
Linux Academy
Amazon Web Services
Certified Solutions Architect – Associate Level

About this course

- This course is currently being developed to replace the AWS CSA course currently listed on LinuxAcademy.com.
- The prior AWS course includes ALL required knowledge to pass the certification but does not include console updates from AWS.
- The expected completion date of this course is April 1st, 2015. All material will be added as soon as it is created.
- While taking the course “in development” you might run across a few missing items. Feel free to report it to the LinuxAcademy.com staff. We thank you for your patients.


Linux Academy
Amazon Web Services
Certified Solutions Architect – Associate Level

Linux Academy

About this course

- Teaches the student required hands-on concepts for managing, designing, and building highly available fault tolerant environments on AWS.
- Covers the concepts and objectives required to pass the AWS CSA. We will be using Linux as our operating system of choice when required.
- At the end of the course the student have sufficient knowledge to take the AWS CSA exam.
- To date (as of March 2015) over 300 students have taken and passed the CSA by using a LinuxAcademy.com course for a 98%+ pass rate!
- This course will provide all required knowledge and material in order to prepare for the exam and assumes no prior AWS knowledge by the student. Third party materials are not required.

Course Prerequisites

- The certification assumes the student has 1 year of hands-on experience when taking the exam. It is OK not to meet this requirement as our course covers all the required hands-on AWS knowledge. However, the following IT knowledge is a requirement
  - Basic IT terminology and understanding
  - Basic Linux terminology and understanding
  - Have at least 1 year working in the IT field or have the taken Linux Essentials course from Linux Academy

What happens when the console changes from the course videos?

- AWS focuses on a services release cycle of deploy often and deploy quickly. This means there will be times when the console design within the instructor video does not match what you are seeing in your console.

- PLEASE CONSIDER THE FOLLOWING:
  - Generally console design changes are incremental and will be minimal enough not to recreate the video.
  - When there are core concept updates the course WILL be updated.

Linux Academy

What To Do After Taking The Course

- Register to take the exam from Web Assessor. A link can be found here: http://aws.amazon.com/certification
- Move on to the AWS Certified Solutions Architect – Professional level prep course at LinuxAcademy.com (available spring 2015)
- Move onto one of the other associate level courses such as SysOps or Developer


Linux Academy
Amazon Web Services
Certified Solutions Architect – Associate Level

Linux Academy

Preparing For The Exam

- Make sure to read the white papers provided on the right hand side of the course syllabus.
- Ensure that EVERYTHING the instructor does in a video you use a self-paced lab or your own lab environment to follow along. ***HANDS-ON IS KEY!***
- Ask questions if something is not clear, that is what the instructors are here for.
- Do not focus on just quiz questions. The exam questions are conceptual and scenario based. So understanding the concepts and having used the hands-on labs will be the only way to be successful. Use quiz questions only as a "gauge" to ensure you know the concepts.
- Aim to be "qualified" to administer AWS and NOT just to have a "paper csa". Exam dumps only diminish the value of a certification once earned.
- Share your success story or view other students on the Linux Academy community!


Linux Academy
Amazon Web Services
Essentials


Linux Academy
Amazon Web Services
AWS Architecture And Terminology

Regions And Availability Zones

- AWS is made up of regions which are a grouping of independently separated data centers in a specific geographic regions known as “Availability zones”.

- Availability zones work together in a region to make up a collection of your AWS resources. Properly designing applications will utilize multiple availability zones for fault tolerance and failover. AZ’s (as they are known) have direct low latency connections between each AZ in a region but each AZ is isolated from other AZ’s to ensure fault tolerance.

- Availability of regions allows the architect to design applications to conform to specific laws and regulations for specific parts of the world. When viewing a region in the console you will only view resources in one region at a time but they will be across all AZs within that region.

Regions And Availability Zones

- Some AWS services work “globally” and not within a specific region. For example users created in IAM will work across regions

Edge Locations

- An edge location is an AWS datacenter which does not contain AWS services. It is used to deliver content to parts of the world. An example would be CloudFront which is a CDN. Cached items such as a PDF file can be cached on an edge location which reduces the amount of “space/time/latency” required for a request from that part of the world.

Terminology

- Scalability: Ability for a system to expand and contract according to workload demands
  - Resilient
  - Operationally Efficient
  - Cost effective as the service grows
- Fault Tolerant: Ability for your system to operate without interruption in the event of service failures
  - Auto Scaling
  - Route 53
  - Availability Zones
  - Multiple Regions

Linux Academy

Terminology

- Elasticity: Fundamental property of the cloud; the ability for an infrastructure to adapt up and down automatically to a given work load.
  - EC2, AMI, RDS, Route 53, Auto Scaling, Bootstrapping
  - Proactive Cycle Scaling: Scale out based off “known” peak periods
  - Proactive Event-Based Scaling: Scale out in anticipation of increase demand
  - Auto-Scaling Based On Demand: Scale out based on metrics such as CPU utilization, network utilization etc. This is considered horizontal scaling VS vertical scaling.

AWS Services

- AWS services are available and grouped together by “type”. In this certification not all services are required and this course will only focus on the services that are covered by the certification.
- AWS Services are grouped in the following categories
  - Compute and Networking
  - Storage and Content Delivery
  - Database Services
  - Analytics
  - App Services
  - Deployment Services
  - Management Services

A properly architected application will make use of multiple types of services. A focus for automation, fault tolerance, disaster recovery, and high availability should always be the focus of any architected AWS application and WILL require the use of multiple AWS services.


Linux Academy
Amazon Web Services
Compute and Networking Services

Linux Academy

Compute and Networking Services

- AWS provides a robust offering of compute and networking services. These services have different use cases depending on your application type, build, and deployment method. In this course and for the certification we will focus on the following compute and networking services.

- Compute and Networking
  - EC2 (Elastic Compute Cloud) (hands-on)
    - Auto Scaling
    - Elastic Load Balancer
    - EBS Volumes
  - Virtual Private Cloud (hands-on)
  - Amazon Route 53 (hands-on)

Amazon (EC2) Elastic Compute Cloud

- Amazon EC2 provides scalable virtual servers in the cloud. The virtual servers can run different operating systems but most commonly run a flavor of Linux or Windows.

- An EC2 virtual server is known as an “instance” and can be made up of different instance types and sizes.

- Pricing Models
  - Reserved Instances
    - Purchase reserved instances when you know the “amount of usage” you will be using for an instance. If an instance is needed 100% of the time purchasing reserved instances at a discount will reduce costs.
  - On-Demand Instances
    - Are used “on-demand” and are paid for by the hour. If you only need an instance for an our only fire it up for that long!
  - Spot Instances
    - Bid on unused EC2 instances for “non production applications”.

Amazon (EC2) Elastic Compute Cloud

- Auto Scaling: Auto Scaling is a service and method provided by AWS in order to increase the number of instances on-demand based on certain metrics. If your application demand increased un-expectantly auto scaling can scale up to meet the demand and then stop instances as soon as the demand decreases. This is known as “elasticity” in the AWS environment.

- Elastic Load Balancer: Load balancing is a common method for distributing traffic among servers in the IT environment. The Elastic Load Balancer is another service by AWS EC2 that allows you to add instances to the elastic load balancer and distribute traffic among those instances. The elastic load balancer can send traffic to different instances in different availability zones and should often be used with auto scaling and designing for fault tolerance.

- Route 53: Route 53 is a domain management service by AWS. Route 53 will host the internal and external DNS for your application environment It is used commonly with ELB to direct traffic from the domain to the Elb.

Amazon (EC2) Elastic Compute Cloud

- AMI: Amazon Machine Image is a template that contains a pre-built software configuration. Amazon Machine Images are used with Auto Scaling and Disaster recovery.

- Instance Store-backed Instances (Ephemeral Storage)
  - Block level temporary storage over the life of an instance
  - Lives for as long as your instance is NOT turned off/shutdown
- EBS Backed Instance (Elastic Block Store)
  - Network attached block storage
  - Easy to backup with snapshots stored on Amazon S3
  - Can provision additional IOPS to help with I/O or even use an EBS optimized instance to help network traffic between the instance and EBS volume
  - Can be as small as 1GiB and 16,384GiB (16Tib) in size
  - Cannot be attached to instances in a different availability zone
  - Can only be attached to one instance at a time
  - Allows for point in time snapshots

Amazon (EC2) Elastic Compute Cloud

- Up to the customer to manage the software level for security on instances
  - Security groups
  - Firewalls (IP tables, Firewalld, etc)
  - EBS encryption provided by AWS
  - Snapshots can also use EBS encryption
  - AWS EBS encryption utilizes AWS Key Management Service
  - Additional encryption can be to encrypt the entire file system using an encrypted file system.
  - EBS encryption is only available on larger instance types and it is suggested to use an encrypted file system on EBS if using an instance size smaller than M3
  - Apply SSL Cert to the ELB (Elastic Load Balancer)

- AWS Manages the hypervisor and physical layer of security for EC2
  - DDOS protection
  - Port scanning protection (not allowed even in your own environment without permission from AWS)
  - Ingress network filtering

Amazon (EC2) Elastic Compute Cloud

- VPC (Virtual Private Cloud) is one of the core components of AWS and no application should be designed without it. VPC allows for the isolation of AWS resources in the cloud. Resources fired up in a single VPC will be part of the same network and can communicate internally. However, if multiple VPCs are used to provision resources then resources in one VPC are completely isolated from the other VPC by default. Resource sharing between VPCs in the same region can be allowed with VPC peering but is not covered in this certification or course.
- VPC does not cost only the resources within the VPC are what costs

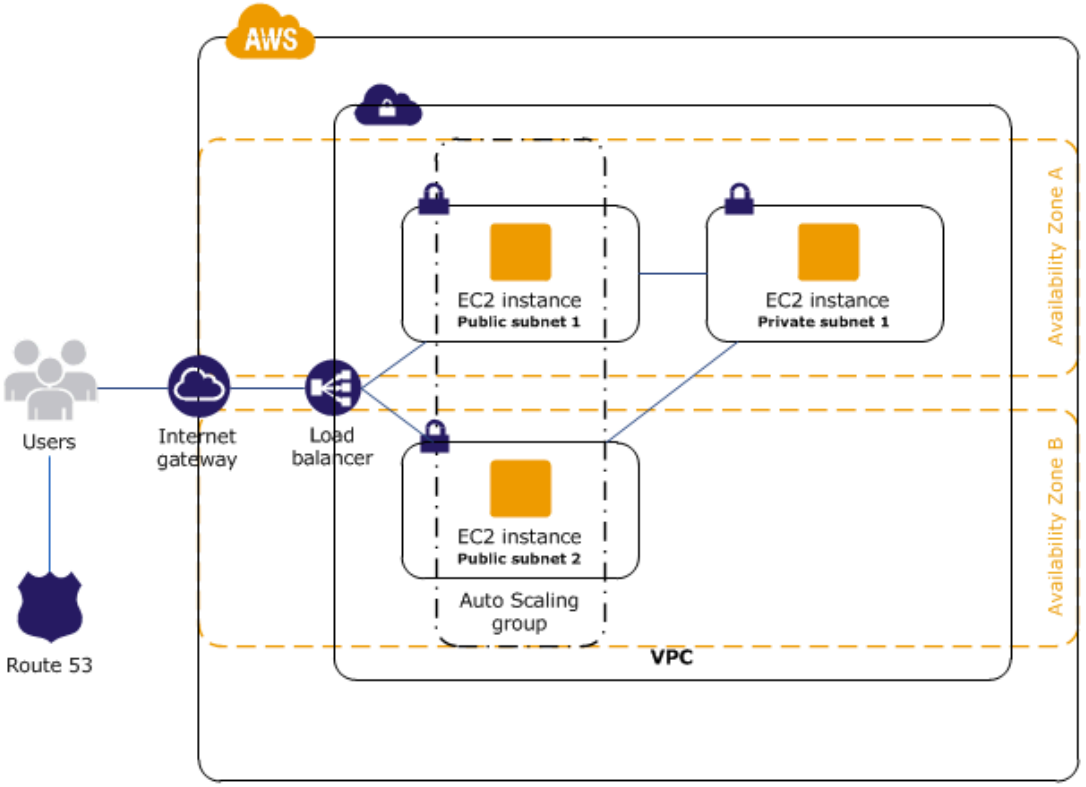
- Network Layer security with ACLs, Elastic Network Interfaces, use of internal elastic load balancer, and VPN connections

- EC2-Classic classic is a deprecated service by AWS. Some accounts that have been around for long periods of time are still using the service. However, EC2 classic instances do not belong to a VPC (can be a security issue) and have certain limitations. If your account was created after Dec 2013 EC2-classic is not part of your account. However, in this course we will have a lesson dedicated to EC2-Classic concepts.

Route 53

- Route 53 is a DNS hosting solution provided by AWS. You can not only host the DNS for domains but can now also register and transfer domains to AWS as the domain authority.
- Route 53 manages external DNS for domain routing www.domain.com to the proper AWS resources such as a CloudFront distribution, ELB, EC2 instance, or RDS server. (Not a comprehensive list)
- Route 53 can also be used to manage internal DNS for custom internal hostnames within a VPC as long as the VPC is configured for it.
- Latency, GEO, basic, and failover routing policies allow for region to region fault tolerant and architecture design.
- Failover to S3 or CloudFront (if website bucket hosting is enabled)

Architecture Example



Source: aws.amazon.com


Linux Academy
Amazon Web Services
Storage And Content Delivery

Amazon S3

- Simple Storage Service (S3) is a object storage service from AWS. It can not only serve objects through a CDN to CloudFront, manage access to specific objects, enable versioning, and lifecycle policies, but it can also serve static HTML files with Route 53. It is a simple key-value store designed for unlimited object storage.

- Designed for “11 nines” (99.999999999%) durability and 99.99% “availability”
  - Charges based off of per gig storage as well as data sent out of the region
  - Data transfer from S3 to an EC2 instance within the same region is free
  - Cost decreases as it scales

- S3 objects can be encrypted using the S3 encryption option as well as data sent to and from end points are encrypted using the HTTPS protocol.

- Most commonly used for file storage (is also a hybrid solution when used with AWS storage gateway), delivering static content, backups and archiving with Amazon Glacier.

- Bucket names are unique across the entire S3 design (all regions included)
- Eventual Consistency for the north Virginia (us-east-1, us standard) region.

Amazon S3

- RRS (Reduced Redundancy Storage)
  - Cost effective
  - Only for “easily reproducible data”
  - 99.99% durability vs “eleven nines” for standard storage type

- Lifecycle policies and object versioning
  - Pay for each version of the object
  - Unlimited versions
  - Needs to be enabled
  - Versioning and lifecycle polices can work together for an automated backup and archiving solution
  - Lifecycle policies with Amazon Glacier

Amazon Glacier

- Archival storage type
- Used for data not frequently accessed
- Check out and check in jobs can take several hours for the data to be changed
- Integrates with Amazon S3 lifecycle polices for easy archiving
- .01/gig per month

Amazon Storage Gateway

- Connects local data center software appliances to cloud based storage such as Amazon S3

- Gateway-Cached Volumes
  - Create storage volumes and mount them as iSCSI devices on the on-premise servers
  - The gateway will store the data written to this volume in Amazon S3 and will cache frequently access data on-premise in the storage device

- Gateway-Stored Volumes
  - Store all the data locally in storage volumes
  - Gateway will periodically take snapshots of the data as incremental backups and stores them on Amazon S3

Amazon Import/Export

- AWS Import/Export gives the ability to take on-premise data and physically snail mail it to AWS. AWS will import the data to either S3, EBS, or Glacier within one **business** day of the physical device arriving at AWS.

- Benefits:
  - Off-site backup policy
  - Quickly migrate LARGE amounts of data to the cloud
  - Disaster recovery (AWS will even take s3 data and ship it back to you)
  - Example


Linux Academy
Amazon Web Services
Databases

Amazon Relational Database Service (RDS)

- Relational databases are databases that organize stored data into tables. The associated tables have defined relationships between them.
- Amazon RDS is a **fully managed** database service for relational databases. This means that access to the underlying operating system is not allowed and software patches and management are handled by AWS.

- Databases Supported By RDS:
  - MySQL
  - PostgreSQL
  - Oracle
  - SQL (MS SQL Server)
  - Aurora

  - What is Aurora?
    - Home grown Relational Database forked, and ***fully compatible with MySQL***. It has fives times better performance then MySQL and a lower price point than commercial databases.

Amazon ElastiCache

- ElasticCache is a fully managed, in-memory cache engine. Available engines that power ElastiCache are Memcached and Redis. It is used to improve performance by caching results of queries, managing web sesssions, and caching dynamically generated data. Generally the application needs to be built to work with either Redis or Memcached

Amazon DynamoDB

- DynamoDB is a NoSQL fully managed database service provided by AWS. It is similar to MongoDB but is a home-grown AWS solution.

- Fully managed NoSQL service
  - Service manages all provisioning of underlying hardware
  - Fully distributed and scales automatically
  - Built as a fault tolerant highly available service
  - Primarily used by developers

- Specify required throughput capacity and DynamoDB does the rest
- Easily integrates with other services such as Elastic MapReduce

Amazon Redshift

- Amazon Redshift is a petabyte-scale data warehousing service.


Linux Academy
Amazon Web Services
Analytics

Amazon Elastic MapReduce (EMR)

- Elastic MapReduce is a Hadoop clustering tool that makes it easy to manage and integrate with Hadoop clusters. Hadoop is used for big-data analytics and through Elastic MapReduce. It can integrate easily with other services such as Redshift and DynamoDB for data analytics.

- EMR is a service that spins up EC2 instances which allows the user full access to the underlying operating system unlike RDS, DynamoDB and ElastiCache.


Linux Academy
Amazon Web Services
App Services

Amazon Simple Work Flow Service (SWF)

- Track work flow executions
- AWS control panel ability to monitor task work flow
- Consistent execution
- Scalable parallel EC2 processing
- Service can be used with on-premise servers
- Guarantees execution of work flow

Amazon Simple Queue Service (SQS)

- Decouple infrastructure systems
- Auto scale based off queue size
- Guarantees delivery of “at least” 1 message but does not guarantee no duplicates
- Scalable and highly available by design
- Image processing example
- Does not guarantee message order but does attempt “best effort” order delivery

Amazon Simple Notification Service (SNS)

- Coordinates and manages the delivery or sending of messages to specific end points. This service can be used for publishing IOS/Android app notifications, gluing together automation based off of sent notifications

- End points
  - SQS
  - Email
  - Email-json
  - SMS
  - HTTPS
  - HTTP
  - Application


Linux Academy
Amazon Web Services
Deployment Services

Amazon Elastic Beanstalk

- Easily deploy complete application environments automatically. Integrates with the Elastic Load Balancer, Auto Scaling, EC2, and additional AWS services which are all covered in this course.
- Key benefit for developers who are building applications but do not have the technical knowledge for building application environments.
- Includes basic configurations such as web applications and worker instances
- Support for Docker containers
- Easily deploy “dev/test/qa/production” environments with the EB command line tool which integrates with git repositories.

Amazon CloudFormation

- Everything in AWS is an API accessible through an SDK, Command Line Tools, or the console.
- CloudFormation is a tool that allows you to “code” your infrastructure and deploy resources based off of a pre-build template. This gives the advantage of easy backup and disaster recovery and even version controlling your AWS infrastructure!
- For example with a template and the template code you can build EC2 instances that belong to an Elastic Load Balancer and a Route 53 entry to your elastic load balancer all with code!
- Building CloudFormation templates are outside the scope of this certification but we go into great detail in the AWS Developer and other LinuxAcademy.com training material.


Linux Academy
Amazon Web Services
Management Services

Amazon Management Services

- IAM (Identity Access Management)
  - Web service that allows managing permissions to AWS resources
  - Can define resource level permissions and API call permissions based off user/group/resource
  - API keys and roles are used to manage access to AWS API to integrate with the SDKs
- CouldTrail
  - CloudTrail is an API logging service that logs ALL api calls made to AWS. It does not matter if the API calls from the command line, SDK, or console. This helps when addressing security concerns and even seeing what users on AWS performed certain access in your environment.
- CloudWatch
  - Used to monitor AWS services such as EC2.
  - Integrates very well with EC2 and helps provide centralized logging and performance metrics into instances such as CPU usage, Network Usage and more. (Details later in the course).
  - Auto Scaling is heavily used with CloudWatch. For example auto scale more worker EC2 instances if a queue size becomes to large.
- Directory Services
  - Allows the ability to easily connect on-premise Microsoft Active Directory with and AD connector. Also has the ability to setup and operate new directory's within the AWS cloud using simple AD.


Linux Academy
Amazon Web Services
IAM (Identity & Access Management)

What is IAM?

- IAM provides access and access permissions to AWS resources
- IAM is global to all AWS regions, creating a user account will apply to all the regions
- IAM policies allow for granular API level permissions for granting users and groups access to specific AWS resources

- Benefits:
  - Central control of AWS resources
  - Consolidated AWS bill for your users
  - Ensure users access only from specified networks
  - Easily manage security credentials
  - Provides temporary user access when needed
  - Federate with SAML providers such as active directory for temporary and single sign on access
  - Provide roles that other AWS resources can assume

What is IAM?

- Allows you to manage users and groups within the AWS account
- Can specify password policy as well as MFA requirements on a per user basis
- Provides pre-built policy templates to assign to users and groups
    - Administrator access
    - Power user access – Does not allow user/group management
    - Read only access- Only view AWS resources (accounting)

Users and Groups

- Groups
  - Assign permission policies to more than one user at a time
- Users
  - Best practice to work as an IAM user NOT as the root user (user@email.com)
  - Receive unique access credentials and do not share with others
  - User credentials should never be stored or “passed” to an EC2 instance
  - Users can have group and regular user policies apply to them
    - By default an explicit deny always overrides and an allow
    - By default a user has a non explicit “deny” on all AWS services and does not have access to use them until a policy granting allow access has been applied to the user account or to the group the user belongs to.

What is IAM?

- Roles
  - You “can” but **should never** pass or store credentials in or to an EC2 instance
  - Instances should be granted a role from IAM with the proper required permissions
  - Instances can perform actions based on the role it assumes
  - Other users can assume a “role” for temporary access
  - Can create “cross account” access where a user from one account can assume a role with permissions in another account


Linux Academy
Amazon Web Services
S3 (Simple Storage Service)

Amazon Simple Storage Service (S3)

- Objects are static files that contain metadata information
  - Set of name-key pairs
  - Cannot be changed after uploaded
  - Contains information specified by the user but also AWS information such as storage type
- Buckets contain a grouping of information and have sub name spaces that are similar to folders
- Each bucket must have a unique name across AWS s3
- File and Object storage
  - Unlimited Storage
  - High availability by design
  - “11 nines” (99.999999999%) Durability
  - 99.99% Availability
  - Objects can be as small as 1Byte and as Large as 5TB

Amazon Simple Storage Service (S3)

- Bucket Limitations
  - Only 100 buckets can be created in an aws account at a time
  - Bucket ownership cannot be transferred once a bucket is created

Amazon Simple Storage Service (S3)

- Standard storage is designed for “11 nines” (99.999999999%) durability and 99.99% availability

- RRS (Reduced Redundancy) 99.99% durability instead of 11 nines
  - Only use with data that is easily replaceable
  - 99.99% availability, same as standard storage

- Glacier is used for Archiving storage for .01/gig and integrates with S3 Lifecycle Policies
  - Used for infrequently accessed data
  - Can take several hours to check in and out data from Glacier

Amazon Simple Storage Service (S3)

- Versioning
  - Stores all versions of an object including deleted/write versions
  - Versioning and lifecycle policies can both be enabled on a bucket at the same time

- Lifecycle Policies
  - Lifecycle policies allow you to define the life of an object and specify to either send the object to Glacier for archival or delete the object permanently
  - Glacier is an archiving storage class on S3
    - Only used for infrequently accessed files
    - Checkout times for data can take hours

Amazon Simple Storage Service (S3)

- Amazon S3 Security
  - Permissions
    - All buckets and objects are private by default
    - ACL's(Can share accounts across accounts with ACLs)
    - IAM policies to grant IAM user access to resources
    - Bucket policies
      - Used for permissions such as granting access to an anonymous User
      - Restricting access based off of IP address
      - Restricting access based off of HTTP Referrer
    - Public content (Downloadable via URL)
    - Signed URLs (Developer/CloudFront)

- SSL terminated endpoints for the API
  - SSE (Server Side Encryption) – S3 can encrypt the object before saving it on the partitions in the data centers and decrypt it when it is downloaded
  - Can use your own encryption keys – Considered client side encryption where you encrypt the data before upload

Amazon Simple Storage Service (S3)

- Using Amazon S3 When Architecting Applications
  - S3 can be used for hosting static websites when used with Route 53
  - S3 can act as an origin to the CloudFront CDN
  - Multipart upload
    - Allows for uploading part of a file concurrently
    - Allows for stopping/resuming file uploads
    - **Required** for objects 5GB and large but suggested use is for objects 100MB and larger
    - Use with SDK/Command Line Interface

Amazon Simple Storage Service (S3)

- Using Amazon S3 When Architecting Applications
  - Object stay within an AWS region and are synced across all AZ's
  - Eventual Consistency
    - Us-east-1 (US Standard Region) for all put/write/read/delete requests
  - Read after write for new objects
    - All regions except US standard (us-east-1)
    - Means the object can be immediately available after "putting" object on S3

Amazon Simple Storage Service (S3)

- When to use S3
  - Hosting static files
  - Origin for CloudFront CDN
  - Hosting static websites
  - File shares for networks
  - Backup/Archiving
    - AWS Storage Gateway

Amazon Simple Storage Service (S3)

- Event Notifications can be sent when specific events/actions occur against a bucket
  - Event types include
    - RRSObjectLost (Used for automating the recreation of lost RRS objects)
    - ObjectCreated (* for all or the following specific APIs called)
      - Put
      - Post
      - Copy
      - CompleteMultiPartUpload
  - Events can be sent to
    - SNS
    - Lamda
    - SQS Queue


Linux Academy
Amazon Web Services
EC2 (Elastic Compute Cloud)

Amazon EC2 (Elastic Compute Cloud)

- Virtual Servers In The Cloud
  - EC2 is the instance computing platform for servers in the cloud. This service makes up several different features such as Elastic Load Balancer, Auto Scaling, EBS Volumes, and EC2 instances.

  - A deprecated version of Amazon EC2 is known as EC2-classic and is no longer available on new accounts created after December 2013.

  - A lesson will be dedicated to Amazon EC2-classic please be sure not to skip it

EC2 Instances

- Instance Types
  - Describe the type of compute, network, memory, and virtualization architecture that an EC2 instance will run on

  - As an architect it's important to understand the proper instance type to handle your application load as well as build your architecture with elasticity using auto scaling

  - T2 "Burstable Performance Instances"
  - M3 Instances (Nice Balance)
  - C4 (Compute Optimized)
  - R2 (Memory Optimized)
  - G2 (GPU Optimized)
  - I2 (Storage Optimized)
  - EBS Optimized

  - Note: EC2 instance sizes will vary in the amount of network capacity and other limitations such as number of Elastic Network Interfaces (ENI) able to be attached to an instance

EC2 Instances

- AWS can phase out old generation instances
- Amazon SLA (Service Level Agreement) http://aws.amazon.com/ec2/sla/

EC2 Instances

- Instance Storage
  - EC2 instances can be created with two “types” of storage
  - Instance-store volumes
    - Instance store volumes are considered ephemeral data, the data on the volumes only exists for the duration of the instance life
    - Once the instance is “stopped” or “shutdown” the data is erased
    - The instance can be rebooted and still maintain its ephemeral data
    - Instance-store volumes are virtual devices whose underlying hardware is physically attached to the host computer for the instance
    - AMI

  - EBS backed volumes
    - EBS backed volumes are network attached storage
    - Provide persistent data across EC2 instances even if they are shutdown
    - AMI

EC2 EBS Volumes

- EBS Volumes measure input/output operations in IOPS
- IOPS are input/output operations per second
- AWS measures IOPS as 256KB or smaller
- Operations that are greater than 256KB are separated into 256KB units
- A 512KB operation would count as 2 IOPS
- The type of EBS volume you specify greatly influences the I/O performance or IOPS your device will receive. It is important as architects to understand if our application requires more I/O to the EBS volume.
- Even volumes with provisioned IOPS may not produce the performance you expect. If this is the case an EBS optimized instance is required which prioritizes EBS traffic over the network OR an instance with higher network traffic capacity.

EBS Volume Types

- General Purpose SSD
  - Commonly used as the “root” volume on a system
  - Use on dev/test environments and smaller DB instances
  - 3 IOPS/GiB (burstable with baseline performance)
  - Volume size of 1GiB to 16TiB
  - Considerations when using T2 instances with SSD root volumes (burstable vs. baseline performance)
- Provisioned IOPS
  - Mission critical applications that require sustained IOPS performance
  - Large database workloads
  - Volume size of 4GiB to 16TiB
  - Performs at provisioned level and can provision up to 20,000 IOPS
- Magnetic
  - Low storage cost
  - Workloads where performance is not important or data is infrequently accessed
  - Volume size of 1GiB to 16TiB

Note: Pre-warming Volumes

EBS Snapshots

- Pay attention, has proven to be a tough concept for some students

- Snapshots are incremental in nature
  - A snapshot only stores the changes since the most recent snapshot thus reducing costs and only having to pay for storage for the “incremental changes” between snapshots
  - What happens when the original snapshot is deleted?
    - The data is still available, snapshot storage might only charge you as an incremental snapshot but the prior data is still there
    - Think about it like this you have “snapshots” point in time but the actual source file is dynamically growing. If you delete old snapshots the data in the source location still exists.

- Frequent snapshots of your data increases data durability

- When a snapshot is being taken against the EBS volume it can degrade performance so snapshots should occur during non-production or non-peak load hours

EC2

- Starting, stopping, and terminating instances
- IAM users with proper permissions
- While an instance is stopped you are not paying for compute time only for storage
- Termination and termination protection
- AWS Command Line Interface / SDK
- User-Data/Cloud-init

- Access an instances user-data and meta-data by opening this URL WITHIN the instance
  - http://169.254.169.254/latest/meta-data or http://169.254.169.254/latest/user-data
    - More examples in the course
    - Information such as user-data and ami-launch-index if launched as a group
    - Use to register an instance-id as part of a cluster or application suite AUTOMATION!

Elastic Load Balancer

- Can load balance traffic across availability zones to instances for high availability and fault tolerance
- When used within a VPC it can act as an internal load balancer and load balance to internal EC2 instances on private subnets often used with multi-tier applications
- Automatically stops serving traffic to an instance that becomes unhealthy
- Reduce compute power and apply to multiple instances by configuring the SSL certificate directly on the elastic load balancer

EC2-Classic

- YES! It's still important
- Instances are assigned a public IP address and cname
- Each instance receives a private IP address but is NOT part of a VPC
- This means if an instance is shut down It will lose its private IP and will change on next boot up

Security Groups

- Security groups are used as a firewall in front of an EC2 instance
- An instance can belong to multiple security groups
- Security groups can reference themselves as “source” traffic in firewall rules


Linux Academy
Amazon Web Services
Security Groups

Security Groups

- Up to 100 security groups per VPC
- Each security group can have up to 50 rules
- Assign up to 5 security groups per EC2 instance
- If an instance needs more rules an instance can belong to multiple security groups
- By default all rules are denied unless specifically allowed, you cannot create deny rules
- Responses to inbound traffic are allowed REGARDLESS of outbound rules (stateful) same applies to outbound traffic
- Instances associated with a security group cannot communicate with each other unless the ports are open to those instances, one exception is the default security group
- VPC security groups: once you launch an instance YOU CAN change the security group of the instance
- EC2-classic: Once you launch an instance YOU CAN NOT change the security group of the instance


Linux Academy
Amazon Web Services
Monitoring EC2

Status Checks

- Note: More important for SysOps Certification
- System Status Checks
  - Loss of network connectivity
  - Loss of system power
  - Software issues on the physical host
  - Hardware issues on the physical host
  - How to solve: Generally starting and stopping the instance so that it launches on a new physical hardware device will resolve the issue

- Instance Status Checks
  - Failed system status checks
  - Misconfigured networking or startup configuration
  - Exhausted memory
  - Corrupted file system
  - Incompatible kernel
  - How to solve: Generally a reboot or solving the file system configuration issue.

CloudWatch Alarms

- Key component for monitoring AWS infrastructure
- By default CloudWatch will automatically monitor metrics that can be viewed at the host level NOT the software level
- Detailed vs. Basic level monitoring
  - Basic: Data is available automatically in 5-minute periods at no charge
  - Detailed: Data is available in 1-minute periods
- OS level metrics that required a third party script to be installed (provided by AWS)
  - Memory utilization, memory used, and memory available
  - Disk Swap utilization
  - Disk space utilization, disk space used, disk space available


Linux Academy
Amazon Web Services
EC2 Placement Groups

Placement Groups

- A placement group is a cluster of instances within the same availability zones
- Instances within a placement group have a low-latency, 10 Gbps network connections between them.
- Used for instances that run applications whose requirements are an extremely low latency network between them.
- Instances that are in the placement group need to have enhanced networking in order to maximize placement groups.

Troubleshooting Placement Groups

- If an instance in a placement group is stopped once it is started again it will continue to be a member of the placement group

- It is suggested to launch all the required instances within a placement group in a single request and that the same instance type is used for all instances within the placement group
  - AWS attempts to place all the instances as close as physically possible to reduce latency

- It is possible, if more instances are added at a later time to the placement group OR if a placement group instance is stopped and started again, to receive an “insufficient capacity error”.

  - Resolve the capacity error by stopping all instances in the member group and attempting to start them again.

Troubleshooting Placement Groups

- Placement group keys
  - Instances not originally launched/created in the placement group cannot be moved into the placement group
  - Placement groups cannot be merged together
  - A placement group cannot span multiple availability zones
  - Placement group names must be unique within your own AWS account
  - Placement groups can be “connected”
  - Instances must have 10 gigabit network speeds in order to take advantage of placement groups


Linux Academy
Amazon Web Services
Relational Database Service (RDS)

RDS

- RDS is a fully managed Relational Database Service in the cloud
  - Does not allow access to the underlying operating system
  - Can connect to the database server itself as normal (i.e MySQL command line)
  - Ability to provision/resize hardware on demand for scaling
  - Multi-AZ deployments
  - Read Replicas (MySQL/PostgreSQL/Aurora)

- Currently supported Database engines
  - MySQL
  - PostgreSQL
  - Oracle
  - Microsft SQL Server
  - Aurora

- Ability to provision/resize hardware on demand for scaling

RDS

- Instances
- Disk space Minimum 5GB | Maximum 3TB
- SSD vs. Provisioned IOPS
- Benefits of running RDS instead of your own instance
  - Automatic minor updates
  - Automatic backups
  - Not required to managed operating system
  - Multi-AZ with a single click
  - Automatic recovery in event of a failover

RDS

- Automatic AZ Failover, Multi-AZ synchronous replicates data to the backup instance located in another availability zone
  - Availability zone outage
  - Primary DB instance fails
  - Instance server type is changed
  - Manual failover initiated
  - Updating software version
  - Backups are taken against the stand-by instance to reduce I/O freezes and slow down IF multi-az is enabled

RDS

- Backups: AWS provides automated point in time backups against the RDS database instance
  - Automated backups are deleted once the database instance is deleted and cannot be recovered
  - Backups on database engines only work correctly when the database engine is “transactional” but do currently work for all supported database types
  - MySQL requires InnoDB for reliable backups

RDS

- Read Replicas
  - MySQL/PostgreSQL/Aurora currently support
  - Uses native replication on by MySQL/PostgreSQL
  - Read replicas can be created from other read replicas
  - Multiple read replicas can have the same source
  - Read replicas allow for elasticity in RDS
  - Monitor replication lag using CloudWatch
  - Only supports InnoDB MySQL storage engine
  - Offload database tasks off of production
  - Can promote a read replica to a primary instance
  - MySQL
    - Replicate for importing/exporting data to RDS
    - Can replicate across regions

RDS

- Read Replicas: When to use them
  - High non-cached database read traffic (elasticity)
  - Running business function such as data warehousing
  - Importing/Exporting data into RDS
  - Rebuilding indexes
    - Ability to promote a read replica to a primary instance

RDS

- RDS CloudWatch/Notifications
  - Subscribe to be notified when specific events take place
    - Snapshots
    - Parameter group changes
    - Option changes
    - Security group changes
  - Integrates with CloudWatch
    - CPU Utilization –Freeable Memory – Swap Usage
    - Database Connections and binary log disk usage
    - Read/Write IOPS
    - Read Replicate latency log
    - Read/Write throughput


Linux Academy
Amazon Web Services
Virtual Private Cloud (VPC)

VPC

What is Virtual Private Cloud?

"*Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own datacenter, with the benefits of using the scalable infrastructure of AWS" – Amazon Web Services*

A VPC Resembles On-Premise:

- Private data centers
- Private corporate network

Private Network

- Private and Public subnets
- Scalable architecture
- Ability to extend corporate/on-premise network to the cloud as if it was part of your network (VPN)

Linux Academy

VPC

Benefits of a Virtual Private Cloud

- Ability to launch instances into a subnet
- Ability to define custom CIDR (IP address range) inside each subnet
- Ability to configure route tables between subnets
- Ability to configure internet gateways and attach them to subnets
- Ability to create a layered network of resources
- More security settings to protect cloud assets
- Extend your network into the cloud with VPN/VPG and an IPsec VPN tunnel
- Layered Security
  - Instance Security Groups
  - Subnet network ACLs (Essentially a firewall for incoming packets on the subnet level)

VPC

Understanding The Default VPC

- Default VPC is a different setup than a non-default VPCs
- Default VPC is meant to allow the user easy access to a VPC without having to configure it from scratch
- In the Default VPC all provided with the Default VPC have an internet gateway attached
- Each instance added has a default private and public IP address (defined on the subnet settings), remember public IP addresses are attached/routed to an ENI that has a private IP address attached to an instance (NAT).

VPC

VPC Peering

- VPC Peering enables the ability to create a direct network route between one VPC and another. This allows the sharing of resources between two subnets as if it was on the same network. Basically, at a high level it creates a link between the two.

- VPC Peering can occur between other AWS accounts and other VPCs within the same region

- VPC Peering connections cannot occur between two regions

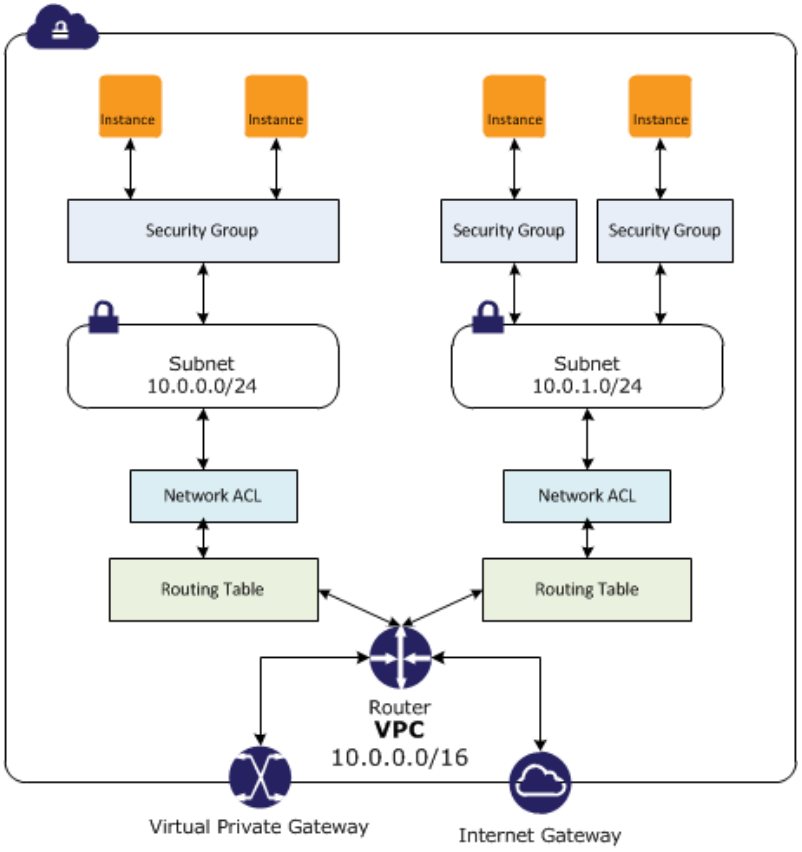
- Scenarios
  - Peering two VPCs – Company runs multiple AWS accounts and you need to link all the resources as if they were all under one private network (assuming resources in the same region)
  - Peering To A VPC – Multiple VPCs can connect to a central VPC but cannot communicate with each other, only communication can occur between the peered VPC and the primary. This use case could be if a third party was sharing a resource that the customers needed to connect to. (File sharing, Customer Access, Active Directory)

VPC

VPC Limits

- 5 VPCs per region (more available upon request)
- 5 internet gateways (this is equal to your VPC limit because you only have on internet gateway per VPC)
- 50 customer gateways per region
- 50 VPN connections per region
- 200 route tables per region / 50 entries per route table
- 5 elastic IP addresses
- 100 security groups
- 50 rules per security group
- Security groups per network interface (remember security groups are on the VPC level)

VPC



- Blocking traffic (subnet or SG)


Linux Academy
Amazon Web Services
VPC Networking

VPC

VPC Networking

- Each subnet must be associated with a route table
- By default all subnets traffic is allowed to each other available subnet within your VPC which is called the local route
- You cannot modify the local route
- Best practice is to leave the default route and create a new route table when new routes are needed for specific subnets

VPC

VPC Networking

- Internet gateway provides NAT translation for instances that have a public IP addresses assigned (public IP to private IP)

- *"To enable access to or from the internet for instances in a VPC subnet, you must attach an Internet gateway to your VPC, ensure that your subnet's route table points to the Internet Gateway**, ensure that instances in your subnet have a public IP address or Elastic IP address**, and ensure that your network access control and security group rules allow the relevant traffic to and from your instance" – AWS*

- Instances launched into a private subnet can't communicate with the internet. This is a higher level of security but it creates the limitation of the instance not being able to download software and software updates. In order to solve this issue you can create a NAT instance.

- NAT instance
  - Must be created in a public subnet

  - Must be part of the private subnets route table

VPC

VPC Networking

- AWS provides a DNS server for your VPC so each instance has a hostname. However, you can run your own DNS servers by changing the DHCP option set configuration within the VPC.


Linux Academy
Amazon Web Services
VPC Security

VPC

VPC Security

- Security Groups
  - Operate at the instance layer
  - Supports allow rules only
  - Is stateful so return traffic requests are allowed regardless of rules
  - Evaluates all rules before deciding to allow traffic
- Network ACL
  - Operates at the network/subnet level
  - Supports allow AND deny rules
  - Stateless so return traffic must be allowed through an outbound rule
  - Process rules in number order when deciding whether to allow traffic if it is denied at a lower rule number and allowed at a higher rule number the allow will be ignored and the traffic will be denied
  - Applies at the network level so it applies to all instances located inside of the subnet one deny will block all traffic on the port denied to all instances

VPC

VPC Security

- Network ACLs
  - Process rules in order starting with the lowest number
  - Deny all is the last rule (by default it is all denied) unless it is all allowed
  - Best practice to increment numbers by 10 so if you have to place in a rule in a certain order it does not create an issue

Inbound				
Rule #	**Source IP**	**Protocol**	**Port**	**Allow/Deny**
100	0.0.0.0/0	All	All	ALLOW
*	0.0.0.0/0	All	All	DENY
Outbound				
Rule #	**Dest IP**	**Protocol**	**Port**	**Allow/Deny**
100	0.0.0.0/0	all	all	ALLOW
*	0.0.0.0/0	all	all	DENY


Linux Academy
Amazon Web Services
VPN/VPG Concepts

VPN/VPG on VPC

VPN (Virtual Private Network): Virtual private network enables the ability to extend a subnet from one geographic location to another geographic location on two separate networks. Extending the subnets allows the network at location A to communicate internally with all resources at location B. This is essentially “extending” the on-premise network to the cloud or the cloud to the on-premise network. For AWS this allows us to communicate with all resources internally without the need for public/private IP addresses or an internet gateway. This provides an additional level of security and ensures traffic to and from the VPN is encrypted.

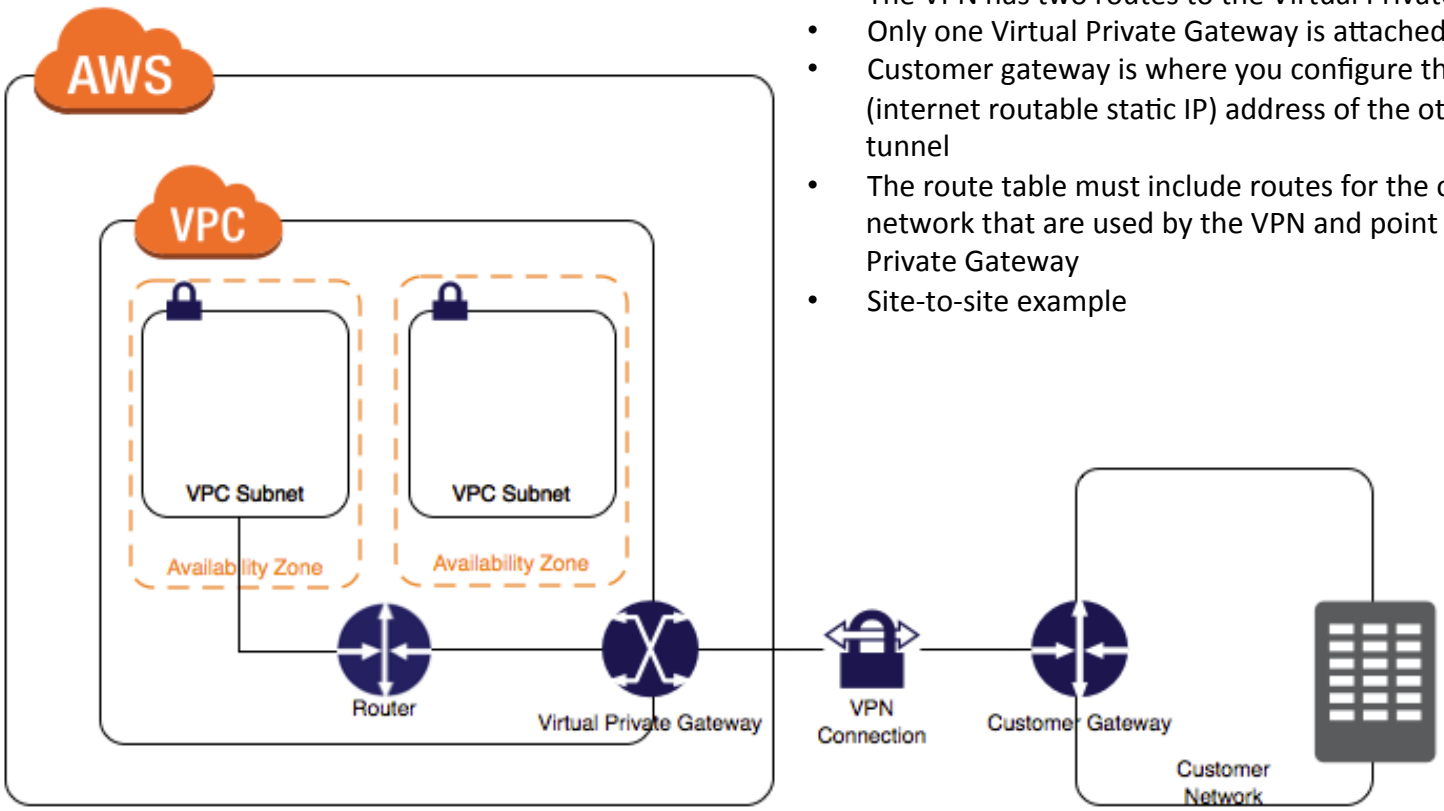
VPG (Virtual Private Gateway): A Virtual Private Gateway acts as the “connector” on the VPC side of the VPN connection. The VPG is connected to the VPC and the VPN is associated with the customer gateway creating the endpoint.

Customer Gateway: A customer gateway acts as the “connector” on the on-premise side of the VPN connection. This is where you configure the public IP address for the on-premise network.

Note: Both a VPG and a Customer Gateway are required to establish a VPN connection.

Linux Academy

VPN/VPG on VPC



- The VPN has two routes to the Virtual Private Gateway
- Only one Virtual Private Gateway is attached to a VPC
- Customer gateway is where you configure the public IP (internet routable static IP) address of the other end of the tunnel
- The route table must include routes for the on-premise network that are used by the VPN and point them to the Virtual Private Gateway
- Site-to-site example

Source: aws.amazon.com

Alternative Methods For Deploying A VPN to AWS

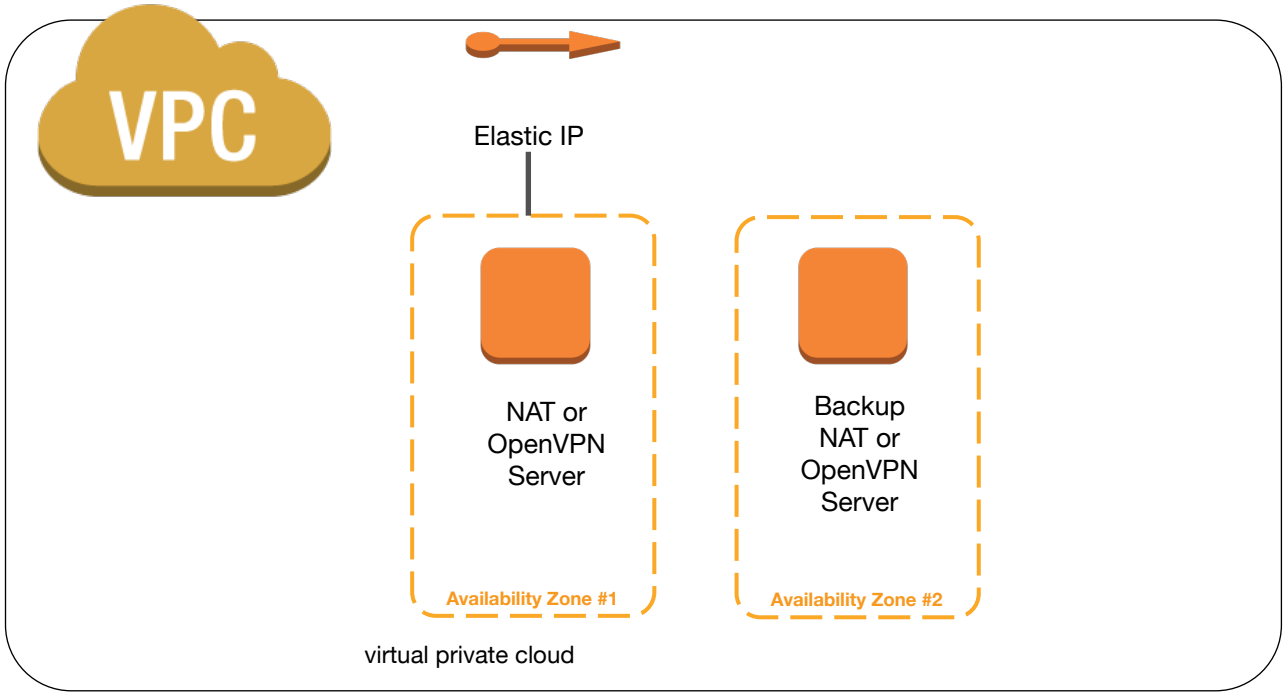
- Configure an OpenVPN instance (Not site-to-site but a VPN tunnel)
    - Lives in a public subnet and you connect to the public IP using the OpenVPN client
    - The OpenVPN client from your machine receives a new IP address from the OpenVPN server
    - The IP only exists on the OpenVPN server but the VPN routes that IP address range to the subnets
    - OpenVPN instance use a route table that lives on the OpenVPN server that is able to route our private OpenVPN client users from the OpenVPN server to the AWS VPC address range

- How to Apply High Availability To OpenVPN Single Instance
    - Perfect example of single instance “fail over”
    - OpenVPN has an elastic IP address
    - A backup OpenVPN instance is running in an alternative AZ
    - Script logic that determines if the OpenVPN is unavailable
    - If the OpenVPN instance is unavailable using a script and AWS CLI detach the Elastic IP and Reattach it to the backup instance

Source: aws.amazon.com

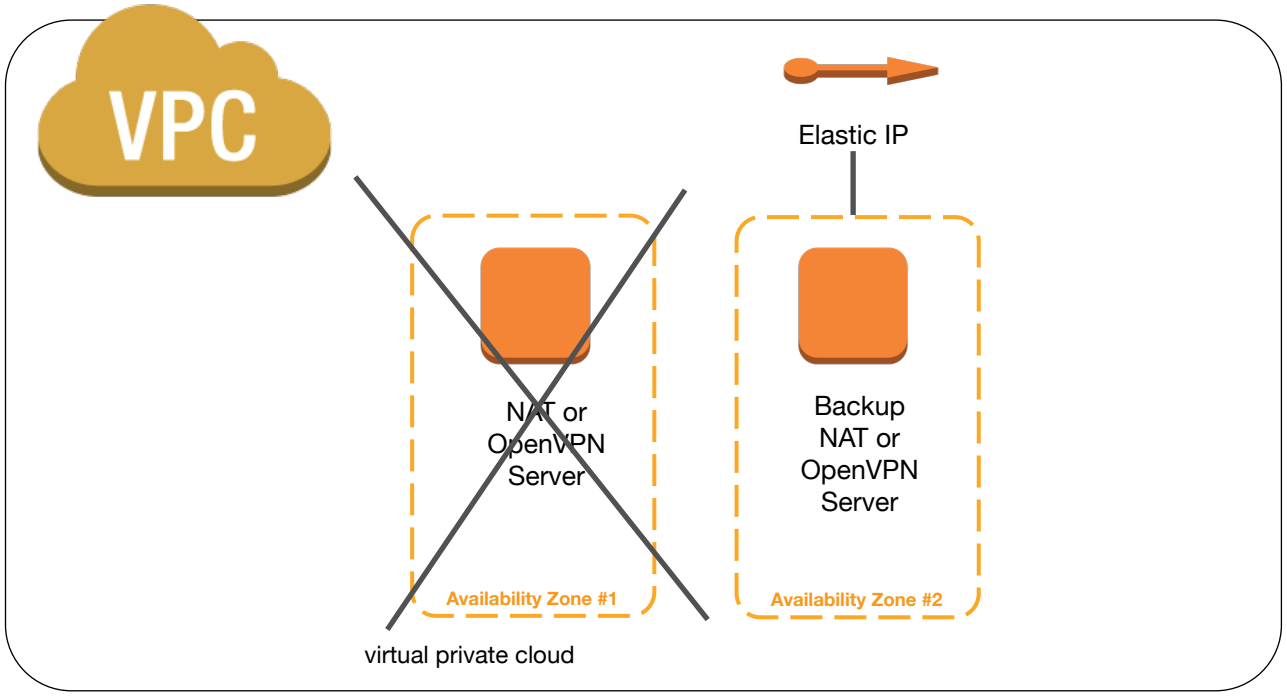
Linux Academy

Single Instance Fail Over (OpenVPN/NAT Example)



Source: aws.amazon.com

Single Instance Fail Over (OpenVPN/NAT Example)



Source: aws.amazon.com


Linux Academy
Amazon Web Services
CloudFront Essentials

CloudFront Essentials

CloudFront

- CloudFront is a global CDN which delivers content from an “origin” location to an “edge” location.
- An origin can be an S3 bucket or an Elastic Load Balancer CNAME that distributes requests among origin instances
- Signed URLs allow access to “private content” by creating a temporary one time use URL based off of the number of seconds it is suppose to be available
- CloudFront can integrate with Route 53 for “alternate” cnames. This allows you to create a URL such as http://cdn.mydomain.com that works with your distribution.
- CloudFront is designed for caching, if caching is enabled then CloudFront will serve the cached file stored on the edge location until cache expires.
  - In order to serve a new version of the object either create a new object with a new name or create an “invalidation” on the CloudFront distribution based off the object name
  - Invalidations cost so if you have to invalidate an entire large CloudFront distribution then perhaps you should just create a new distribution and move DNS names


Linux Academy
Amazon Web Services
CloudFront Performance Considerations

CloudFront Performance Considerations

CloudFront

- File size and type of file (HLS Streaming Chunks)
    - Having to remake the request from the EDGE location to the origin will decrease performance because the edge location has to download the object from the origin as well as write it to cache as it is responding to the end user request
- The end location the user request goes to is dependent upon a DNS check to determine the closest EDGE location so slow DNS issues can cause performance issues
- Longer cache periods increases performance
- Load testing
    - Send client requests from multiple geographic regions
    - Configure test so each client makes independent DNS request

CloudFront Essentials

CloudFront

- Query strings reduce cache “hits”
- http://cdn.linuxacademy.com/?querythis=querythat
- It reduces performance because query strings are often unique so it reduces the cache hits and also requires extra “work” in order to forward to the origin location.

- The more requests that have to go to the origin the higher the load is on your source which can also cause latency and load performance issues.


Linux Academy
Amazon Web Services
SNS (Simple Notification Service)

Linux Academy

Simple Notification Service

SNS: SNS is integrated into many AWS services. We are able to use it to receive notifications when events occur in our AWS Environment. With CloudWatch and SNS a full environment monitoring solution could be created that notifies administrators of alerts, capacity issues, downtime, changes in the environment, and more!

- TOPIC: A topic what a “message is sent to”
- Subscription end point: SNS sends all messages to subscriptions subscribed to a specific topic
- Subscriber end points include the following
  - Application, Mobile APP notifications (IOS/Android/Amazon/Microsoft)
  - SMS
  - HTTPS
  - JSON
  - E-Mail JSON
  - SQS Queue


Linux Academy
Amazon Web Services
Amazon SQS (Simple Queue Service)

SQS

SQS: Hosted/highly available queues used for messages being sent between computers. SQS allows the creation of distributed/decoupled application components.

- Each message can contain up to 256KB of text in any format
- Amazon SQS guarantees delivery of each message at least once BUT DOES NOT guarantee the order (best effort) in which they are delivered to the queue, in other words it does not guarantee first in first out order. SQS is also highly available and redundant managed service by design
- Used for created decoupled application environments
- SQS does not guarantee there will be no duplicate messages sent to the queue
- Generally a “worker” instance will “poll” a queue to retrieve waiting messages for processing
- Auto Scaling can be applied based off of queue size so if that component of your application has an increase in demand worker instances can increase

SQS

SQS

- Two types of polling
  - Long Polling
    - Allows the SQS service to wait until a message is available in a queue before sending a response (reduces API requests) and will return all messages from all SQS services
  - Short Polling
    - SQS samples a subset of servers and returns messages from just those servers.
    - Will not return all possible messages in a poll
    - Increases api requests which increases costs

Want to know more? SQS is covered in more detail in the
AWS Certified Developer – Associate Level Course By The Linux Academy


Linux Academy
Amazon Web Services
Decoupled/Distributed Architectures

Tightly Coupled System – A system architecture of components that are not just linked together but are also dependent upon each other

- If one component fails all components fail

Loosely Coupled/Decoupled Systems – Multiple components that can process information without being connected

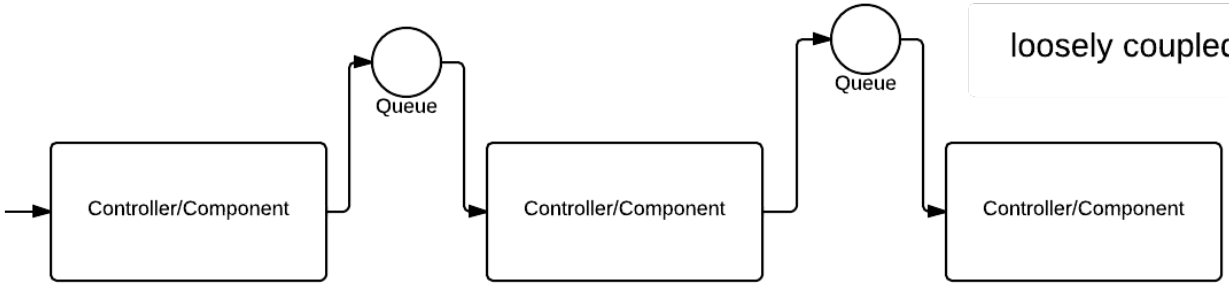
- Components are not connected if one fails the rest of the system can continue processing – fault tolerant/highly available.

AWS Services that are used for distributed/decoupled system architectures

- SWF (Simple Work Flow Service)
- SQS (Simple Queue Service)

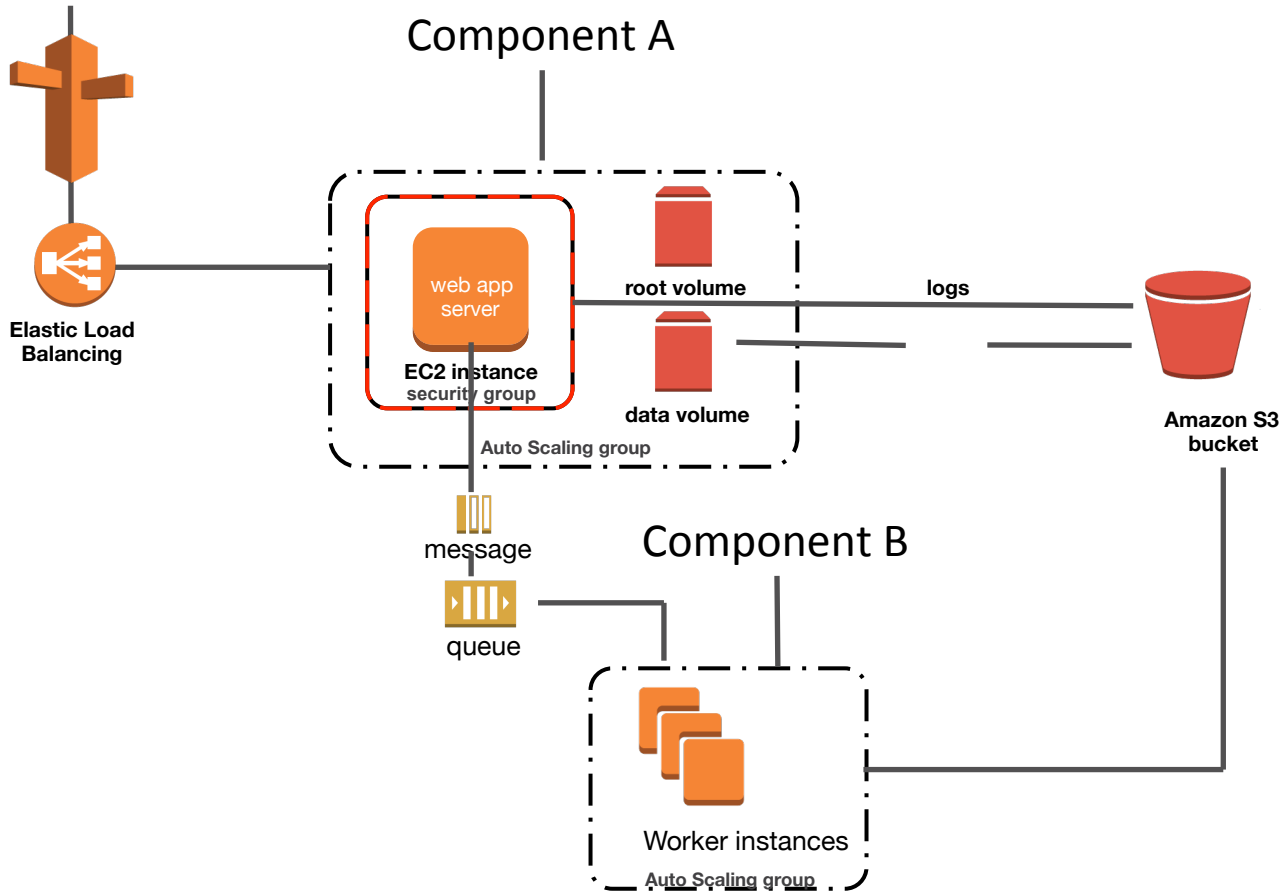
Tightly Coupled System

loosely coupled system



Linux Academy

www.changemyimage.com




Linux Academy
Amazon Web Services
Amazon SWF

SWF

SWF: SWF is a fully managed “work flow” service by AWS. Work flows allows an architect/developer to implement distributed, asynchronous applications as work flows.

What is a work flow? A work flow coordinates and manages the execution of activities that can be run asynchronously across multiple computing devices.

- Tasks are what interacts with the “workers” that are part of a work flow
  - Activity task – Tells the worker to perform a function
  - Decision task – Tells the decider the state of the work flow execution which allows the decider to determine the next activity to be performed

SWF

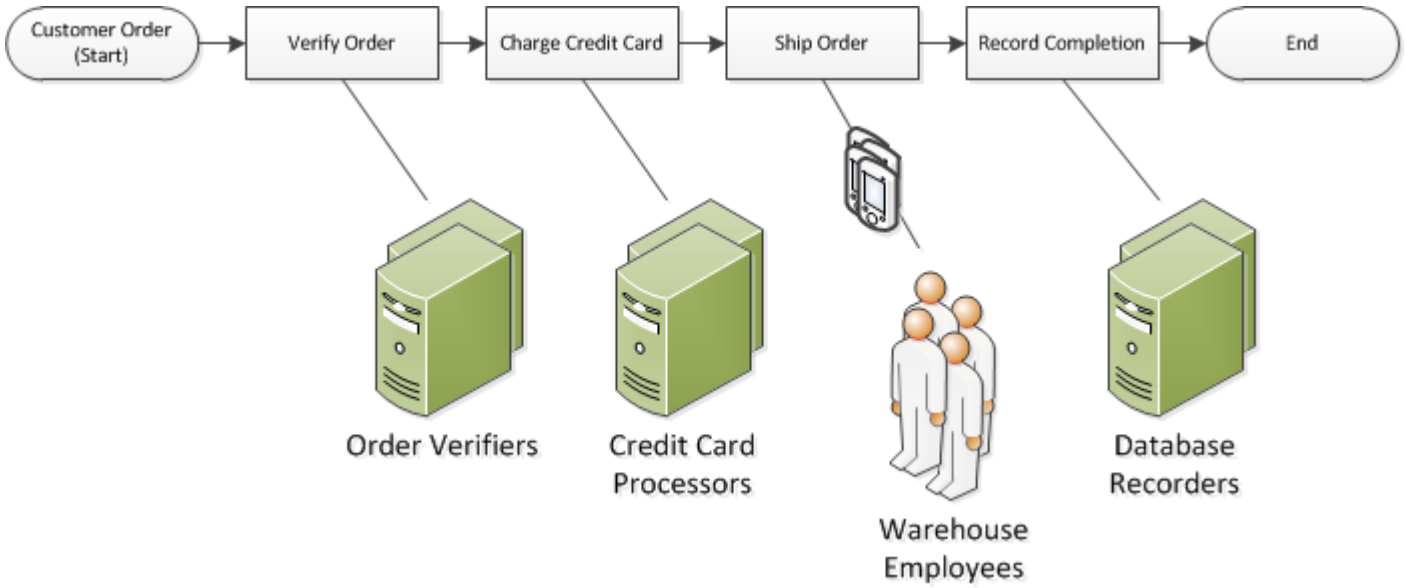
A worker and decider can be any type of component such as an EC2 instance or even a person!

SWF has consistent execution and guarantees order in which tasks are executed and that there are no duplicate tasks.

The SWF service is primarily an API which application can integration the work flow service into. This allows the service to be able to be used from even non AWS services such as an on-premise data center.

A workflow execution can last up to 1 year!

SWF



Source: https://aws.amazon.com


Linux Academy
Amazon Web Services
EMR (Elastic MapReduce)

Amazon EMR

Amazon EMR is a service which deploys out EC2 instances based off of the Hadoop big data software. EMR is used to analyze and process vast amounts of data.

The data which is mapped to a cluster of Hadoop Master/Slave nodes for processing, allows for computations to be performed by the servers (computations coded/created by the developer) and reduces to a single output set of return information.

EMR can run any OpenSource application built on Hadoop

Amazon EMR

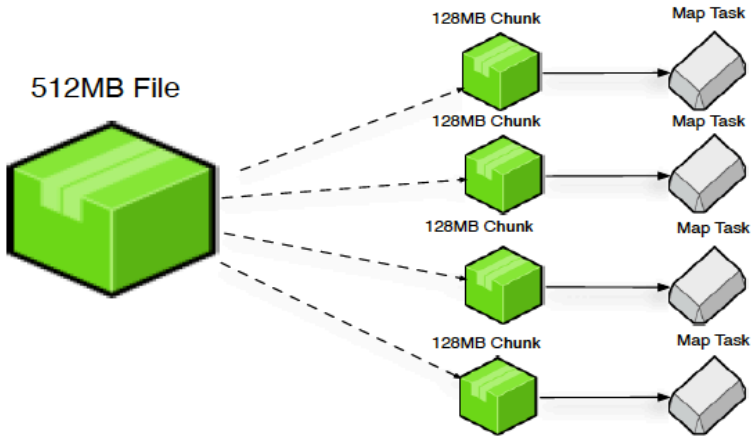
EMR launches EC2 instances for processing the data passed to Hadoop

- Gives the admin ability to access the underlying operating system
- Integrates in with AWS services such as S3, DynamoDB, RedShift (Petabyte Scale Data Warehouse) to receive and output data
- Bootstrapper enables the ability to pass configuration information before Hadoop starts when a new cluster is created

Amazon EMR

Mappers are essentially the processes that split the large data file for processing
Reducers takes the result (if there is) and combines it back into a data file on the disk

Word counter:
Mapper: The text being processed looking for the key word
Reducer: The outputted number result found (5 results for the word being searched for)
and combined back into the single result data file



Mapper tasks are loaded into memory

Amazon EMR

AWS EMR has already pre-configured the Hadoop AMI to have the best possible performance based off of instance size.

Amazon EC2 Instance Name	Mappers	Reducers
m1.small	2	1
m1.medium	2	1
m1.large	4	2
m1.xlarge	8	4
c1.medium	4	2
c1.xlarge	8	4
m2.xlarge	4	2
m2.2xlarge	8	4
m2.4xlarge	16	8
cg1.4xlarge	12	3

- A single mapper is configured by AWS to handle 128MB “split” files (this is the default on certain versions of Amazon AMI Hadoop and can be changed)
- To Increase performance you can adjust the “split” size of the file note that changing the split size might require more mappers to be started
- Split files are loaded into memory for processing
- If there are 5 128MB split files waiting for mapping then AWS will queue the ones not being processed increasing the amount of time for a single job to complete
- Cluster can launch multiple instances for parallel processing reducing the amount of time a job takes to complete by processing multiple split files with the mapper at one time

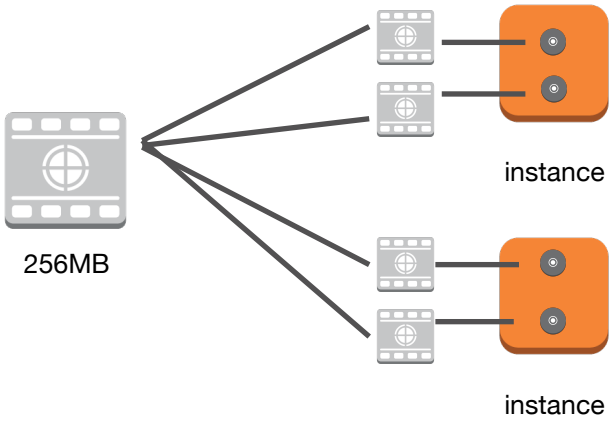
Amazon EMR

Goal is to use as many mappers as you can without running out of memory for loading of data

If your job is queuing up it can increase job run time here are some considerations:

- Determine if you have unused mappers, if you have unused mappers then consider changing the input split size
- If you do not have unused mappers speed up the process by adding more mappers (adding more core/task nodes to the cluster)
- Both of these methods would be considered as increasing the number of available mappers

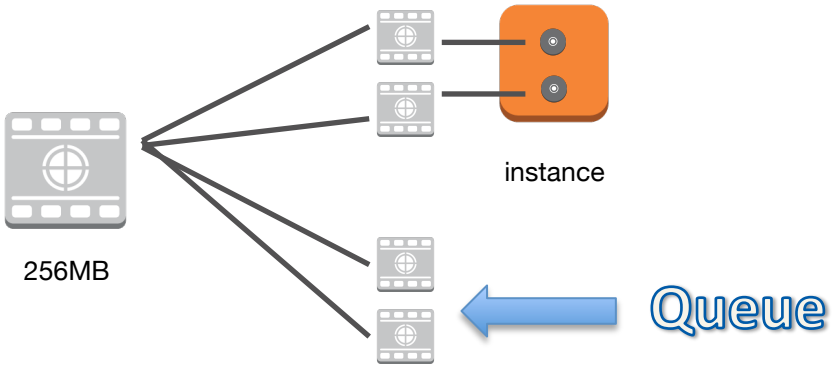
Amazon EMR



Chunk files 64MB in size

All available mappers being used

Amazon EMR



- Split files are waiting to be mapped
- Job processing time increases
- Solve by adding mappers to the cluster
  - Launch new cluster nodes
  - Change instance size

Chunk files 64MB in size

Amazon EMR Cluster Architecture

Workflow:

1. Load the the data into the cluster (Can come from S3) also can load S3 as a local file system (increases performance and decreases job loading time)
2. Analyze the data (done by programming/scripts)
3. Store the results in the Hadoop Distributed File System (HDFS) or S3 (Faster/persistent)
4. Read the results from the cluster HDFS

Master Node: Only provides information about the data (where it is coming from) and where it should go
Core Node: Processes the data and stores it on HDFS or S3
Task Node: Processes the data and sends back to the Core node for storage on the HDFS or S3

The HDFS stores the “reduced” result on the local file system which means the data is only persistent for as long as the cluster is running. Once terminated the data is lost

Storing the “reduced” data on S3 helps create persistent storage and reduces the amount of time it takes to copy the data from the HDFS storage disks to the original location (S3).

Amazon EMR Cluster Architecture

Bootstrap Action: Can be used to install additional software or change configurations before Hadoop starts on a new EMR cluster

Performance increases when running S3 as a local file system as it directly connects to S3 and enables multi-part uploading for faster uploading processing of data to and from S3. By default the S3 file system is already installed on all nodes in a cluster

- Bootstrap action can be used to enable multi-part upload
- Using S3 for the primary file system with multi-part upload enabled decreases the job run time and provides persistence for your cluster
- S3 increases durability of reduced data

Amazon EMR Cluster Architecture

How could EMR be used within an application environment? Process log files!


Linux Academy
Amazon Web Services
Elastic Beantstalk

Elastic Beanstalk

Elastic Beanstalk is designed to make it easy to deploy out less complex applications. This helps reduce the management required for building and deploying applications. Elastic Beanstalk is used to deploy out easy single tier applications that take advantage of core services such as EC2, Auto Scaling, ELB, RDS, SQS, and CloudFront.

The clear choice on “when” to use Elastic Beanstalk is:

- In order to quickly provision an AWS environment with little to no management requirements
- The application fits within the parameters of the Beanstalk service
- Can deploy from repositories or from uploaded code files
- Easily update applications by uploading new code files or requesting a pull from a repository

Supported Platforms

- Docker
- Java
- Windows .NET
- Node.js
- PHP
- Python
- Ruby


Linux Academy
Amazon Web Services
Amazon CloudFormation

Linux Academy

What is CloudFormation?

- CloudFormation allows you to create and provision resources in a reusable template fashion
- CloudFormation allows you to source control your infrastructure by building templates which allow you to “script” the ability to delete/create resources on demand
- CloudFormation templates are built using JSON syntax
- CloudFormation allows you to launch a single collection of resources together that are defined within the CloudFormation template


Linux Academy
Amazon Web Services
How To Design Cloud Services

How To Design Cloud Services

Take advantage of Multi-Availability Zone architectures

- Always design with instances in at least two availability zones, the more the better

Purchase reserved instances in disaster recovery availability zones

- Guarantees you have “reserved” capacity in the event of an emergency
- No, AWS does not guarantee on-demand instance capacity

Use Route 53 to implement failover DNS techniques

- Latency based routing
- Failover DNS routing

Have a disaster recovery and backup strategy that utilizes:

- Multiple Regions
- Maintain up to date AMI’s
- Copying AMI’s from one region to another
- Copying EBS snapshots to other regions
- CRON jobs that take snapshots of EBS volumes and store on S3

Linux Academy

How To Design Cloud Services

Design for failure

Rigorously test to find single points of failure and apply high availability (might require custom solutions)

Automate everything in order to easily re-deploy resources in the event of a disaster

Apply elasticity when available

Utilize Elastic IP addresses to fail over to “stand-by” instances when auto scaling and load balancing are not available

Decouple application components using services such as SQS when available

How To Design Cloud Services

"Throw away" old or broken instances

Utilize bootstrapping to quickly bring up new instances with minimal configuration and allows for "generic" AMI's

- You can tell an instance what "role" it plays at boot time

Utilize CloudWatch to monitor infrastructure changes and health

Always enable RDS Multi-AZ and automated backups (InnoDB table support only for MySQL)

Create Self Healing application environments

Utilize MultiPartUpload for S3 uploads

Cache static content on Amazon CloudFront using EC2 or S3 Origins

Linux Academy

How To Design Cloud Services

- Protect your data in transit
  - Use HTTPS/SSL Endpoints
  - Connect to instances inside of the VPC using a bastion host or VPN connection
- Protect data at rest using encrypted file systems or EBS/S3 encryption options
- Take advantage of AWS Key Management service
- Use centralized sign on for on-premise users and apply to EC2 instances and IAM logins
- Use IAM roles on EC2 instances instead of using API keys; Never store API keys on an AMI


Linux Academy
Amazon Web Services
Monitoring

Monitoring With CloudWatch

Use CloudWatch For:

- Shutting down inactive instances
- Monitoring changes in your AWS environment with CloudTrail integration
- Monitor Instances resources and create alarms based off of usage and availability

    - EC2 instances have “basic” monitoring which CloudWatch supports out of the box and includes all metrics that can be monitored at the hypervisor

    - CloudWatch for Status Checks which can automate the recovery of failed status checks by stopping and starting the instance again

    - EC2 metrics that include custom scripts to work with CloudWatch
        - Disk Usage; Available Disk Space
        - Swap Usage; Available Swap
        - Memory Usage; Available Memory

Linux Academy

Monitoring With AWS Config

AWS Config is a service which provides detailed configuration information about an environment

- Take a point in time “snapshot” of all supported AWS resources to determine the state of your environment
- View historical configurations within your environment by viewing snapshots
- Receive notifications whenever resources are created, modified, or deleted
- View relationships between resources, I.E what EC2 instances an EBS volume is attached to

Linux Academy

Monitoring With AWS CloudTrail

AWS CloudTrail is great for security and compliance and monitors all actions taken against the AWS account which CloudTrail enabled.

- Monitor and be notified of changes to IAM accounts with CloudWatch/SNS Integration
- View what API Keys/User performed any given API action against an environment I.E view what user terminated a set of instances or instance
- Can be used in order to meet auditing requirements inside of organizations


Linux Academy
Amazon Web Services
Architecture Trade-Off Decisions

Architecture Trade-Off Decisions

Storage Trade Off Options

- S3 Standard Storage
  - %99.999999999 durability and %99.99 availability but is the most expensive
- S3 RRS
  - Reduce redundancy at %99.99 but the storage costs are cheaper
  - Should be used for easily reproducible data that we can take advantage of lost object notification from AWS S3
- Glacier
  - Requires hours to check in and check out data from archiving
  - Costs are significantly reduced compared to S3 storage options

Architecture Trade-Off Decisions

Database Trade Off Options

- Running Databases on EC2
  - Have to manage the underlying operating system
  - Have to build for high availability
  - Have to apply your own backups
  - Can use additional software to cluster MySQL
  - Requires more time to manage than RDS

- Managed RDS database provides:
  - Fully managed database updates and does not require managing of the underlying OS
  - Provides automatic point in time backups
  - Easily enable Multi-AZ failover, when a failover occurs the DNS is switched from the primary instance to the standby instance
  - If Multi-AZ is enabled then backups are taken against the stand-by to reduce I/O freezes and updates are applied to the standby then is switched to the primary
  - Easily create read replicas


Linux Academy
Amazon Web Services
Elasticity and Scalability

Elasticity and Scalability

Proactive Cycle Scaling: Scaling that occurs at a fixed interval

Proactive Event-based scaling: Scaling that occurs in anticipation of an event

Auto-scaling based on demand: Scaling that occurs based off of increase in demand for the application

Plan to scale out rather than up (Horizontal scaling)

- Add more EC2 instances to handle increases in capacity rather than increasing instance size
- Be sure to design for the proper instance size to start
- Use tools like Auto Scaling and ELB
- A scaled service should be fault tolerant and operationally efficient
- Scalable service should become more cost effective as it grows

Elasticity and Scalability

DynamoDB is a fully managed NoSQL services from AWS. With high availability and scaling already built in. All the developer has to do is specify required throughput for the tables.

RDS requires scaling in a few different ways. RDS does not support a cluster of instances to load balance traffic across. Because of this there are a few different methods to scale traffic with RDS

- Utilize read replicas to offload heavy read only traffic
- Increase the instance size
- Utilize ElastiCache clusters for caching database session information


Linux Academy
Amazon Web Services
Data Security
Security Architecture with AWS


Linux Academy
Amazon Web Services
Shared Security Responsibility Model

What is the Shared Security Responsibility Model?

- AWS is responsible for portions of the cloud and you as the customer have portions of the cloud that you are responsible for thus giving you a shared security responsibility.

- Reduces the operational burden as AWS operates, manages, and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the services operate.

- You as the customer using AWS assumes the responsibility and management of the guest operating system (including, updates and security patches), other associated applications software, as well as the configuration of the AWS-provided security group firewall. You are also responsible for your own coded applications and custom applications built on top of the cloud.

AWS for EC2 is responsible for:

- Facilities
- Physical security of hardware
- Network infrastructure
- Virtualization infrastructure

You as the customer are responsible for:

- Amazon Machine Images (AMIs)
- Operating systems
- Applications
- Data in transit
- Data at rest
- Data stores
- Credentials
- Policies and configuration


Linux Academy
Amazon Web Services
AWS Platform Compliance

The AWS cloud infrastructure has been architected to be flexible and secure using world-class protection Using it's built-in security features:

- **Secure access –** Use API endpoints, HTTPS, and SSL/TLS
- **Built-in firewalls –** Virtual Private Cloud (VPC)
- **Unique users –** AWS Identity and Access Management (IAM)
- **Multi-factor authentication (MFA)**
- **Private subnets –** AWS allowing private subnets on your VPC
- **Encrypted data storage –** Encrypt your data in EBS, S3, Glacier, Redshift, and SQL RDS
- **Dedicated connection option –** AWS Direct Connect
- **Perfect Forward Secrecy –** ELB and CloudFront offer SSL/TLS cipher suites for PFS
- **Security logs –** AWS CloudTrail
- **Asset identification and configuration –** AWS Config
- **Centralized key management –** Centralized key management service
- **Isolated GovCloud –** US ITAR regulations using AWS GovCloud
- **CloudHSM –** Hardware Security Model (HSM) hardware based cryptographic storage
- **Trusted Advisor** – With premier support


Linux Academy
Amazon Web Services
Incorporating Common Conventional
Security Products

OS side Firewalls

- IPTABLES, FirewallD, Windows Firewall

AntiVirus Software

- TrendMicro
  - Integrates in to AWS EC2 instances


Linux Academy
Amazon Web Services
Design Patterns


Linux Academy


Linux Academy
Amazon Web Services
DDOS Mitigation

When mitigating against DOS/DDOS attacks use the same practice you would use on your **on-premise** components when establishing your Cloud presence:

- Firewalls:
  - Security groups
  - network access control lists
  - host-based firewalls
- Web application Firewalls (WAFS)
- Host-based or inline IDS/IPS (Trend Micro)
- Traffic shaping/rate limiting

Along with your traditional approaches for DOS/DDOS attack mitigation AWS cloud provides capabilities based on its elasticity.

- You can potentially use CloudFront to absorb DOS/DDOS flooding attacks. A potential attackers trying to attack content behind a CloudFront is likely to send most requests to CloudFront edge locations, where the AWS infrastructure would absorb the extra requests with minimal to no impact on the back-end customer web servers.

- Must have permission to do Port Scanning
- INGRESS Filtering on all incoming


Linux Academy
Amazon Web Services
Encryption solutions

- S3 has built-in features that allow you to encrypt your data
  - AES-256 bit encryption that encrypts data at REST
  - It is decrypted at it is sent to the customer at download

- EBS encrypted volumes
  - The data is encrypted on the EC2 instance and coped to EBS for storage
  - If a snap-shot it taken that snap-shot is automatically encrypted

- RDS encryption
  - MySQL, Oracle, PostegreSQL, MS SQL all support this feature
  - Encrypts the underlying storage space for the instance
  - Automated Backups encrypted as well as snap-shots
  - Read Replicas are encrypted
  - Provides SSL to encrypt a connection to a DB instance


Linux Academy
Amazon Web Services
Complex Access Controls

- IAM polices with resource level permissions
  - EC2 – create permissions for instances such as reboot, start, stop, or terminate based all the way down to the instance ID
  - EBS volumes – Attach, Delete, Detach
  - EC2 actions that are not one of these above are not governed by resource-level at this time
- This is not EC2 limited can also include services such as RDS, S3, etc

- Require additional security measures, such as MFA authentication, when acting on certain resources.
    - For example, you can require MFA before an API request to delete an object within an S3 bucket.


Linux Academy
Amazon Web Services
CloudWatch For The Security Architect

CloudWatch Security

- Request are signed with HMAC-SHA1 signature calculated from the request and the user's private key
- CloudWatch control API is only accessible via SSL encrypted endpoints
- CloudWatch access is given rights via IAM essentially giving users permissions that are only needed.
- Use CloudWatch and CloudTrail to monitor changes inside the AWS environment


Linux Academy
Amazon Web Services
Disaster Recovery

Disaster Recovery

Business disaster recovery key words: Very important for AWS CSA PRO

Recovery time objective (RTO): Time it takes after disruption to restore operations back to its regular service level as defined by the companies operational level agreement. I.E If the RTO is 4 hours you have 4 hours to restore back to acceptable service level

Recovery point objective (RPO): Acceptable amount of data loss measured in time. I.E if the system goes down at 10PM and RPO is 2 hours then the recovery should recover all data as part of the application as it was before 8PM.

Many AWS services can be used for designing disaster recovery solutions. Not only should you design for disaster recovery for your current application if it is running on AWS, you can also use AWS as a disaster recovery solution for your on-premise applications or data. The services used should be determined based off of the business RTO and RPO operational agreement.

Disaster Recovery

Pilot Light: Minimal version of your production environment is running on AWS. Replication from on-premise servers to AWS, in the event of a disaster the AWS environment spins up more capacity (elasticity/automatically) and a DNS switch from on-premise to AWS is made. It is important to keep up to date AMI and instance configurations.

Warm Standby: Has a larger foot print than a pilot light setup and would most likely be running business critical applications in "standby". This type of configuration could also be used as a test area for applications.

Multi-Site Solution: Essentially clones your "production" environment which can either be in the cloud or on premise. Has an active-active configuration which means instances size and capacity are all running in full standby and can easily convert at the flip of a switch. Methods like this could also be used to "load balance" using latency based routing or Route 53 failover in the event of an issue.

Services Examples:

- Elastic Load Balancer and Auto Scaling
- Amazon EC2 VM Import Connector
- AMI's with up to date configurations
- Replication from on-premise database servers to RDS
- Automate the increasing of resources in the event of a disaster
- Use AWS Import/Export to copy large amounts of data to speed up replication times (also used for off site archiving)
- Route 53 DNS Failover/Latency Based Routing Solutions
- Storage Gateway (Gateway-cached volumes/Gateway-stored volumes)


Linux Academy
Amazon Web Services
Troubleshooting EC2

Common Troubleshooting Thought Processes

Connectivity issues to an EC2 instance

- Ports on the correct security group are not open. Remember, all ports are closed by default and only the default security group has ports open to all instances in the same security group (can create this on your own by referencing the SG id as a source)

Cannot attach and EBS volume to an EC2 instance

- EBS volumes must live in the same availability zone as the EC2 instance they are to be attached to
- Create a snapshot from the volume and launch the volume in the correct availability zone

Cannot launch additional instances

- Reached EC2 capacity limit and need to contact AWS to increase limit

Unable to download package updates

- EC2 instance does not have a public/Elastic IP address and/or does not belong to a public subnet

Applications seeming to slow down on T2 micro instances

- T2 micro instances utilize CPU credits so changes are your application is using to much processing power and needs a larger instance or different instance type

Elastic IP address detaches from an instance when the instance is stopped

- EC2-classic does not use the VPC so when an EC2 instance is shut down the private IP is unassigned which is what the EIP routes too

AMI unavailable in other regions

- AMI's are only available in the regions that they are created.
- An AMI can be copied to another region but will receive a new AMI id

Capacity error when attempting to launch an instance in a placement group

- Start and stop all the instances in the placement group (AWS tries to locate them as close as possible)


Linux Academy
Amazon Web Services
VPC Troubleshooting Scenarios

Common Troubleshooting Thought Processes

New EC2 instances are not being assigned a public IP address automatically
- Modify Auto-Assign Public IP setting on the subnet

Successful site-to-site VPN connection but unable to access extended resources
- Need to add on-premise routes to the Virtual Private Gateway route table

NAT instance configured but instances inside a private subnet still cannot download packages
- Need to add 0.0.0.0/0 route to the i-xxxxx on the route table for private subnets

Failure to create a VPC peering connection between two VPC's in different regions
- Peering connections can only be created between two VPC's in the same region

Traffic is not making it to the instances even though security group rules are correct
- Check the Network Access Control Lists to ensure the proper ports from the proper sources are open

Error when attempting to attach multiple internet gateways to a VPC
- Only one internet gateway can be attached to a VPC at any given time

Error when attempting to attach multiple Virtual Private gateways to a VPC
- Only one Virtual Private Gateway is needed on a VPC

VPC Security group (for EC2 instances) does not have enough rules for the required application
- Assign the EC2 instance to multiple security groups

Cannot SSH/communicate with resources inside of a private subnet
- VPN is not setup or you have not connected to an EC2 instance within the VPC to launch connections from (think Bastion host)


Linux Academy
Amazon Web Services
ELB Troubleshooting Scenarios

Common Troubleshooting Thought Processes

Load balancing is not occurring between instances in multiple availability zones
- Enable Cross-Zone load balancing

Instances are healthy but are not registering as healthy with the ELB
- Check the status check and ensure the resource it is attempting to check is available or modify it to one that is

The ELB is configured to listen on port 80 but traffic is not making it to the instances that belong to the ELB
- Listeners are not the same as the security group rules, port 80 still needs to be open on the SG that the ELB is using

Access logs on web servers show IP address of the ELB not the source traffic
- Enable Access Logs to Amazon S3

Unable to add instances from a specific subnet to the ELB
- The specific subnet that instances will be launching in needs to be added to the ELB


Linux Academy
Amazon Web Services
Auto Scaling Troubleshooting Scenarios

Common Troubleshooting Thought Processes

An Auto Scaled instance continues to start and stop in short intervals
- Change the threshold of the scaling policy to be lower for decreasing the group or the intervals at which the policy is checked

Auto Scaling does not occur even though scaling policies are configured correctly
- Ensure that the “max” number of instances set to auto scale is not the same as the current EC2 instances running as part of the group

Please note: that there are more troubleshooting considerations that are outside the scope of the AWS CSA. However, they are covered in the AWS SysOps Certification and training course at the Linux Academy.


Linux Academy
Amazon Web Services
How To Prepare For The Exam

Linux Academy

How to Prepare For The Exam

- The exam grades you on your knowledge of “hands-on” experience in the AWS environment
- AWS is looking to see if you have the knowledge to choose specific services/architectures for different scenarios
- AWS is looking to know if you understand and know the technical details and configuration to deploy out those environments

1. Practice everything that is done hands on in the video.
2. There were some “conceptual” videos, take that as a hint of what you should understand conceptually.
3. Make sure to understand all the technical concepts as they are presented and performed in the hands-on video. These videos help gain the experience required in order to pass the exam.
4. Spend time investigating items such as
    1. Where items are configured in Auto Scaling/RDS
    2. How applications work together and trade off decisions
    3. Even what data is returned when displaying metadata from within an instance


Linux Academy
Amazon Web Services
What To Do After you're a CSA

Linux Academy

What To Do After You Pass

1. Share on the community
2. Populate your Linux Academy certification profile
3. Move onto Certified Developer or Certified SysOps
4. Move Onto AWS Certified Solutions Architect Professional Level
5. Do you know Linux?
6. Learn Deployment tools in our DevOps section