials or visa versa?
        - Yes. ADFS first

S3 (Storage) Concepts

- Object based
- File/Object size is from 1byte to 5TB
- Unlimited storage
- Files are stored in buckets like directory. Buckets can has folder (but these are not considered buckets)
- Buckets have a unique namespace within a region
- Supports metadata, Lifecycle Mgmt, Versioning and Encryption
- Types
    - Standard S3 - 99.99% availability across replicated AZ, 99.999999999% durability
    - Reduced Redundancy Storage S3 - 99.99% availability across AZ, 99.99% durability (use for replaceable data
    - Glacier
        - Data archival
        - $.01/GB
        - Retrieval time is 3-5 hours
- Versioning
    - Store all versions
    - Once enables, versioning cannot be disabled, only suspended
- Life Cycle Mgmt
    - Archive Only

    - Permanently Delete Only
    - Archive and then Delete
- Encryption
    - Everything is encrypted automatically. SSL encrypted end points
    - S3 gives you the choice to manage your keys through AWS Key Management Services (AWS KMS)
- S3 Security
    - All buckets are private by default
        - Select 'Make Public'
    - Allows access control lists (by user, by bucket)
- Functions
    - Static website can be hosted on S3
    - Concurrently upload parts of files – Recommended for files over 100MB, required for files over 5GBs
    - Files are replicates across AZ. Using Eventual Consistency

Buckets Practical

- After creating a bucket, one can set properties
    - Permissions
    - Enable/Disable Static Website Hosting
    - Logging
    - Event Notifications – Put, Post, Cop, etc. SNS notification can send text
    - Versioning
    - Lifecycle
    - Tags
    - Requested Pays
- Create Folders
- Upload files
    - Files are private by default

Versioning Practical

- Once enabled, versioning cannot be disabled, only suspended
- One pays for each version
- Way to restore a file is to delete the delete marker
- Way to permanently delete a file I s to delete 'delete marker' and all its version

Life Cycle Management Practical

- Life Cycle on buckets can be enabled with or without versioning
- Setup rules for LC Mgmt
- Rules
    - Archive Only – Archive after certain period
    - Expire Only - Delete after certain period
    - Archive and Expire – Archive and delete

Cloud Front (CDN)

- Content Deliver Network (CDN) that deliver webpages and other web content to a user based on location, origin and delivery server
- TTL – Time to Live (sec). How long should content be cached
- Content is cached in Edge locations
- Terminology
    - Origin – Origin of all e.g. S3 bucket, EC2, ELB or Route 53
    - Distribution – Edge location
        - Web Distribution – Website
        - RTMP – Media streaming
    - You can have one distribution with more than one origins
- Setup Cloud Front
    - Process
        - Create Distribution
        - Setup Origin Parameters (e.g S3)
        - Setup Distribution (Edge Location)
        - Setup landing page
    - Origin
        - Select domain e,g, S3
        - Restrict Bucket Access – 'Yes' will only allow CloudFront access even though public
        - Grant Read Permissions on Bucket – if yes, will make all public in a bucket and grant read permission to cloudfront
        - Path Pattern – if pdf go to this bucket
    - Distribution
        - Default Root Object – default landing page e.g. index.html

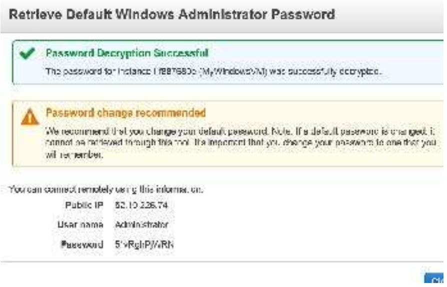
EC2 Essential

- EC2 Options
    - On Demand
        - Fixed hourly billing
        - For unpredictable needs
        - Types
            - General purpose
            - Compute optimized
            - Memory optimized
            - GPU instance
            - Storage optimized
        - Storage Options
            - Local Instance
                - Data stored on local instance store. Stays while the instance is active
            - EBS storage
                - Data is stored on Amazon EBS volume and persist independently of the life of the instance

        - An EBS volume can be mounted on one EC2 instance only. Same EBS cannot be mounted on two
      - Types of EBS Storage
        - General Purpose SSD
          - 99.999% availability, 3IOPS per GB burst 3000IOPS
        - Provision IOPS SSD
          - Large rational or NoSQL db
        - Magnetic
  - Reserved
    - Discounted pricing
    - Upfront payment
    - For predictable usage
  - Spot
    - Name your own price – Bid and ask price
    - Should be used with information that is being replicated

EC2 Practical

- Process
  - Launch Instance
  - Choose AMI
  - Configure Instance
    - Here you can setup 'Roles'
    - Here you can enable 'CloudWatch'
    - Shutdown behavior
      - You don't get charged for stopped instance
    - Protect against accidental termination
    - Add scripts
  - Add Storage
    - Set Size
    - Set Type (SDD, Magnetic)
    - Delete on Termination (delete EBS on instance termination)
  - Tag Instance
  - Configure Security Group
  - Review and Launch
  - Create Key pair
    - Public key is stored by AWS
    - Private key is stored by you (this is needed to log on to the instance)
  - For Windows Instance, you will need to remote desktop into the instance for which you need a password
    - Action 'Get Windows Password'
    - Go to remote desktop and type – IP address, username and password to connect

        - 

          Retrieve Default Windows Administrator Password

          Password Decryption Successful

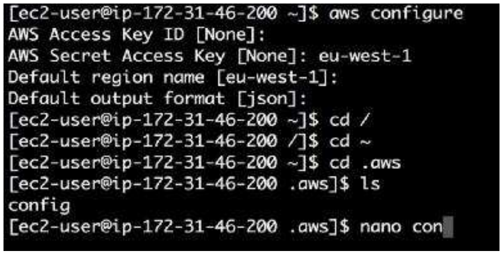
          The password for Instance i-f8876830 (MyWindowsVM) was successfully decrypted.

          Password change recommended

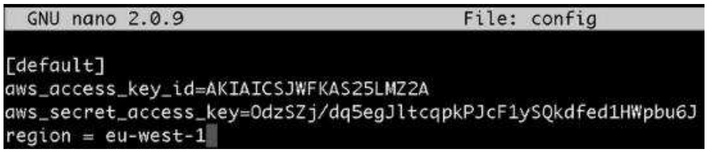
          We recommend that you change your default password. Note: If a default password is changed, it cannot be retrieved through this tool. It's important that you change your password to one that you will remember.

          You can connect remotely using this information:

Public IP	52.10.226.74
User name	Administrator
Password	5'vRghP/WRN

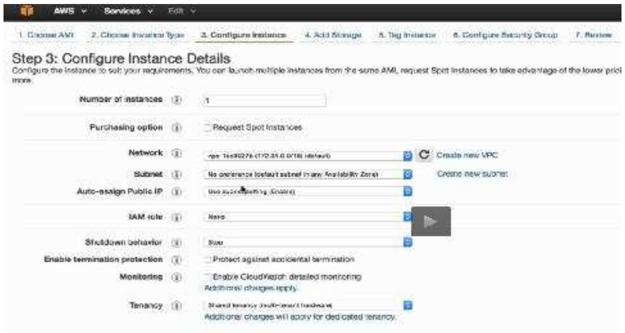
- Connect using command line
    - ls – list content of directory (ls)
    - cd – go into a directory (cd ssh)
    - sudo su – login to administrator and elevates to root user status (sudo su)
    - mv – rename a file (mv MyEC2Key.pem.txt MyEC2Key.pem)
    - Get EC2 Public IP Address
    - ssh – ssh 54.154.110.158 –l ec2-user –i MyEC2Key.pem
    - Are you sure – y, yes
    - You are now logged in
    - sudo su – elevate to root
    - yum update – update the kernel
    - exit – to logout
    - trick ssh 54.154.110.158 –i MyEC2Key.pem – will give you user name
- Other command line commands (webserver)
    - yum install httpd – install apache webserver
    - service httpd status – check webserver status
    - service httpd start – start webserver
    - chkconfig httpd on – starts webserver everytime instance starts
    - cd /var/www/html – default webserver location
    - nano – text editor (nano index.html)
    - history – c – clear the bash history (should do this from a security perspective)
- AWS > Command Line (29)
    - aws s3 help – gives list of s3 related help (hit enter to scroll)
    - q – quit
    - aws configure – set user credentials
        - You will have to give permissions to access S3 or put the user in a group that has access. E.g. Developer with Power user Access has access to S3
    - To access config file
        - 

          ```
          [ec2-user@ip-172-31-46-200 ~]$ aws configure
          AWS Access Key ID [None]:
          AWS Secret Access Key [None]: eu-west-1
          Default region name [eu-west-1]:
          Default output format [json]:
          [ec2-user@ip-172-31-46-200 ~]$ cd /
          [ec2-user@ip-172-31-46-200 /]$ cd ~
          [ec2-user@ip-172-31-46-200 ~]$ cd .aws
          [ec2-user@ip-172-31-46-200 .aws]$ ls
          config
          [ec2-user@ip-172-31-46-200 .aws]$ nano con
          ```
        - Update with credentials

-
    -
        - 

          ```
            GNU nano 2.0.9                              File: config

          [default]
          aws_access_key_id=AKIAICSJWFKAS25LMZ2A
          aws_secret_access_key=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
          region = eu-west-1
          ```
        - Now aws s3 ls will work and you will be access s3
    - S3 Command
        - aws s3 ls s3://acloudguru – list content of bucket acloudguru
        - mkdir cloudguru – makes a directory
        - aws s3 cp s3://acloudguru cloudguru -- recursive – copies directory content from s3 to cloudguru



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pric more.

Number of instances: 1

Purchasing option: □ Request Spot Instances

Network: vpc-1ea90276 (172.31.0.0/16) (default)

Subnet: No preference (default subnet in any Availability Zone)

Auto-assign Public IP: Use subnet setting (Enable)

IAM role: None

Shutdown behavior: Stop

Enable termination protection: □ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy: Shared tenancy (multi-tenant hardware)
Additional charges will apply for dedicated tenancy.

Security Groups

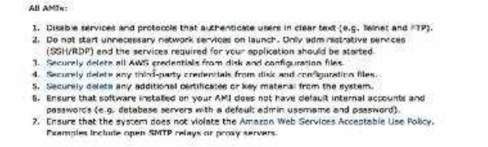
- All inbound is blocked
- All outbound is enabled by default

Snapshot & Volumes

- Create Volume
- Select Type, Size, IOPS, AZ, Snapshot ID, Encryption
    - Availability Zone needs to be same as instance AZ
- Attach to EC2 Instance
- Commandline
    - lsblk – lists available volumes and their mountpoints
    - file –s /dev/xvdf – check if volume (/dev/xvdf) has data
        - if it returns 'data' means it is empty
    - mkfs –t ext4 /dev/xvdf – Format to ext4
    - mkdir /fileserver – Make directory
    - mount /dev/xvdf /fileserver – Mount volume to fileserver
    - cd /fileserver – go to fileserver directory
    - ls – displays everything in fileserver
    - cd … – get out of fileserver directory
    - unmount /dev/xvdf – unmount volume from fileserver
- Snapshot
    - Volume > Actions > Create Snapshot – Create a snapshot from volume
    - Snapshot > Create Volume – Create a volume from snapshot
        - Snapshot could be magnetic but you can create SSD volume with it

- If you need additional disk IO then add additional EBS disk volumes and create a RAID and then migrate data from root volume to the new volume

AMI

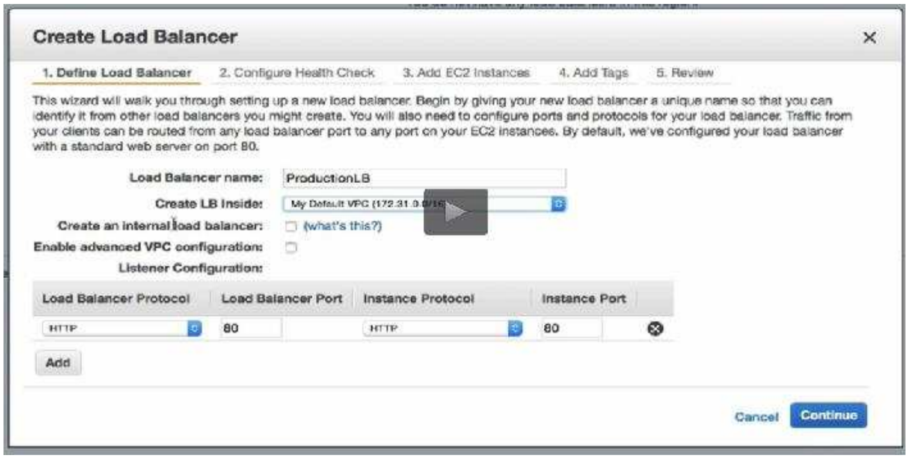
- Volume > Actions > Create Snapshot – Create snapshot of root volume
- Snapshot > Actions > Create Image (from snapshot created)
    - You can change properties if you want
- AMI – to view the image. You can use this as a template for creating a new instance. This could be public/private. If you are going public remember the following – delete bash history, key information stored on instance
- 

  All AMIs:

  1. Disable services and protocols that authenticate users in clear text (e.g. Telnet and FTP).
  2. Do not start unnecessary network services on launch. Only administrative services (SSH/RDP) and the services required for your application should be started.
  3. Securely delete all AWS credentials from disk and configuration files.
  4. Securely delete any third-party credentials from disk and configuration files.
  5. Securely delete any additional certificates or key material from the system.
  6. Ensure that software installed on your AMI does not have default internal accounts and passwords (e.g. database servers with a default admin username and password).
  7. Ensure that the system does not violate the Amazon Web Services Acceptable Use Policy. Examples include open SMTP relays or proxy servers.

Load Balancer

- EC2 > Load Balancers
    - Define LB
        - Internal load balancer – Is this an internal LB with internal IP
    - Assign Security Groups
    - Configure Security Settings
    - Configure Health Check
        - Resp timeout – timeout in how many secs
        - Health check interval – how often it pings
        - Unhealthy threshold – if x consecutive time ping gets timeout means unhealthy
        - Healthy threshold – if x consecutive time ping gets a response means healthy
    - Add instances – add any existing instance to perform check
    - Review and Create
    - Use DNS name to connect to the instance. IP address change every time so use DNS
    - Remember if the health is bad, you can use Autoscaling to take action



Create Load Balancer

1. Define Load Balancer   2. Configure Health Check   3. Add EC2 Instances   4. Add Tags   5. Review

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer name: ProductionLB

Create LB Inside: My Default VPC (172.31.0.0/16)

Create an internal load balancer: □ (what's this?)

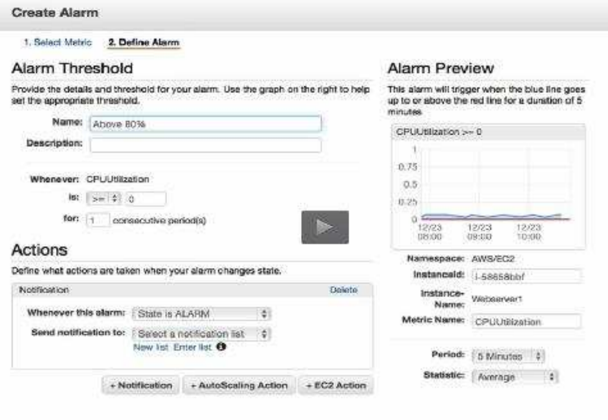
Enable advanced VPC configuration: □

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	80

CloudWatch

- EC2 > Lunch Instance > 3. Configure Instance > Monitoring 'Enable CloudWatch'
- Standard (free by default) vs. Details – Std 5 mins/Free vs Detailed 1 min/Paid
- 10 free alarms
- CloudWatch > Alarm
    - Create Alarm
    - Select Metrics (e.g. EC2 Metrics > CPU Utilization)
    - Define Alarm
        - Notification Action - Email
        - Auto Scaling Action – Add/Remove Instance
        - EC2 Action – Stop/Terminate Instance
    - Create Alarm
- CloudTrail vs. CloudWatch – User Activity Logging/Auditing vs Real-time Monitoring



Create Alarm

1. Select Metric 2. Define Alarm

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: Above 80%

Description:

Whenever: CPUUtilization

is: >= 0

for: 1 consecutive period(s)

Actions

Define what actions are taken when your alarm changes state.

Notification

Whenever this alarm: State is ALARM

Send notification to: Select a notification list

Alarm Preview

This alarm will trigger when the blue line goes up to or above the red line for a duration of 5 minutes

CPUUtilization >= 0

Namespace: AWS/EC2

InstanceId: i-58658bbf

Instance-Name: Webserver1

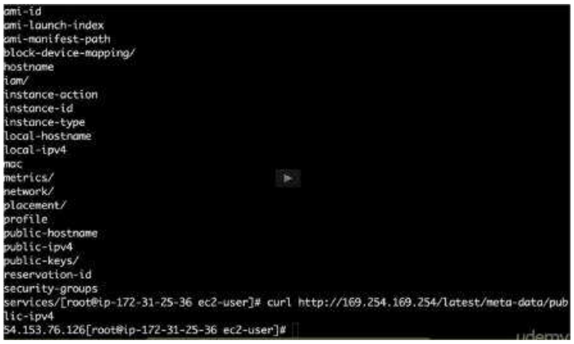
Metric Name: CPUUtilization

Period: 5 Minutes

Statistic: Average

EC2 Instance Meta-Data (important)

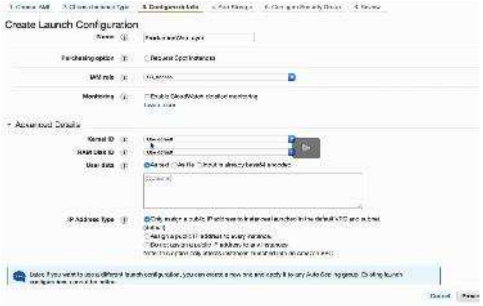
- Type at root – curl http://169.254.169.254/latest/meta-data/public-ipv4



```
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/[root@ip-172-31-25-36 ec2-user]# curl http://169.254.169.254/latest/meta-data/pub
lic-ipv4
54.153.76.126[root@ip-172-31-25-36 ec2-user]#
```

AutoScaling

- Start with no instance
- Launch Configuration

- EC2 > Auto Scaling > Lunch Configurations
- Create Auto Scaling Group
  - Step 1: Create Lunch Configuration
    - Choose AMI
    - Configure Instance
      - Name the launch configuration
      - S3 Roles
      - Add scripts
    - Add Storage
    - Tag Instance
    - Configure Security Group
    - Create Lunch Configuration
    - 

      Create Launch Configuration

      Name

      Purchasing option: ○ Request Spot Instances

      IAM role

      Monitoring: □ Enable CloudWatch detailed monitoring

      Advanced Details

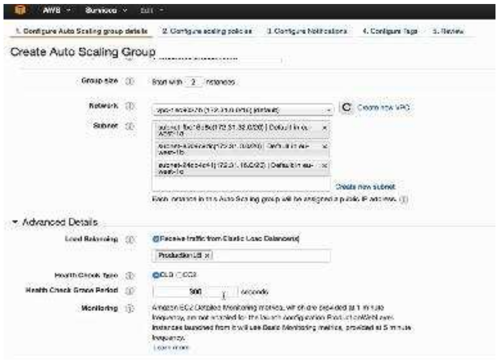
      Kernel ID

      RAM Disk ID

      User data

      IP Address Type

      Cancel
  - Step 2 Create Auto Scaling Group
    - Configure Auto Scaling group details
      - Select group size – this is the starting number of instance that will be created (e.g. 2)
      - Select all available AZ to spread load
      - Advance details
        - Load balancing – receive traffic from Elastic Load Balancer
        - Health Check Type – ELB or EC2. Select ELB if traffic is received from LB.
        - Basically if instance is reported unhealthy or not available, autoscaling kicks in and terminate old and create new instance
    - Configure scaling policies
      - If you select 'Use scaling policies to adjust the capacity of this group' then you can use CloudWatch Alarms (e.g. CPU Utilization) to increase or decrease group size
      - E.g. If an alarm trigger do this action e.g. add instance. Similarly if CPU falls below 10% decrease group size
    - Configure notifications
      - Send a notification to – pick email
      - Whenever instances – launch, terminate, fail to launch, fail to terminate
    - Configure tags
    - Create AutoScaling Group
  - At the end of this, two instance are initiated or the number you picked on 'select group size' in step 1

- 

  1. Configure Auto Scaling group details | 2. Configure scaling policies | 3. Configure Notifications | 4. Configure Tags | 5. Review

  Create Auto Scaling Group

  Group size: Start with 2 instances

  Network: [illegible] (default) — Create new VPC

  Subnet: subnet-[illegible](172.31.32.0/20) | Default in eu-west-1a
subnet-[illegible] | Default in eu-west-1b
subnet-24cb4c41(172.31.16.0/20) | Default in eu-west-1c — Create new subnet

  Each instance in this Auto Scaling group will be assigned a public IP address.

  Advanced Details

  Load Balancing: ☑ Receive traffic from Elastic Load Balancer(s) — ProductionLB ×

  Health Check Type: ● ELB ○ EC2

  Health Check Grace Period: 300 seconds

  Monitoring: Amazon EC2 Detailed Monitoring metrics, which are provided at 1 minute frequency, are not enabled for the launch configuration ProductionWebLaunch. Instances launched from it will use Basic Monitoring metrics, provided at 5 minute frequency. Learn more

Placement Group

- Logical grouping of instance in a single AZ enabling application to participate in a low latency, 10 Gbps networks
- Recommended for applications that benefit from low network latency, high network throughput or both

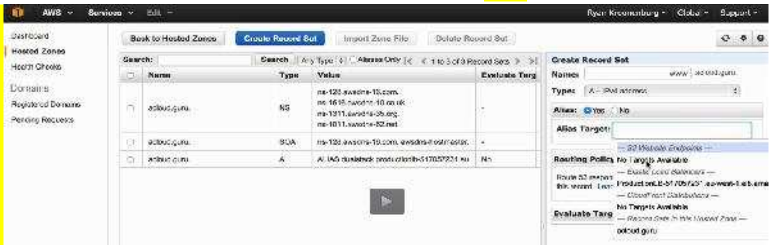
Lambda

- All you need to do is supply the code
- Runs code in response to events and automatically manages the underlying compute resources for you
- Code can be run in response to modification to S3Buckets, messages arriving in kinesis or table updates in dynamodb, AWS API call logs created by CloudTrail, web applications or other web services
- It performs compute administration like server and operating system maintenance, capacity provisioning and automatic scaling, code and security patch deployment, and code monitoring and logging
- What is supported language – Javascript
- What is the availability – 99.99% for both service and functions it operates
- How much does it cost
  - First 1M requests are free
  - $.20 per 1M requests there after
  - Plus duration – calculated from time you code begins executing until it returns or otherwise terminated and is rounded to nearest 100ms
  - Memory - $0.00001667 for every GB second used

Route 53

- DNS Management - Register Domain
  - Create Hosted Zone
    - Domain Names
    - Type – Public or Private
  - Create
    - Will give you list of name servers that you need to specify on e.g. godaddy
- Record Set

  - Different Record Sets
    - NS – Nameserver
    - SOA – Start of Authority or Zone
    - A – resolve DNS to IP address IPv4
    - AAAA - resolve DNS to IP address IPv6
    - C – resolve one name to other
    - TXT – plain text record. Used by email services to lookup sfp record
    - MX – for email
  - Create Record Set
    - Name – like www or leave blank for A
    - Type – A
    - Alias – Yes if you want to connect to ELB
      - So when user types www.agloud .guru, it resolves to ELB that points to two EC2 instance that sit underneath it
    - Routing Policy – simple, weighted, latency, failover, geolocation
    - Evaluate target health – yes or no
    - Create
- Failover between regions not just AZ. If one region fails it will rollover to the other region
  - Remember Route 53 is based on Global and not region like EC2
  - Create two instances that are outside of ELB
    - One in Ireland and other in Frankfurt. So if Irish region fails completely, then it will failover to Frankfurt
  - Health Check > Create Health Check
    - Name – e.g. regionalhealthcheck
    - Specify Endpoint – IP Address/Name of Primary Site
    - Path – index.html (where it is going to check the health)
  - Hosted Zones
    - Create Record Set
      - E.g. www is currently going to ELB
      - 1st one – In Ireland (primary)
        - Names - www2
        - Alias - No, since we have a failover instance at a specific ip addres (irish one)
        - Value – Add ip address
        - Routing policy – Failover
          - Failover type – Primary
          - Associate with health check – Yes
          - Health Check to Associate with – regionalhealthcheck (from above)
      - 2nd one – In Frankfurt (secondary)
        - Names - www2
        - Alias - No, since we have a failover instance at a specific ip addres (Irish one)
        - Value – Add ip address
        - Routing policy – Failover

            - Failover type – Secondary
            - Associate with health check – No. We say yes only for 1st one
- 

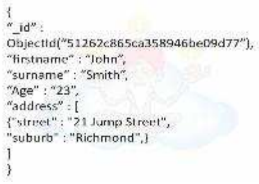
Dashboard
Hosted Zones
Health Checks
Domains
Registered Domains
Pending Requests

Search: Search Any Type Aliases Only 1 to 3 of 3 Record Sets

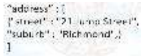
	Name	Type	Value	Evaluate Targ
○	acloud.guru.	NS	ns-128.awsdns-16.com. ns-1618.awsdns-10.co.uk. ns-1311.awsdns-36.org. ns-1011.awsdns-62.net	-
○	acloud.guru.	SOA	ns-128.awsdns-16.com. awsdns-hostmaster.	-
○	acloud.guru.	A	ALIAS dualstack.productionlb-517057221.eu	No

Create Record Set
Name: www .acloud.guru.
Type: A – IPv4 address
Alias: ● Yes ○ No
Alias Target:
— S3 Website Endpoints —
No Targets Available
— Elastic Load Balancers —
ProductionLB-517057221.eu-west-1.elb.ama
— CloudFront Distributions —
No Targets Available
— Record Sets in this Hosted Zone —
acloud.guru
Routing Policy
Route 53 respon this record. Lea
Evaluate Targ

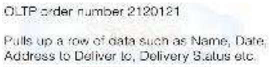
Database

- Relational (OnLine Transaction Processing)
    - RDS (MySQL, SQL, Postgres, Oracle, Aurora)
    - Similar to excel – rows, columns, table, database, records
- Non-Relational Databases (NoSQL)
    - DynamioDB
        - Document oriented db
        - Database > Collection (table) > Document (rows) > Key Value Pairs (fields/column)
        - Starts with curly brackets
        - 

```
{
"_id" :
ObjectId("51262c865ca358946be09d77"),
"firstname" : "John",
"surname" : "Smith",
"Age" : "23",
"address" : [
{"street" : "21 Jump Street",
"suburb" : "Richmond",}
]
}
```

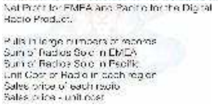
        - Different btw DynamoDB and MongoDB, that DDB does not allow embedded data structures
        - 

```
"address" : [
{"street" : "21 Jump Street",
"suburb" : "Richmond",}
]
```

- Data Warehousing Databases (OnLine Analytics Programing)
    - RedShift
    - `Used for BI
- OLTP vs OLAP – difference is type of queries
    - 

OLTP order number 2120121

Pulls up a row of data such as Name, Date, Address to Deliver to, Delivery Status etc.

    - 

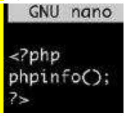
Net Profit for EMEA and Pacific for the Digital Radio Product.

Pulls in large numbers of records
Sum of Radios Sold in EMEA
Sum of Radios Sold in Pacific
Unit Cost of Radio in each region
Sales price of each radio
Sales price - unit cost

- ElastiCache (Database Engine)
    - ElastiCache is a web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud
    - Allows application to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk based DBs
    - Applications can cache repetitive information in memory to not overload databases
    - Types of memory engine
        - Memcached

        - Redis

RDS

- High-level Process
    - Create an EC2 instance with its own SG and apache server installed
    - Create an RDS instance with its own SG
    - Connect the two Security groups
    - Create the connection php file and store on EC2
    - Done
- Step 1: Create a new security group for DB. Don't use the same launch group
- Step 2: Create a new EC2 instance
    - Pick role S3
    - Add a boot strap scripts to install apache and get the EC2 ready
    - Put the security group as 'launch wizard 1'. Do not use the security group created in Step 1
- Step 3: Create RDS Instance
    - Select Engine
        - My SQL
    - Production? – Is this DB for Production? Yes or No
        - No
    - Specify DB Details
        - DB Instance Class – db.t2.micro
        - Multi AZ Deployment – No. Production environment uses Yes
        - Identifier/username/password/password – acloudguru for all
    - Configure Advance Settings
        - Publicly Accessible – No. Do you want anyone in the world to access the DB DNS. Pick No unless you have a good reason
        - AZ – No Preference
        - VPC SG – RDSSecurity (created in Step 1)
        - DB Name – acloudguru
        - Backup Retention Period – 7. How far back should you backup. What is the max # of days = 35 (exam)
        - Backup Window – No Preference
        - Maintenance – No Preference. You only have control on DB and not the machine. Remember. You cannot SSH/RDP into underline server since you don't control the server
        - Launch DB Instance
- Step 4: SSH in EC2 and create a index.php file using nano
    - Go to cd //var/www/html
    - Type nano index.php. You only need this to check whether SQL module is installed correctly
    - 

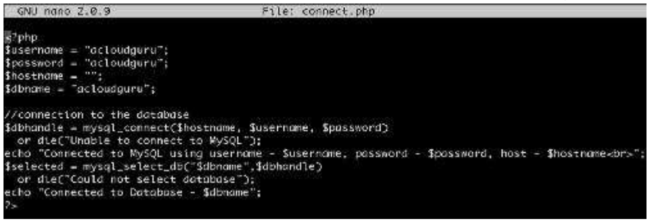
```
<?php
phpinfo();
?>
```

    - Now if you go the ip address of EC2, you see

-

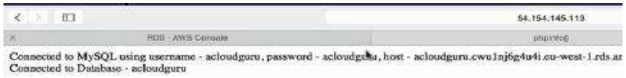
  PHP Version 5.5.25

System	[illegible]
Build Date	[illegible]
Configure Command	[illegible]

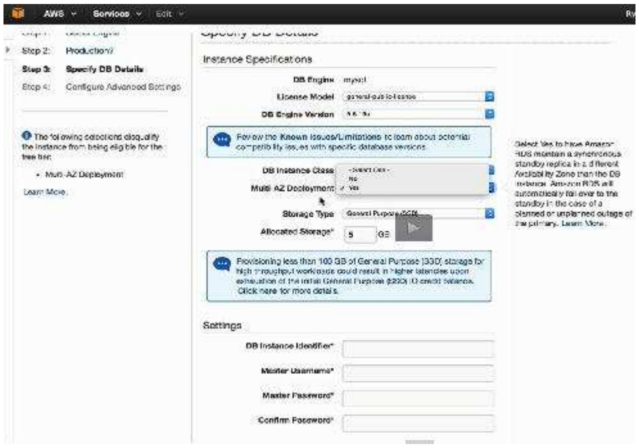
- Step 5: Connect EC2 and RDS DB (by default inbound permission in disabled)
  - Go to VPC > Security Groups
  - Check RDS Security Group > Inbound Rules > Edit
  - MySQL/Aurora > Source > EC2 group id eg. Sg-f07fe895
  - Go back to DB > Copy 'Endpoint' (DNS address)
  - SSH in EC2
  - Go to cd //var/www/html
  - Type nano connect.php (a simple connection string that allows us to connect to RDS)
  - 

    ```
      GNU nano 2.0.9                              File: connect.php

    <?php
    $username = "acloudguru";
    $password = "acloudguru";
    $hostname = "";
    $dbname = "acloudguru";

    //connection to the database
    $dbhandle = mysql_connect($hostname, $username, $password)
      or die("Unable to connect to MySQL");
    echo "Connected to MySQL using username - $username, password - $password, host - $hostname<br>";
    $selected = mysql_select_db("$dbname",$dbhandle)
      or die("Could not select database");
    echo "Connected to Database - $dbname";
    ?>
    ```
  - Hostname here is DB 'endpoint
  - Now go to 'EC2 ip address/connect.html
  - Should go to a page that say, 'Connected to MySQL'
  - 

    ```
    Connected to MySQL using username - acloudguru, password - acloudguru, host - acloudguru.cwu1nj6g4u4i.eu-west-1.rds.ar
    Connected to Database - acloudguru
    ```
- Additional RDS notes
  - You can edit properties of DB after created (size, instance class, change password, security groups, DB engine, multiple AZ)
  - Restore to Point in Time
    - Restore to a point in time based on last update. This will create a new instance with a new end point
  - MySQL vs SQL Server
    - 5-3TB vs 200GB-1TB
    - SQL Server - License Model – You can bring your own license
    - 3306 vs 1433. Different port 1433 (so you have to open 1433)
    - For SQL Server - You can't change the storage size. You will have to create a new DB and migrate data
    - SQL Server does not come with a dataB. You need to create after setting up

- 

Step 2: Production?
Step 3: Specify DB Details
Step 4: Configure Advanced Settings

The following selections disqualify the instance from being eligible for the free tier:

- Multi-AZ Deployment

Learn More.

Specify DB Details

Instance Specifications

DB Engine: mysql
License Model: general-public-license
DB Engine Version

Review the Known Issues/Limitations to learn about potential compatibility issues with specific database versions.

DB Instance Class: - Select One -
Multi AZ Deployment: No / ✓ Yes
Storage Type: General Purpose (SSD)
Allocated Storage*: 5 GB

Provisioning less than 100 GB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) IO credit balance. Click here for more details.

Select Yes to have Amazon RDS maintain a synchronous standby replica in a different Availability Zone than the DB instance. Amazon RDS will automatically fail over to the standby in the case of a planned or unplanned outage of the primary. Learn More.

Settings

DB Instance Identifier*
Master Username*
Master Password*
Confirm Password*

DynamoDB

- Notes
  - NoSQL DB
  - Always stored on SSD
  - Spread across 3 geo distant data center
  - Read Type
    - Eventually consistent read – Put a record in A but not available in B (within a second). Best Read Performance
    - Strongly Consistent Reads – Put record in A and will be read only when replicated across all location
  - Pricing
    - Provisioned throughput capacity
      - Write throughput - $0.0065/c/hr for every 10 units (unit can handle 1 write per second)
      - Read throughput - $0.0065/c/hr for every 50 units
    - Storage costs of $0.25GB per month
- Practical DynamoDB
  - Step 1: Primary Key
    - Table Name
    - Primary Key
      - Type – Hash and Range, Hash e.g. hash and range
      - Hash Attribute Name/Type – String, Number, Binary e.g. Number, UserID
      - Range Att Name – String, Number, Binary e.g. String, First Name
  - Step 2: Add Indexes (optional)
    - Ignore – not for exam, leave as default
  - Step 3: Provisioned Throughput Capacity
    - Pick read/write units e.g. 1, 1
  - Step 4: Throughput Alarms (optional)
    - Use Alarms e.g. if read capacity units consumed > 0.8, send notification to email
  - Step 5: Summary/Create
- Browse to see information
- Use Append to add entry

- Adding new userid creates new record, since it is the primary key. You can also click on 'New' under list tables
- Exam tips
    - Doesn't come much
    - What is Amazon NoSQL
    - Billing criteria
    - Comes in SSD storage only
    - Spread across 3 distinct data centers
    - Eventual vs Strong

Redshift

- Data warehousing service
- Configuration
    - Single Note (160GB)
    - Then scale to Multi note
        - Leader note – receive queries
        - Compute note – Upto 128 Compute – competes queries
- Performance – 10 Times Faster and tenth cheaper than traditional DW solutions
    - Column Data Storage
        - Instead of storage as row it stores data as column which is ideal for analytically
        - Required fewer IOs thereby improving performance
    - Advance Compression – Columnar data stores can be compressed better because similar data is stored sequentially on disk
    - Take less space than traditional warehousing solutions
    - Massively Parallel Processing – distributed data and query across nodes
- Pricing
    - Compute Node Hours
        - 1 unit per note per hour
        - Leader node hours are free if you have compute node running
    - Backup
    - Data transfer within VPC
- Security
    - Encryption in transit SSL and at rest AES 256 encryption
- Availability
    - Only available only in 1 AZ (vs. data being spread across 3 AZ)
    - Can restores snapshots
- Exam Tips
    - 10 times faster – column, compression, MPP
    - Leader node is free
    - Max Compute Node – 128
    - Cannot do across multiple AZ
    - Can restore from snapshot but failover is not automatic

ElastiCache (Low Priority)

- Webservice that makes it easy to deploy, operate, and scale an in-memory cache in the cloud
- Used for read heavy and compute intensive application
- Engine
  - MemcacheD
  - Redis

Aurora (Low Priority)

- MySQL compatible relational DB engine
- Start with 10GB, scales 10gb increments to 64Tb
- Compute resource can up to 32vCPs and 244GB memory
- 2 copies is contained in each AZ with a minimum of 3 AZ. 6 copies of your data
- Can handle the loss of up to two copies of data without affecting DB write availability and up to 3 copies without affecting read availability
- Is self-healing – scan and repairs errors
- Exam Tip
  - MySQL Compatible
  - Anything that used MySQL
  - 2 copies, 3AZ, 6 sets of Data
  - Self-healing
  - Can handle the loss of up to two copies of data without affecting DB write availability and up to 3 copies without affecting read availability

VPC Essentials

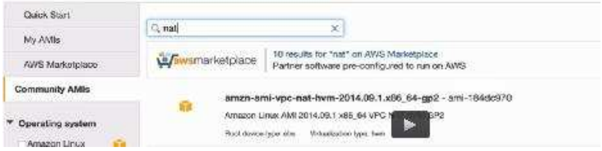
- Amazon Virtual Private Cloud
- Default VPC
  - All subnets have internet gateway attached
  - EC2 instances have both public/private address
  - If you delete default VPC you will have to call Amazon to get it back
- VPC Peering
  - Connecting multiple VPC
  - Connect via a direct network route using private IP addresses
  - You can peer VPCs with other AWS accounts
  - Peering is in a star configuration
- VPC Restrictions
  - 5 Elastic IP addresses
  - 5 Internet Gateways
  - 5 VPCs per region (can be increased upon request)
  - 50 VPN connection per region
  - 50 customer gateways per region
  - 200 route tables per region
  - 100 security groups per VPC

    - 50 rules per security group

Build VPC

- Step 1: Create VPC
    - VPC > You VPCs
    - Name tag: Name of VPC. ACloudGuru-VPC
    - CIDR block – Classless inter-domain routing. It is subnet range. 10.0.0.0/16 (most common)
    - Tenancy – Default or dedicated. Default
        - Exam: if you select dedicated here but you select shared in EC2, you will still be tagged as dedicated and pay for it since your VPC is dedicated
    - Yes, Create
    - Exam: Route tables gets created automatically once you create a VPC
- Step 2: Create subnet
    - VPC > Subnet > Create Subnet
    - Name tag: 10.0.1.0 – us-east-1a
    - VPC: Select VPC created in step 1
    - AZ: Subnets are mapped on 1 AZ. One subnet = 1 AZ. us-east-1a. However you can put all the subnets in one AZ but a subnets cannot span across AZ
    - CIDR block 10.0.1.0/24 (by using 24 you get 254 IP address but Amzn reserves 3 so you have 251)
    - Yes, Create
    - Repeat for two more subnets - 10.0.2.0 – us-east-1b, 10.0.3.0 – us-east-1c
    - Note: Subnets created here, all get assigned to route table that was created by default in step 1 (since you aligned it to VPC). This mean all three subnets can communicate with each other
- Steps 3: Add Internet gateway (allow internet access to EC2 instance)
    - VPC > Internet Gateway > Create Internet Gateway
    - Name tag: Name the gateway (by default it is detached). igw-58583d3d
    - Attach to VPC
    - Note/Exam: You can only have one internet gateway per VPC
- Step 4: Create Route Table (allows internet gateway to communicate to EC2 instances, Create a Route table that will need to be associated to Internet Gateway and Internet Facing Subnet)
    - VPC > Route Table > Create Route Table
    - Name tag: Name
    - VPC: Select the custom VPC you created in Step 1
    - Select the newly create route
    - Click on tab 'Route'
    - Click Edit > Add another route
        - Target: internet gateway created in step 4 - igw-58583d3d
        - Destination: 0.0.0.0/0 (allow route out to the internet)
        - Save
    - Click on tab 'Subnet Association'. Allows you to decide which subnet has internet access vs. not
        - 'Check the Subnet 10.0.1.0'. This will be web facing. Any EC2 instance on this will have internet access
        - Save
- Step 5: Deploy two EC2 Instance

- Deploy 2 instances – 1 with internet access the other with no internet access. Everything remains same except for the following
    - Internet facing
        - Network – Select custom VPC configured in Step 1 ACloudGuru-VPC
        - Subnet – Select us-east-1a 10.0.1.0 for internet facing
        - Auto-assign Public IP – Enable
        - Create New Security Group with SSH and HTTP open with Anwhere/0.0.0.0/0. Launch-wizard-1
        - Create a new key pair – MyNewKeyPair. Copy Key after downloading and opening
        - In Terminal mode – Create MyNewKeyPair.pem using nano. Paste key so you have a version on HDD
        - Type: chmod 600 MyNewKeyPair.pem (this enables permission in VPC)
        - SSH into this instance and hit yes
        - Elevate and run yum update
    - Internal Facing
        - Network – Select customer VPC configured in Step 1 ACloudGuru-VPC
        - Subnet – Select us-east-1b 10.0.2.0 for internet facing
        - Auto-assign Public IP – Disable
        - Select existing Security Group created under Internet facing with SSH and HTTP open with Anwhere/0.0.0.0/0. Launch-wizard-1
            - Exam: SG can stretch across different subnets different AZ. Subnet cannot stretch go across AZ
        - Use existing key pair – MyNewKeyPAir (set above)
        - In Terminal mode – Make sure you have key saved on HDD. If logging from EC2 from above step, recreate key on storage of EC2
        - If newly created on EC2, Type: chmod 600 MyNewKeyPair.pem (this enables permission in VPC)
        - SSH into this instance and hit yes
        - Try to elevate and run yum update. It does not have internet access
- Step 6: Network Address Translation (Allowing internal DB services access to internet)
    - Create new security group EC2 > Security Groups
        - Name: Name the group
        - VPC: assign to acloudguru-VPC
        - Inbound
            - Add http and https
            - Source the DB subnet 10.0.2.0/24 (subnet for DB server)
        - Outbound
            - Only allow http and https (0.0.0.0/0)
        - This allow private subnet to communicate with nat instance that we will create in this SG
- Step 7: Create a NAT instance
    - Pick NAT from community instance

        - 

nat

10 results for "nat" on AWS Marketplace
Partner software pre-configured to run on AWS

amzn-ami-vpc-nat-hvm-2014.09.1.x86_64-gp2 - ami-184dc970
Amazon Linux AMI 2014.09.1 x86_64 VPC N GP2
Root device type: ebs   Virtualization type: hvm

    - Network – Select ACloudGuru-VPC
    - Subnet – 10.0.1.0/24 notice this is the public subnet
    - Auto IP – Disable (you will use elastic IP)
    - SG – Pick the one created in Step 6
    - Use existing key pair
- Step 8: Assign elastic IP
    - Allocate a new address
    - Associate Address to NAT instance created in Step 7
- Step 9: Disable Source/Destination Check
    - EC2 > NAT Instance > Actions > Networking > Change Source/Dest. Check (Exam)
    - Disable
- Step 10: Associate Route Table for NAT
    - Select the AcloudGuru VPC that is currently not associated with any subnets. Not the ITG one
    - Select Route at the bottom
    - Edit
    - Add another Route
        - Target – myNATVM
        - Destination 0.0.0.0/0
        - Save
    - SSH into the private EC and it has an internet access
    - To install SQL
        - yum install mysql -y
- Exam
    - When you create VPC, route tables are created by default
    - Subnets are mapped on 1 AZ. One subnet = 1 AZ
    - SG can stretch across different subnets different AZ. Subnet cannot stretch go across AZ
    - You got a NAT instance and there are servers in private subnet but still can't communicate. What do you do? Ans: Disable source destination check

Access Control List (high-level)

- ACL is like firewall. It has the prime authority and override SG group rule
- It is a numbered list of rule evaluated in order starting with lowest number first to decide whether traffic is allowed in or out with a subnet that is associated with ACL
- When VPC is created, a modifiable ACL is created by default. It allows all inbound/outbound traffic
- Each subnet must be associated with ACL if not it is associate with the default one
- You can only have 1 ACL associated to a subnet
- VPC > Network ACLs
    - Create Network ACL
        - Name: Name
        - VPC – the custom on
        - Default is deny everything

  - Associate Subnet
    - Edit
    - Pick Subnet to associate with
    - Save

Simple Queue Service (SQS) (Important)

- Web Service that give access to a message queue that can be used to store messages while waiting for a computer to process them. Queue is a temporary repository for messages that are awaiting processing
- Messages can contain up to 256 Kb of text in any format
- SQS ensures delivery of each message at least one
- Order of message is not taken into consideration. You can apply sequencing in the message if important
- SQS always pulls messages. EC2 servers polls the queue and pulls messages
- Visibility timeout period only starts when msg is picked up from the queue
- A request is complete when message is deleted from the queue. If not it can be picked up by another EC2 server
- Exam
  - Does not offer FIFO. Order not important
  - 30 secs visibility time out windows by default. 12hrs Max. 12 months in SWF
  - Deliver messages at least once
  - 256kb message size/payload is available but you are billed at 64kb chuck
  - Decouples. What application service allows you to decouple your infrastructure using messaged based queues?
  - You can apply auto scaling

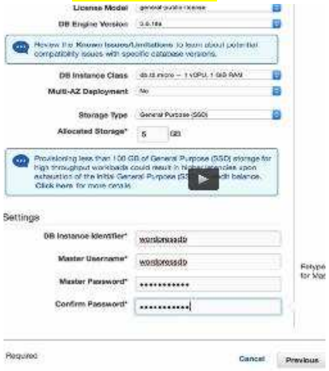
Simple Workflow Service SWF (super low priority)

- Web service that makes it easy to coordinate work across distributed application components. It includes human actions and is broken down into tasks
- SWF vs SQS
  - SWF Task oriented vs message oriented
  - SWF Tasks are assigned only once vs you need to handle duplicated message and may also need to ensure that a message is processed only once
  - SWF Tracks all tasks and events in application vs you will need to implement your own application level tracking especially if you application uses multiple queues

Simple Notification Services

- Web service allows setup, operate and send notifications from cloud
- SNS = push notification vs SQS = pull/poll
- To avoid loss of message, all messages are stored redundantly across multiple availability zone
- Allows grouping multiple recipients using topics

Wordpress Site

- Step 1: Set S3 role
- Step 2: Create 2 Security Groups
  - MyWebDMZ/Default VPC
    - Inbound – http/https/SSH open for all
  - RDSSecurity/Default VPC
    - Inbound – MySQL Source MyWebDMZ Group ID sg-9f46dbfa
- Step 3: Create RDS Instance
  - 

    License Model: general-public-license

    DB Engine Version: 5.6.19a

    Review the Known Issues/Limitations to learn about potential compatibility issues with specific database versions.

    DB Instance Class: db.t2.micro — 1 vCPU, 1 GiB RAM

    Multi-AZ Deployment: No

    Storage Type: General Purpose (SSD)

    Allocated Storage*: 5 GB

    Provisioning less than 100 GB of General Purpose (SSD) storage for high throughput workloads could result in higher latencies upon exhaustion of the initial General Purpose (SSD) credit balance. Click here for more details.

    Settings

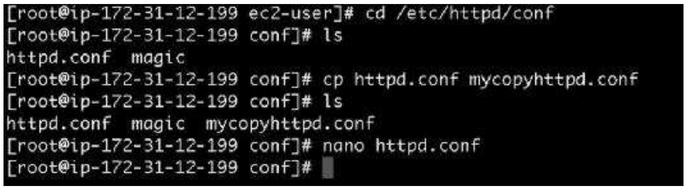
    DB Instance Identifier*: wordpressdb

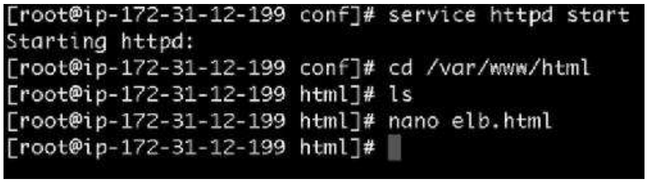
    Master Username*: wordpressdb

    Master Password*: ●●●●●●●●●●●

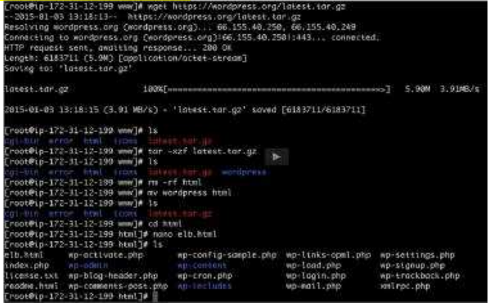
    Confirm Password*: ●●●●●●●●●●●

  - SG – RDSSecurity
- Step 4: Set Elastic Load Balancer
  - MyWordPressLB
  - elb.html
  - SG – MyWebDMZ
- Step 5: Setup Route 53
  - Set domain name to load balancer. Point DNS to load balancer
- Step 6: Create 2 S3 buckets
  - 1 – Code. This will include code and will be private
  - 1 – Application Files CDN
- Step 7: Setup Cloud Front
  - Pick ‘Web’
    - Origin
      - S3 CDN
      - Restrict Bucket Yes, Update Bucket Policy Yes
- Step 8: Launch EC2 instance
  - Give S3 access role
  - Choose existing key pair
- Step 9: Put the new EC2 instance behind the Load Balancer
- Step 10: SSH into EC2
  - yum install httpd php php-mysql – y
  - yum update – y
  - Do url redirect. This will allow us to load images from CDN and not S3

        - 

          ```
          [root@ip-172-31-12-199 ec2-user]# cd /etc/httpd/conf
          [root@ip-172-31-12-199 conf]# ls
          httpd.conf  magic
          [root@ip-172-31-12-199 conf]# cp httpd.conf mycopyhttpd.conf
          [root@ip-172-31-12-199 conf]# ls
          httpd.conf  magic  mycopyhttpd.conf
          [root@ip-172-31-12-199 conf]# nano httpd.conf
          [root@ip-172-31-12-199 conf]#
          ```
        - Allowoverride All (this is outside of scope of the exam)
        - Start httpd service
        - Create elb.html – Success
        - 

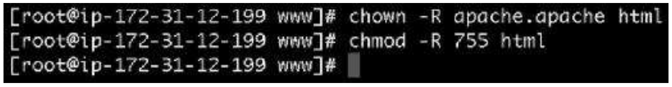
          ```
          [root@ip-172-31-12-199 conf]# service httpd start
          Starting httpd:
          [root@ip-172-31-12-199 conf]# cd /var/www/html
          [root@ip-172-31-12-199 html]# ls
          [root@ip-172-31-12-199 html]# nano elb.html
          [root@ip-172-31-12-199 html]#
          ```

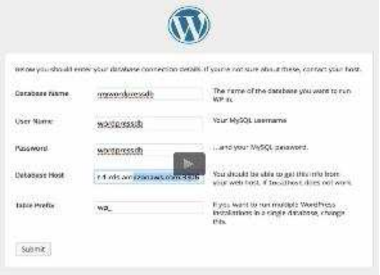
- Step 11: Install Wordpress
    - 

      ```
      [root@ip-172-31-12-199 www]# wget https://wordpress.org/latest.tar.gz
      --2015-01-03 13:18:13--  https://wordpress.org/latest.tar.gz
      Resolving wordpress.org (wordpress.org)... 66.155.40.250, 66.155.40.249
      Connecting to wordpress.org (wordpress.org)|66.155.40.250|:443... connected.
      HTTP request sent, awaiting response... 200 OK
      Length: 6183711 (5.9M) [application/octet-stream]
      Saving to: 'latest.tar.gz'

      latest.tar.gz            100%[======================================>]  5.90M  3.91MB/s

      2015-01-03 13:18:15 (3.91 MB/s) - 'latest.tar.gz' saved [6183711/6183711]

      [root@ip-172-31-12-199 www]# ls
      cgi-bin  error  html  icons  latest.tar.gz
      [root@ip-172-31-12-199 www]# tar -xzf latest.tar.gz
      [root@ip-172-31-12-199 www]# ls
      cgi-bin  error  html  icons  latest.tar.gz  wordpress
      [root@ip-172-31-12-199 www]# rm -rf html
      [root@ip-172-31-12-199 www]# mv wordpress html
      [root@ip-172-31-12-199 www]# ls
      cgi-bin  error  html  icons  latest.tar.gz
      [root@ip-172-31-12-199 www]# cd html
      [root@ip-172-31-12-199 html]# nano elb.html
      [root@ip-172-31-12-199 html]# ls
      elb.html     wp-activate.php        wp-config-sample.php  wp-links-opml.php  wp-settings.php
      index.php    wp-admin               wp-content            wp-load.php        wp-signup.php
      license.txt  wp-blog-header.php     wp-cron.php           wp-login.php       wp-trackback.php
      readme.html  wp-comments-post.php   wp-includes           wp-mail.php        xmlrpc.php
      [root@ip-172-31-12-199 html]#
      ```
    - Run following commands so Wordpress is given permission to write to html
    - 

      ```
      [root@ip-172-31-12-199 www]# chown -R apache.apache html
      [root@ip-172-31-12-199 www]# chmod -R 755 html
      [root@ip-172-31-12-199 www]#
      ```
    - Go to acloud.guru and start installing workpress via the gui
    - 

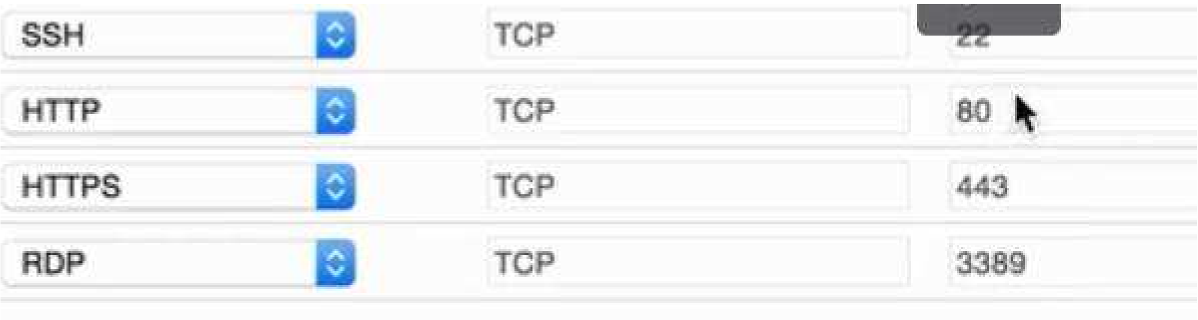
      Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	mywordpressdb	The name of the database you want to run WP in.
User Name	wordpressdb	Your MySQL username
Password	wordpressdb	...and your MySQL password.
Database Host	[illegible]	You should be able to get this info from your web host, if localhost does not work.
Table Prefix	wp_	If you want to run multiple WordPress installations in a single database, change this.

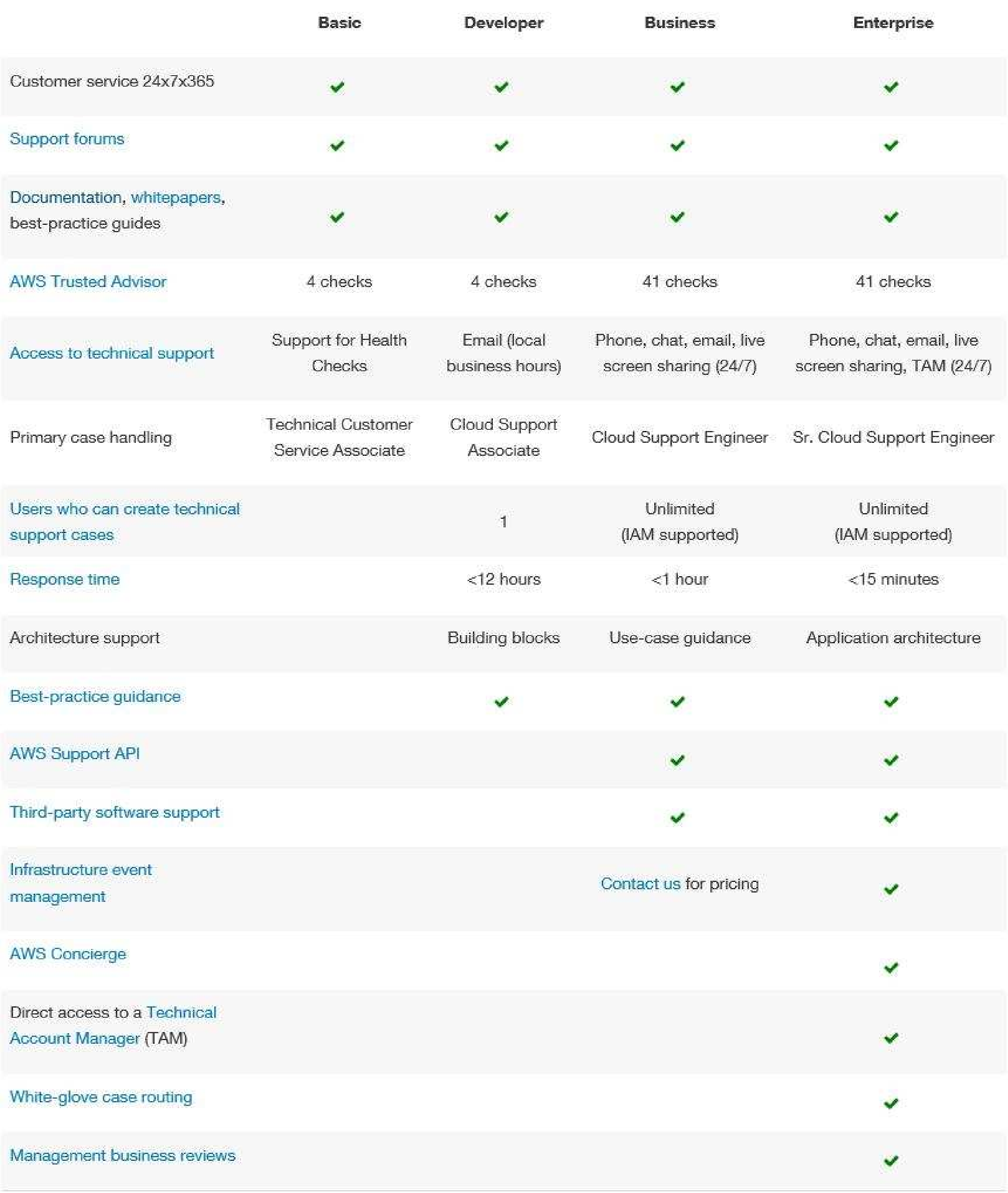
    - Run the Install. Installation done
- Step 12: Configure Wordpress

Whitepapers

Overview of Amazon Web Services



SSH	TCP	22
HTTP	TCP	80
HTTPS	TCP	443
RDP	TCP	3389



	Basic	Developer	Business	Enterprise
Customer service 24x7x365	✓	✓	✓	✓
Support forums	✓	✓	✓	✓
Documentation, whitepapers, best-practice guides	✓	✓	✓	✓
AWS Trusted Advisor	4 checks	4 checks	41 checks	41 checks
Access to technical support	Support for Health Checks	Email (local business hours)	Phone, chat, email, live screen sharing (24/7)	Phone, chat, email, live screen sharing, TAM (24/7)
Primary case handling	Technical Customer Service Associate	Cloud Support Associate	Cloud Support Engineer	Sr. Cloud Support Engineer
Users who can create technical support cases		1	Unlimited (IAM supported)	Unlimited (IAM supported)
Response time		<12 hours	<1 hour	<15 minutes
Architecture support		Building blocks	Use-case guidance	Application architecture
Best-practice guidance		✓	✓	✓
AWS Support API			✓	✓
Third-party software support			✓	✓
Infrastructure event management			Contact us for pricing	✓
AWS Concierge				✓
Direct access to a Technical Account Manager (TAM)				✓
White-glove case routing				✓
Management business reviews				✓