



Linux Academy

Amazon Web Services

Certified Solutions Architect – Professional Level



About this course

- I'm Anthony, Your CSA Professional Certified Instructor
- Only students who have received the AWS Certified Solutions Architect – Associate level are encouraged to take this course. Prior knowledge from the CSA Associate level course on LinuxAcademy.com is assumed.
- This course will include all knowledge required in order to prepare for the AWS CSA – Pro certification.



About this course

- This course will start off by focusing on “assumed” AWS and prior “general” IT knowledge for review. These concepts are extremely important in order to be able to understand and pass the AWS CSA Pro.
- The course will then move to cover individual services and how cost, design principles, performance, and security apply to each.
- Finally, this course will focus on scenario based training, and comprehensive application deployments.
- Due to the complexity and skill set required for the certification, please do not skip any of the training material.



About this course

- Labs will be provided when possible. At this point, if you are preparing for the AWS CSA PRO, it is assumed you have substantial prior IT experience, the ability to run Linux on local machine, and an AWS account. While labs are used when available to reiterate important concepts, they are not a replacement for practice on this level of training.
- DO NOT register for the CSA pro exam until the course is completed and you've completed the best practices for studying listed at the bottom of this course.



Linux Academy

Amazon Web Services

AWS Knowledge



Linux Academy

Amazon Web Services

Security features that AWS provides and best practices



Network Security Features

- Secure network access
 - AWS endpoints are secured with HTTPS/TLS for secure communication
- Built-in firewalls
 - Egress and ingress filtering of network traffic through VPC network ACLs
 - Instances utilize security groups as built-in firewalls
- Private subnets
 - Private subnets for isolated private resources
 - Ability to add IPSec VPN tunnel between on-premise and cloud VPC
- End-to-end encrypted transmission
 - Ability to add SSL/TLS endpoints on self-managed resources such as ELB
- Dedicated connection option
 - AWS Direct Connect provides a dedicated connection from on-premise to AWS. Both public and private IP access can be configured with AWS direct connect
- Advanced cipher suites
 - Available with services like ELB or CloudFront and also utilize Perfect Forward Secrecy to ensure data is not compromised even if the long term keys are



Access Control

- API request authentication
 - Every API request is digitally signed using cryptographic hash function and the API users secrete access key
- SSH access to instances
 - Access to Linux instances have password authentication disabled by default and require the use of RSA key pair for accessing the instance
- Unique users
 - IAM allows each AWS user to have unique setup, API keys, and password policy. This ensures that users do not need to share passwords to access AWS resources and easy to maintain log trail of who performs certain API calls
- Multi-factor authentication (MFA)
 - Available for root and IAM users when used with CloudTrail, CloudWatch, and SNS



Access Control

- Fine-grained permissions for S3 buckets and objects
 - ACLs to grant S3 bucket and object access to specific groups of users within other AWS accounts.
 - IAM is used to grant permissions to bucket or object access to users within the same AWS account
- Restricted viewer access to private CloudFront content
 - Geo-restriction allows CloudFront to restrict access to requests originating from certain IP addresses
 - Signed URLs create a temporary unique URL that expires at a specific time
- Temporary IAM security credentials
 - Grant temporary access to users and/or services that do not have normal AWS access.
Credentials last from 1 to 12 hours and cannot be reused after expiration



Monitoring and Logging

- Asset identification and configuration
 - AWS CloudConfig monitors AWS resource configuration and changes
 - Integrates with SNS to send notifications of resource changes
 - Does not support every AWS resource
- Security logs
 - Utilize CloudTrail service to monitor ALL api requests and the user/api keys that made the request
- Resource and application monitoring
 - CloudWatch integration with SNS allows for the monitoring of application logs on EC2 instances and the health of AWS resources
- Fine-grained access logging for S3 buckets
 - When configured, access logs for each object and access request will be provided
 - Logs include request type, requested resource, requester's IP, and time/date of the request



Monitoring and Logging

- Automated identification of security gaps
 - Trusted advisor is only for higher level accounts and is not available to all accounts
 - Trusted advisor provides insights such as:
 - Security
 - Testing of opened reports
 - Unrestricted access
 - S3 Bucket permissions
 - MFA on Root account
 - IAM password policy
 - RDS Security Group Access Risk
 - CloudTrail logging
 - Route 53 MX and SPF Resource Record Sets
 - ELB Listener Security
 - ELB Security groups



Backup and Replication

- EBS data backups
 - EBS backups are stored automatically in multiple physical locations to create redundancy. EBS (snapshots) data backups will be encrypted if the EBS volume is encrypted
- Automatic snapshots of Redshift data
 - Redshift snapshots are backed/stored by Amazon S3
- RDS database instance replication
 - Multi-AZ failover, when enabled, provides synchronous replication to a standby in another AZ
- Object versioning in S3
- Automated and continuous archiving to Glacier
- Protection from accidental deletion of S3 objects
 - Enabled S3 versioning MFA delete feature
 - Each version to be deleted must be verified with MFA
- Seamless, secure backups for on-premise data
 - AWS Storage Gateway



Data Encryption

- Encrypted data storage
 - The following services allow data to be encrypted: EBS, S3, Glacier, Redshift, SQLServer, and MySQL server
- Centralized key management
 - AWS Key Management Service provides a management feature for administrating keys for AWS services that utilize encryption at rest
- Dedicated, hardware-based crypto key storage
 - CloudHSM, higher security on dedicated key storage hardware.



Best Practices (high level)

- Keep the number of ports open on a security group limited and limit who can access them when available (for example limited for SSH port 22)
- Ensure that users are using IAM
- Utilize “least privilege” permission design and grant the least amount of privileges required
- Enforce password policy for IAM users
- Ensure RDS security groups are locked down and any data not being sent within the same region is utilizing HTTPS endpoints
- Enable CloudTrail logging in order to log all API calls and the accounts that make them
- Ensure proper ELB security permissions and take advantage of HTTPS/TLS when encryption is required
- As we progress on each service we'll have a different look at applying security



Best Practices (high level)

- Use IAM roles on EC2 instances
- Use policy conditions for extra security
- Rotate API keys no less than once a year



Linux Academy

Amazon Web Services

Design and implement for elasticity and scalability



Review from AWS CSA Associate

- Scalability allows the application to expand and scale to increase in demand with minimal effort
- Elasticity is the ability of an application to expand and contract based off of utilization requirements and needs
 - Time/proactive based scaling
 - Load/Performance based scaling



Review from AWS CSA Associate

- Scalable applications include the following characteristics:
 - Increasing AWS resources will result in a proportional increase in performance
 - A scalable service is capable of handling and working with many different vendor applications in the environment (heterogeneity)
 - Is operationally efficient
 - Is resilient
 - Should become more cost effective as it grows



Amazon S3 (Simple Storage Service)

S3 is scalable and elastic by design

Principles of elasticity and scalability include:

- Supports virtually unlimited number of files in any bucket
- Can store virtually unlimited number of bytes without partitioning or file system management
- S3 automatically manages the scaling and distribution of redundant copies of objects stored in S3
- S3 asynchronously replicates the information to all availability zones within a region
- S3 bandwidth can scale to virtually any load given which makes it perfect for integration with CloudFront, using as a static webhosting solution, and serving static objects.



Amazon Glacier

Amazon Glacier which is used for long term storage archive purposes is also has the following scalability and elasticity principles:

- Each archive can have up to 4TBs of data stored
- Unlimited amount of data that can be stored when used with multiple archives
- Amazon Glacier automatically scales based off of demand without need to provision more disk space



Amazon Elastic Block Store (EBS)

Amazon EBS is a block storage device that can be attached to single instances. EBS volumes are network-attached storage that persists independently of an EC2 instance. The following are scalability and elasticity principles of EBS volumes:

- Quickly provision additional capacity by adding new EBS volumes
- Resize an existing volume by creating a snapshot and launching a new volume from the snapshot
- Nice to know:
 - EBS volumes are redundantly replicated on different hardware within the same availability zone of a EBS volume



AWS Import/Export

Import/Export is a service that takes physical storage devices sent to AWS and imports them onto EBS volumes, Glacier Storage, or Amazon S3. The service is used to help data migrations from on-premise storage to the cloud. Elastic and scalability principles of Import/Export include:

- Upload unlimited amounts of data (Only limitation is the physical hardware sent to AWS)
- S3 file sizes can be up to 5 terabytes in size
- Glacier archives are limited to 4 terabytes in size



AWS Storage Gateway

AWS Storage Gateway connects on-premise hardware to AWS cloud based storage such as Amazon S3. It is used in disaster recovery as well as increasing the amount of available storage accessible on-premise. Elastic and scalability principles of Storage Gateway include:

- Gateway-cached/gateway-stored volume configurations allow for virtually unlimited files stored in Amazon S3



Amazon CloudFront

CloudFront is a Content Delivery Network (CDN) used for distributing cached static files from EDGE locations around the world. Elastic and scalability principles of CloudFront include:

- Easily grow the number of items in a CloudFront distributions that are being served by using Amazon S3 as an origin
- AWS EDGE locations are designed to handle increased connections automatically based off of demand
- CloudFront uses multiple layers of caching on EDGE locations to reduce the load on origin servers such as EC2 instances. This will allow for accepting a growing number of incoming connections without having to scale backend servers.



Amazon SQS (Simple Queue Service)

SQS is a hosted message queue service. Messages are produced within an application and used to “glue” together components of an infrastructure to create decoupled and fault tolerant components. Elastic and scalability principles of SQS include:

- Accepts virtually unlimited number of servers (EC2 instances or even on premise servers) writing/reading from a queue at any given time
- Allows for parallel processing of messages due to the ability of accepting read/write requests from unlimited number of VM's



Amazon RDS (Relational Database Service)

RDS is a hosted relational database service which provides access to the database server but not the underlying hosted operating system. Elastic and scalability principles of RDS include:

- Scale I/O performance by increasing the number of IOPS to the Database storage
- Scale by specifying the instance size which will change without downtime if Multi-az is enabled
- Utilize read replicas by offloading read only requests from the primary database to an asynchronously replicated read replica
- Advanced configurations include partitioning or sharding to distribute the workload over multiple database instances



Amazon ElastiCache

ElastiCache is a hosted Memcache or Redis caching engine that allows for in-memory cache of databases in the cloud. Elastic and scalability principles of ElastiCache include:

- Ability to add or delete nodes from a caching cluster on demand
- The more available nodes the more cache that can be stored



Amazon Redshift

Redshift is a fully managed petabyte-scale data warehouse that integrates with existing business intelligence tools. Elastic and scalability principles of Redshift include:

- Easily scale the number of nodes within the Redshift service
- Additional nodes can be added to the cluster as read only while the existing cluster is working



Linux Academy

Amazon Web Services Network Technologies



DNS

Domain Name System (DNS) serves as a directory of network hosts and resources. DNS resources can be public or private. Private resources rely only on local internal DNS servers to resolve on the local network only. Public DNS works with the directory of network hosts to provide domains such as linuxacademy.com.

Authoritative name servers are name servers that are responsible for assigning domain names to a specific IP address. Slave/caching name servers only exist to replicate information from Authoritative servers and rely on the domain record TTL to determine how often to update the cached name record.

A domain is made up of a hierarchy which are delineated by the . character. A domain represents a collection of resources that make up a subtree of the DNS name space i.e linuxacademy.com

The .com is considered the “top level” linuxacademy.com is considered the root of the domain and aws.linuxacademy.com is considered a “sub domain” of linuxacademy.com.



DNS

Authoritative name servers contain DNS records which maps the domain name to the IP address. Every domain name internal or public is mapped to an IP address. A “zone” is a record in which the name server is responsible for.

Within a zone resource records exists as basic information for the domain name system.

Common types of resource records:

A – Address record which is used to map hostnames (domain names) to IPv4 addresses

cname – Alias of one name to another (one hostname to another hostname)

AAAA – Address record which is used to map hostnames (domain names) to IPv6 addresses

NS – Name server record delegates a DNS zone to use the given authoritative name servers

MX – Mail exchange record which maps a domain name to a MTA (message/mail transfer agent)



DNS

Traditional DNS servers include the BIND DNS server and unbound. However, AWS provides a hosted DNS solution and options to integrate with external DNS servers as part of the VPC. The hosted solution is called Route 53 and is used as an authoritative name server for both public and internal DNS.

Examples of Route 53 usage as an authoritative DNS service:

- Host public domain names for external web applications
- Configure for failover, geo-based routing, weighted based routing, and latency based routing to resources
- Configure resource records for internal DNS hostnames

TTL (Time To Live) can be configured for each resource record within a zone. The TTL specifies how long that specific record should be cached by DNS resolvers.



DNS

Authoritative name servers provide information recording the mapping of hostnames/domain names to IP addresses. However, your instances need to have access to local DNS servers in order to lookup the resource records. In other words a configuration or service within your environment needs to know how to lookup what IP address a hostname should map to. You could also manually configure external DNS servers on each instance. However, configuring this for a VPC is much easier and scalable. An individual instance can be configured in /etc/resolv.conf or DNS settings can be configured specifically on the VPC. As part of AWS you can specify a new DNS server rather than using AWS built-in DNS for lookups. An EC2 instance automatically inherits its /etc/resolv.conf settings from the VPC configuration. If you want to use Route53 as an internal DNS provider you must maintain usage of the AWSDNS record in the VPC.

By specifying an on-premise DNS server that is connected over VPN to your VPC you can extend your internal DNS configuration into the cloud and add resource records to your internal EC2 instances. DNS servers that instances utilize inside of a VPC can be specified within the “DHCP options set” within VPC. The option set must then be associated to the VPC. Only one option set can be associated to a VPC at a time.



DNS

- example



Load Balancing

Load balancing is the process of distributing workloads across computing resources such as EC2 instances, VMs, or physical servers. Load balancing can be used in multi-tier application environments to serve internal data to multiple computing resources.

Within AWS the EC2 Elastic Load Balancer is used to distribute work loads across EC2 instances. It uses “round robin” load balancing.

Stickiness when applied to a load balancer determines if an existing session (cookie based or ELB based) is to go back to the specific instance they were on. Stateless webservers where sessions are managed by databases (DynamoDB is a good example) do not require this. This also has performance issues when scaling.



Load Balancing

To reduce CPU usage and additional configuration SSL/TLS certificates should always be configured on the Elastic Load Balancer. This way any instance associated with the ELB can utilize the SSL/TLS certificate over port 443.



Virtual Private Cloud

The VPC is used to create an isolated set of resources within the AWS cloud. VPC features allow for extending your private on-premise network to the cloud as well as having public resources available on the cloud. Resources utilize subnets which can either be private or public (internet gateway attached or not attached). Private subnets provide an additional layer of isolation and security.

To extend on-premise network to the cloud a VPG/VPN needs to be configured to an on-premise router such as cisco. You also have the option of using AWS Direct Connect for a more secure and efficient connection.



Virtual Private Cloud

VPCs within the same region can also be “peered” to each other to extend other account resources to your VPC. Service providers might extend a specific subnet to another customers account VPC within the same region. A service provider might require resources from another AWS account access instances from within their VPC. Peering the VPC creates a “private” extension of one VPC to another.

Scenarios for peering:

- One VPC Peered With Two VPCs
- One VPC Peered with Multiple VPCs
- Two VPCs Peered to Two Subnets in One VPC
- One VPC Peered to Specific Subnets in Two VPCs
- One VPC Peered With Two VPCs using Longest prefix match



AWS Direct Connect

Direct Connect creates a dedicated link from on-premise private networks to an AWS region. In order to use direct connect you must have on-premise servers located on a Direct Connect provider. Direct Connect uses a dedicated line to the AWS regions. 1gigabit or 10gigabit networking is required. Once established direct connect will work with VPN/VPG inside of the VPC in order to create the secure communication.

Direct connect has the following benefits:

- Increase bandwidth throughput to AWS
- More consistent network experience
- Uses industry standard 802.1q VLANs
- Use the same connection to access public resources and objects stored in Amazon S3 using the public IP address space.



Linux Academy

Amazon Web Services

Storage And Archival Options



Storage And Archival Options

Amazon S3

- RRS (99.99% durability)
- Standard storage (Eleven nines durability)
- Versioning
- Lifecycle policies
- Maximum 5TB files

Amazon Glacier

- 4TB archive limit but allows unlimited archives
- Works with S3 Lifecycle policies

Amazon Storage Gateway

- Gateway-cached
- Gateway-stored

Amazon EBS

- Single network attached volumes attached to one instance at a time
- Has redundancy built-in only for the same AWS availability zone
- To migrate an EBS volume to another region you must first create a snapshot and then copy the snapshot and create a volume from the copied snapshot
- You can encrypt a snapshot during the copy process even if the underlying volume is not encrypted



Linux Academy

Amazon Web Services

State Management

State Management

Maintaining the “state” of an application can be very important. For example when a user arrives at your website if a session is created on an instance is that session associated with the instance?

- Stickiness
- Database sessions
- DynamoDB (Popular solution for managing sessions for applications to maintain session state)

State Management

Maintaining the “state” and monitoring changes in your environment is also important.

AWS Config

AWS CloudTrail



Linux Academy

Amazon Web Services

Database And Replication Methodologies

Replication

Replication considerations

- Distance between locations
- Available bandwidth
- Data rate required by the application
- Replication technology

Types of replication:

- **Synchronous replication** - Automatically updated in multiple locations means good network consistency is important.
- **Asynchronous replication** - Data is transferred as network performance and availability allows. If throughput is down then replication will wait.



Database And Replication Methodologies

- Replication can be used to scale workloads with high I/O requests by offloading the reads to a read replica.
- If the “source” or “primary” DB becomes unavailable the read replica can act as a backup to serve traffic while the source DB is being repaired. Only read data is available during this time.
- Read replicas utilize built-in MySQL asynchronous replication technology.
- Multi-AZ failover utilizes synchronous replication.
- MySQL instances can launch read replicas in other regions to help assist with disaster recovery and making read requests closer to end users located close to another region.
- Always use SSL/TLS certificates on RDS when using cross region replication. The reason for this is because most of the data goes over the open internet.

Replication as a disaster recover or data migration mechanism

Replication with MySQL can be used to export data to an on-premise network

- Configure the RDS MySQL instance
- Configure the MySQL DB instance on RDS to be the replication source
- Use mysqldump and transfer the database from RDS to the on-premise MySQL
- Start replication to the instance running external to RDS (it is set as the slave)
- After the export is completed stop replication

Replication as a disaster recover or data migration mechanism

Replication for MySQL can also be configured from on-premise to RDS

- Set the source MySQL instance to read-only
- Determine the binlog location
- Use mysqldump to copy existing database to RDS
- Make the source writeable again
- Configure the security group to allow for your external IP address to communicate with the instance
- Create the MySQL replication user and grant permissions
- Configure the RDS instance to be a replica by using the `mysql.rds_set_external_master` command at the command line of the RDS instance
- Issue the `mysql.rds_start_replication` command on the replication RDS instance

Replication as a disaster recover or data migration mechanism

- On-premise to RDS backup (using AWS as a failover)
- RDS MySQL to another region with read replicas
- Multi-AZ failover for synchronous replication
- MySQL replication for importing data to the cloud (also use mysqldump/mysqlimport)



Linux Academy

Amazon Web Services

Self-Healing Techniques And Fault-Tolerant Services

Self-Healing

Many different ways to create self-healing application architectures

- Utilize SQS
- Utilize CloudWatch and assign a “terminate” function to instances that have failed status checks
- Utilize Auto Scaling which will automatically start new instances
- Use cloud-init to boot strap new instances to easily assign “roles” or job “functions” to instances

SQS Self-Healing

When using SQS to decouple your application architecture then each component is operated on its own. This means each component can operate without relying on the previous component or the after component.

Fault tolerant services

Fault tolerant services are those that have built-in tolerance to issues that can occur in your environment. For example the loss of an AZ or an unhealthy instance. If parts of your infrastructure break then it is not completely taken down.

Fault tolerance with EC2:

- Utilize multiple availability zones and the ELB to serve traffic to the instances
- Utilize Auto Scaling and CloudWatch alarms to terminate instances that have failed status checks
- Utilize EBS volumes and snapshots for backups and redundancy

Fault tolerant services

Many AWS services have built-in fault tolerance by working across availability zones

- DynamoDB
- SWF
- SQS
- S3
- Etc..

Note: AWS Releases new services almost quarterly. Please make sure to review the FAQ for each service as part of the study prep. As we go through the required services for AWS PRO we will discuss more about design for fault tolerance. However, this should be already familiar from the CSA associate exam.



Linux Academy

Amazon Web Services Disaster Recovery And Fail-Over Strategies

Disaster Recovery and Failover

RTO (Recovery Time Objective) – The acceptable amount of time it takes to restore applications to the business process service level.

RPO (Recovery Point Objective) – The acceptable amount of data that can be lost due to failure as it is measured in time.

Disaster Recovery and Failover

Services that can be used for backup and disaster recovery

Amazon S3

- Lifecycle policies, versioning, MFA, and eleven nines durability make this a perfect backup solution

Amazon Glacier

- Amazon Glacier provides low cost storage for long term archival

Amazon Elastic Block Store

- Durable point-in-time snapshots stored on the S3 standard storage type
- Ability to copy a snapshot to a new region and launch a new volume (good for preparing your disaster recovery methods)
- EBS volumes are replicated across multiple servers in the same availability zone for durability

AWS Import/Export

- Copy large amount of data to AWS to prepare for a disaster recovery solution
- Send physical storage devices for archival/storage as a backup solution

AWS Storage Gateway

- Gateway-cached volumes
 - Store the data on S3 with versioning enabled as a backup and cache frequently accessed objects on-premise
- Gateway-stored volumes
 - Inexpensive offsite backups by creating point-in-time snapshots of data on the storage gateway to Amazon S3

Disaster Recovery and Failover

Services that can be used for backup and disaster recovery

Amazon EC2

- Copy AMIs to different AWS regions which can be spun up in the event of a disaster
- Use multiple availability zones to design for fault-tolerance and failure
- Route53 to failover to backup environments in another region

EC2 VM Import Connector

- **Important** tool used to import virtual machine images from an existing on-premise environment to Amazon EC2 instances
- Also used to export EC2 instances to go back to on-premise
- Allows for the duplicating environments as a disaster recovery solution either on AWS or on-premise
- Also used for migrating existing applications to AWS
- Supported hypervisors
 - VMware (ESX/Workstation)
 - Microsoft Hyper-V
 - Citrix Xen virtualization (this is actually what AWS is based off of)
- Can also copy an entire VM Image Catalogs to EC2 as AMIs

Disaster Recovery and Failover

Services that can be used for backup and disaster recovery

Route 53

- Can be used to failover from on-premise to AWS or from one AWS region to another
- Failover routing and weighted routing (for migrating applications)
- Allocate DNS ahead of time to prepare for potential failover

Elastic Load Balancer

- Pre-allocate the load balancer for the backup environment to receive the cname (DNS name) this allows for setting up Route 53 record sets in anticipation of a failover/disaster recovery situation

VPC

- Configure VPN or direct connect to extend your on-premise network to the cloud to allow for seamless and secure failover of applications including internal applications that might be available to intranet only

Direct Connect

- Consider using for extremely large workloads that rely on reduced latency and increase bandwidth throughput
- VPN to secure direct connect data

RDS

- AWS to AWS failover by creating snapshots and copying them to another region
- Create a read replica in another region that can be promoted in the event of a disaster (when promoting you need to enable multi-AZ as well as backups and a read replica cannot be promoted unless auto backup is enabled)

Disaster Recovery and Failover

Services that can be used for backup and disaster recovery

DynamoDB

- AWS to AWS disaster recovery with DynamoDB
- Ability to copy data to S3 or replicate it to another region (replication is now built into the DynamoDB service). However, using Data Pipeline which starts an EMR cluster to copy data is still a widely used method.
- Easily scale up your backup region within minutes by an API call to increase throughput (Developer course focuses on throughput).

Amazon Redshift

- Redshift snapshots can be copied to other regions.

CloudFormation

- Build a template of your environment that can be used in multiple regions and only requires inputs such as region specific AMI IDs, IP addresses, or Hostnames. Allows for quick deployment, version control of your backup infrastructure, and a backup of your backup!

OpsWorks

- When combined with CloudFormation the ability to easily deploy new stacks in additional regions is available.



Backup and Disaster Recovery Methods

Pilot Light - A minimal version of an environment that is always running in the AWS cloud. In the event of a failover it takes only a few minutes for a scripted solution to “turn on the furnace” and deploy the disaster recovery solution. Examples include small RDS instances for replication, data being replicated to EBS volumes with smaller size instances. In the event of failover the application will launch larger instances and/or increase the number of instances in the auto scaling group to meet demand. Have pre-configured AMIs with business roles (bootstrap scripts) to easily deploy in the event of a failover. Only “minimal” components used for replication of data are running in the AWS Cloud environment.

Basic Principles:

1. Replicate data from on-premise to EC2
2. Update packages on AWS to ensure all software configurations are in place
3. Maintain proper AMIs with updated configurations
4. Test on a regular basis
5. Use CloudFormation or scripts to automate the recovery process

Backup and Disaster Recovery Methods

Warm Standby - A scaled down version of a fully functional duplicate application in the cloud. In the event of failover auto scale to handle full production load.

In the event of a failover:

1. Scale up with auto scaling to handle the load and use the ELB
2. Increase the size of the EC2 instances if required (have an additional auto scaling policy)
3. Change DNS to cutover from primary to backup solution
4. Ensure the database has multi-AZ enabled and has enough capacity to handle increase in load. Read replicas or changing instance size will assist in the scaling process.



Backup and Disaster Recovery Methods

Multi-Site Solution – An exact operating duplicate of your primary application. In fact, DNS should be able to “load balance” or use weighted based routing to serve traffic from both application environments. One on premise and one in the AWS cloud. In the event of a failover it auto switches to the backup solution. In this situation you generally will only have one “writing” database. In the event of failover ensure that the backup application is writing to the correct database or in this case it would be an RDS instance.

AWS To AWS multi-region failover and disaster recovery – Copy snapshots of EBS/RDS/Redshift to another region, utilize read replicas, and Route 53 for easy failover to design cross region disaster recovery.

Reminder: Replication lesson about replication options



Linux Academy

Amazon Web Services

Application Migration Plans To AWS



Application Migration Plans

- Determine if all software licenses from on-premise are eligible for use on the cloud
- Determine the assets that need to be moved
- Configure replication, instances, and proof of concept on the cloud
- Determine required resources
- Use AWS Import/Export to assist in large data migrations
- To integrate with legacy on-premise applications create a hybrid cloud by configuring a VPN tunnel to the on-premise location
- Write a web wrapper around the legacy application which exposes a developer API to the new application. Utilize SQS queues to glue the application together (this is a hybrid environment). Consider this only if the application is running externally and not internally only. Use VPN tunnel if internal only.



Linux Academy

Amazon Web Services

Network Connectivity Options

Network Connectivity Options (on-premise to Amazon VPC)

Hardware VPN – Cisco Router, Palo Alto, Sonicwall, Watch Guard, and other hardware VPNs are used to connect directly to AWS. Limitations are the internet connection from the data center to the VPC. Requires the customer hardware device to support single-hop BGP and failover on the customer end if it is leveraging dynamic routing. ISP has to support BGP connections.

- BGP connections with Hardware VPN (dynamic routing vs static)
- IPsec connection from on-premise to AWS cloud.
- AWS managed VPN endpoints provide automated multi-data center redundancy and failover on the AWS cloud side of the configuration.
- VPG supports multiple user gateway connections to enable redundancy on the on-premise (client side) of the VPN connection.
- BGP peering is used to exchange routing information between AWS and the on-premise.
- When BGP is configured the hardware device must be capable of terminating both IPsec and BGP connections.

Network Connectivity Options (on-premise to Amazon VPC)

AWS Direct Connect

- 1Gigabit or 10 Gigabit direct connections to AWS region
- Industry standard VLANs
- Reduced latency and more reliable internet connections

Software VPN

- Useful if you have to manage both ends of the VPN software for compliance reasons
- Useful if you have a gateway device that is not currently supported by AWS VPN/VPG
- Single point of failure (OpenVPN)

Note: With VPNs you need to be sure to use different CIDR blocks on-premise/aws as they cannot overlap



Linux Academy

Amazon Web Services

Deployment And Management On AWS

Deployment And Management On AWS

AWS Elastic Beanstalk

- Quickly deploy out web based environments using EC2/RDS/Auto Scaling/CloudWatch/ELB and containers

AWS OpsWorks

- Write custom chef recipes, utilizes self-healing, and works with layers

AWS CloudFormation

- Version control the infrastructure and make deploying out environments easily and repeatable



Linux Academy

Amazon Web Services

Enterprise Account Management With IAM



Enterprise Account Management

Big Cloud Jumbo Corp is a large scale company that needs to provide access to developers, third party auditors, accounting staff, and system administrators. Big Cloud Jump Corp also has several of their own AWS accounts that are used for dev, staging, and prod in order to easily manage budgets.

In this section we are going to describe strategies for managing this account information, budget information, and managing application level budgeting.



Enterprise Account Management

Part of Big Cloud Jumbo Corp services is to provide management of customers AWS accounts and implementation of cloud services and products based around AWS. Managing hundreds of customers AWS accounts so a customer could take their account and leave if required.

Solution:

- Add each customer account as a “consolidated billing” account
- Big Cloud Jumbo Corp will be responsible for all billing
- *Bulk/Volume discounts will span across all accounts*
- AWS combines the usage from ALL accounts to determine which volume pricing tiers to apply, giving a lower over all price whenever possible. This will either provide a source of profit margin or a cost savings for Big Cloud Jumbo Corp customers



Dashboard
Bills
Cost Explorer
Budgets
Payment Methods
Payment History
Consolidated Billing
Reports
Preferences
Credits
Tax Settings
DevPay

Manage Requests and Accounts



Consolidated billing combines multiple AWS accounts under a single payer account. Use this page to invite other accounts to join a consolidated billing account family. Accounts you invite are listed here, with the status of your request. You can also cancel or resend requests, or remove accounts from your account family.

Account Number	Name	Status	Updated	Actions
[REDACTED]	[REDACTED]	Linked	2012-12-14	Remove
[REDACTED]	[REDACTED]	Linked	2014-01-18	Remove
[REDACTED]	[REDACTED]	Linked	2014-01-25	Remove
[REDACTED]	[REDACTED]	Linked	2014-03-24	Remove
[REDACTED]	[REDACTED]	Linked	2014-04-03	Remove
[REDACTED]	[REDACTED]	Linked	2014-05-18	Remove
[REDACTED]	[REDACTED]	Linked	2014-07-20	Remove
[REDACTED]	[REDACTED]	Linked	2014-07-24	Remove
[REDACTED]	[REDACTED]	Linked	2014-07-24	Remove
[REDACTED]	[REDACTED]	Linked	2014-07-24	Remove
[REDACTED]	[REDACTED]	Linked	2014-07-26	Remove
[REDACTED]	[REDACTED]	Linked	2014-11-17	Remove
[REDACTED]	[REDACTED]	Linked	2015-02-06	Remove

[Send a Request](#)



Enterprise Account Management

Consolidated billing best practice is to never have resources fired up in the “payee account” only in the consolidated billing accounts. The payee account should only be used for accounting and consolidated billing purposes.

Important note: AWS Limits work on the account level only and aws support is a per account only



Enterprise Account Management

Consolidated Billing Method: Using one master account and many sub accounts

Pros:

- Volume benefits on services that allow
- Ability to view costs based off of tagged resources as well as accounts
- Easier architecture visibility and configuration
- Use roles for IAM account simplicity across multiple AWS linked accounts

Cons:

- Requires strict and sometimes complex tagging across accounts



Enterprise Account Management

Consolidated Billing Method: Use one account but multiple VPCs to break out environments

Pros:

- Simple billing and insights into the environment
- Easy governance

Cons:

- Requires more complex setups for resource level permissions
- Requires more complex setups with multiple VPCs



Linux Academy

Amazon Web Services

Consolidated Billing and EC2 Reserved Instances



Enterprise Account Management

BCJC (Big Cloud Jumbo Corp) has 3 consolidated billing accounts; dev, staging, and production. The dev account has purchased 2 reserved instances with instance type of m4.large in Availability Zone 1a. However, no instances are running in the dev account but a m4.large is running in the staging account inside of availability zone 1a. Who can receive the pricing?

- Like “volume discounts” reserved instances will work across all accounts that are connected to consolidated billing
- Since billing is at the payee level, consolidated billing does not care which account purchases or uses a reserved instance.
 - This is a consideration if BCJC wants to host customer accounts as part of their consolidated billing



Linux Academy

Amazon Web Services

Budgets And CloudWatch Alarms



Budgets

- Budgets are used to track how close your current costs are to exceeding the set “budget” for a given billing period.
- Budgets are updated every 24 hours
- Budgets do not show refunds
- Budgets are not automatically created by AWS
- Budgets can be compared against AWS “estimated” costs to see how much budget is left over
- Budgets can work with SNS/CloudWatch for billing alerts to receive notifications if you have gone over your designated budget or even if you are “close” to going over
- AWS Credits currently “skew” Forecasts provided by AWS



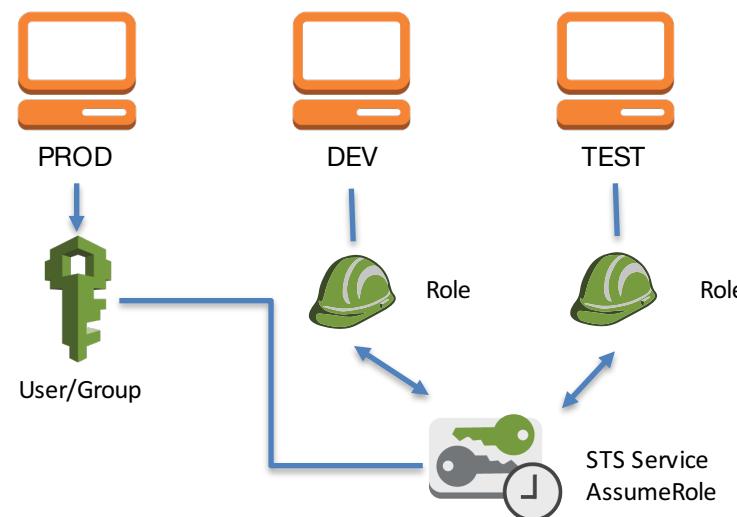
Linux Academy

Amazon Web Services

IAM Cross Account Users

Enterprise Account Management

Scenario: BCJC managers need administrative access to the test, dev, and production accounts





Enterprise Account Management

Scenario: BCJC managers need administrative access to the test, dev, and production accounts to “start/stop” instances, exercise “least privilege” security design method.

1. Create the role with “stop/start” only permissions in both the dev/test account named “manager”

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "Stmt1441397689000",  
      "Effect": "Allow",  
      "Action": [  
        "ec2:StartInstances",  
        "ec2:StopInstances"  
      ],  
      "Resource": [  
        "*"  
      ]  
    }  
  ]  
}
```



Enterprise Account Management

2. Create an IAM user account in the “master/production” environment
3. Add permissions for the IAM user to “sts:AssumeRole” on the role ARNs for the Dev/Test accounts

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "Stmt1441393420000",  
      "Effect": "Allow",  
      "Action": [  
        "sts:AssumeRole"  
      ],  
      "Resource": [  
        "arn:aws:iam::account-id-for-dev:role/manager"  
      ]  
    }  
  ]  
}
```

*Note: Duplicate the policy statement to add another ARN one ARN per policy statement or * for “all” arns but this opens security issues and does not use the least privilege security strategy*



Enterprise Account Management

Scenario: Developers need access to “view” EC2 resources within the production account and developers IAM accounts are built on the developer AWS account.

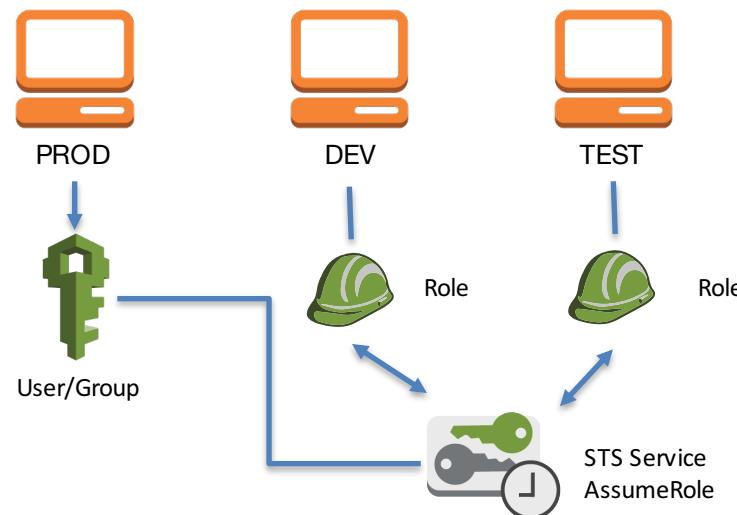
Notes:

- The IAM accounts for the developer on the dev account only need a permissions policy to “AssumeRole” on the production role ARN
- Once they assume the role in the production account the permission policies associated with the role inside of production is what determines the permissions for the developers who assume the role
- This is a great method for granting read only access to an auditor when you have multiple AWS accounts
- Third party cross-account role option will required the “external” account id
- Roles use STS permissions to assume a role and each role as a temporary unique security access key and secret access key associated with it when assumed



Enterprise Account Management

Scenario: Auditor needs read only access to all AWS accounts owned by your company





Linux Academy

Amazon Web Services

Temporary Access Using Roles and STS

Temporary Access Using Roles and STS (Security Token Service)

- The endpoint is <https://sts.amazonaws.com>.
- STS is enabled by default in all regions (this was changed on 11-11-15)
- Roles is of the configurations utilized with STS to gain temporary security credentials
- Temporary credentials require the “token” as well as the access key and secret access key in order to make API calls

Temporary Access Using Roles and STS (Security Token Service)

- EC2 Instance Roles: Any role used, such as an EC2 instance role, will receive “temporary” security access key and secret access key
- SDKs automatically use the temporary credentials with the role, however, you can view and access the temporary credentials using the following command:
- curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role-name

```
[root@ip-10-0-0-56 ~]# curl http://169.254.169.254/latest/meta-data/iam/security-credentials/awscli
{
  "Code" : "Success",
  "LastUpdated" : "2015-09-07T17:04:56Z",
  "Type" : "AWS-HMAC",
  "AccessKeyId" : "ASIAI4HGA64SE2KAIC5A",
  "SecretAccessKey" : "xe1m1sN3PQ2rljYdv0+wqvR0IwRh+AtWBfhce6",
  "Token" : "AQoDYxdzG1a4APL8s+29K4rGrjh6IZld1RpPWYYB76Z/zkC2l1+uI+Zuz65SVa0U6snyXshx3jSXiwkBJS52xAzWWsm0zqUwnochNNiKSTLiKyFqEaWgApZUPvmoMSa/q+8R/xW3zDPkBcb18XnlQJ7pVW5+8EdLKxI65INdpTxULTDbJ+qe2Jy+hzeroef1fsFegW1APyvGK1c5VYY7NafdlmVCQ412kkeglnsQ+yI6Qos39m06b4M4EaGHN6LMmdbNU14ewa0jHW1dxoP/fvmFYwySjJa9167TuJ7RgL6/FBqqBILklg2h2/SaK8Sn6aEVrRyQs++088d7bxn6jUW1ezJuMqbhXxNA1Gn4454pJKQgFMH81JH3aKgCiqGxMwzyx/1oQfxbiLcznLxnJD0/wn2I8U4cTm63iFT/zQwfB0D6JzGhVLtfTfbepL3ehWTFWgBWY01ZcpbmBvxpq7BNoyHPPS9cx6+LeE10v4NXTp2Ra9FL5FyFrpk1K/9lKtLHAzFD00QY3jk06B0d6Uh80oPZsqgY+e6190Ka0/aJUTRKJLo2FAV7f4al97Es04FqBFxtGmSwrnOyGoGKRk3wcYwk7gJeGxQctQ8m5dSa1y7Ttv/tqHrzaZTKQx6wg+V4flzqYTcg44e3rwU",
  "Expiration" : "2015-09-07T23:19:59Z"
}[root@ip-10-0-0-56 ~]#
```



Temporary Access Using Roles and STS (Security Token Service)

- An AWS service can “assume” a role by requesting temporary security credentials for a role (we have an example in the EC2 section)
- IAM users can temporarily switch to a role to use the permissions of the role
- Mobile applications: You have a mobile application which needs access to DynamoDB tables, do not embed IAM credentials in the application, instead use roles that allow a web identity federated user to assume a role that allows access to the DynamoDB table by providing temporary credentials. Again, the SDK will use those credentials automatically.
- Single Sign-on (SSO): Identity Federation (next lesson)



Linux Academy

Amazon Web Services

Federated Access Using SAML



Federated Access Using SAML

- AWS IAM Supports SAML 2.0-based Federation
- Pre built services such as Active Directory need to work with SAML 2.0 or a custom Identity broker will need to be created
- Federation allows an identity provider to enable single sign-on so users can login to an AWS Management Console or use the AWS APIs
- When working with SAML to assume a role, the AssumeRoleWithSAML call is used



Steps to configure SAML

1. On the identity provider, register AWS as a service provider using the SAML metadata located at <https://signin.aws.amazon.com/static/saml-metadata.xml>
2. With the identity provider, generate the proper metadata XML file which describes the identify provider to AWS
3. Upload the XML document from step 2 into IAM when “creating a SAML Identity” provider
4. Create one or more roles and, as part of the roles trust policy, set the SAML provider as the principle the permissions policy establishes which users from your identity provider are able to perform what tasks
5. Use “assertions” to map what users/groups will map to which AWS roles
6. Call AssumeRoleWithSAML API call and pass the roles ARN to be assumed and the SAML assertion about the current authenticated user from the identity provider
7. If successful, the API will return a set of API access keys and a session token



Federated Access Using SAML

Web-based single sign-on (WebSSO) to the AWS Console from an organization with Active Directory Federation Services using SAML 2.0:

1. The “web-based” login portal is not part of AWS but rather provided by the Identity provider (ADFS)
 2. The portal verifies credentials on your organization’s AD
 3. Once verified, the portal generates a SAML authentication response that includes assertions which identify the user and includes information about the user and sends response to browser
 4. Once response is received, the client’s browser is redirected to the AWS single sign-on endpoint, the browser is redirected to the single sign-on endpoint and posts SAML assertion (<https://signin.aws.amazon.com/saml>) but is a URL generated in IAM. The endpoint calls AssumeRoleWithSAML API to request temporary credentials from STS and creates a sign-in URL that uses those temporarily credentials
 5. AWS sends the role sign-on URL back to the client browser with a “redirect”
- Roles which are configured to work with SAML will have a “saml:group”: “groupname”



Federated Access Using SAML

Scenario: Providing S3 home directories to users in your organization which access the directories by using a SAML based SSO configuration and not allow each user to see each others folders.

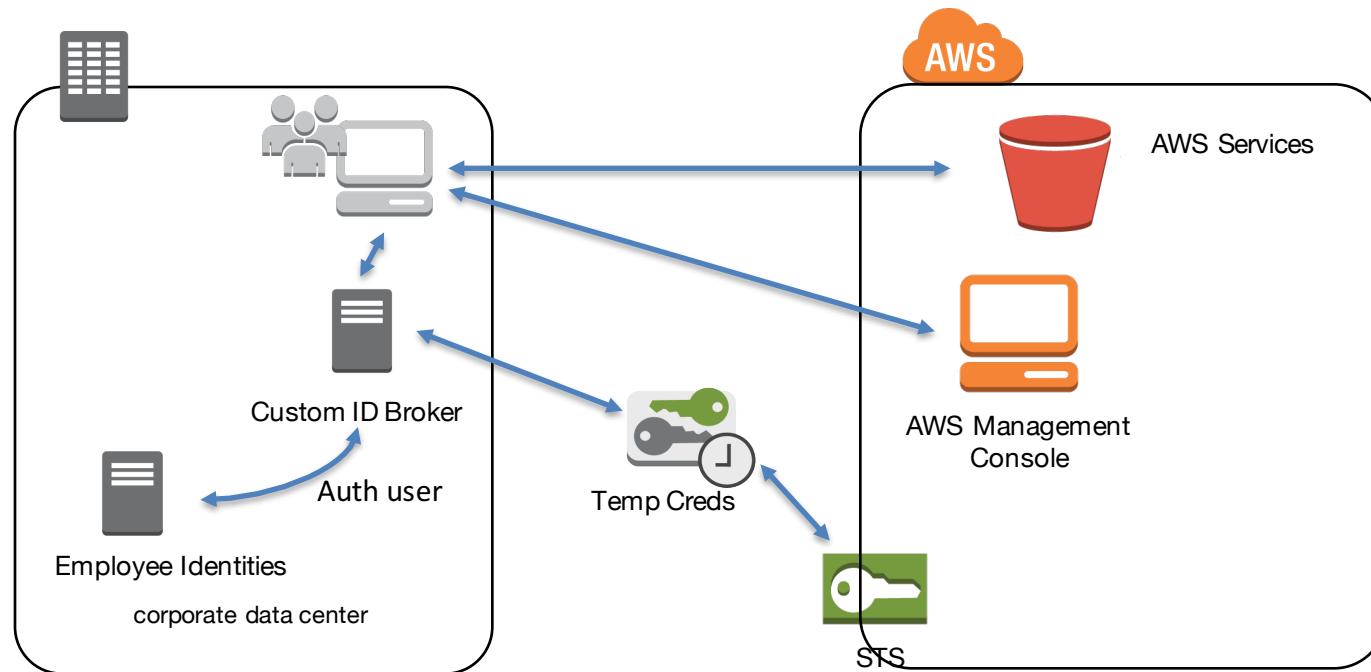
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::bcjc/employee/${saml:namequalifier}/${saml:sub}",
                "arn:aws:s3:::bcjc/employee/${saml:namequalifier}/${saml:sub}/*"
            ],
            "Condition": {"StringEquals": {"saml:sub_type": "persistent"}}
        }
    ]
}
```



Problem: What if your Identity Provider does not support SAML 2.0?

Solution: Write a custom identity broker application

- Uses the AssumeRole or GetFederationtoken API calls to gain temporary access credentials
- The Identity Broker Application has permission to access STS to create temporary credentials
- Identity broker application verifies the on-premise employees within the existing auth system
- Users are able to get a temporary URL or API keys to access AWS





Linux Academy

Amazon Web Services

Web Identity Federation



Web Identity Federation

- Let users sign in to an app using a third party identity provider like Amazon, Facebook, Google or any OpenID 2.0 compatible provider.
- Once a user authenticates, you can allow the authenticated user access to STS to gain temporary role based access to an AWS service such as DynamoDB
- The app should cache the STS credentials until they are expired so only one call is made each time the user logs in and by default the credentials are good for one hour but can be changed in the request
- `AssumeRoleWithWebIdentity` is the API call used when using Web Identity federation



Linux Academy

Amazon Web Services

Monitoring And Security With CloudTrail

What is CloudTrail?

Every action that occurs on AWS is the result of a single API call

Working in an agile environment means many people could have access to your environment each making requests using CLI, SDK's or, the console

It audits and certain compliance certifications require that you log and report every event that occurs in your environment

- Time of the event
- Who made the event call
- The source of the call
- Etc.

CloudTrail Use Cases

Security Analysis

- Activity pattern matching (similar to monitoring network traffic)

Track and Monitor Changes to AWS Resources

- Know who, what, how, and where from changes were made to AWS resources

Compliance Aid

- Compliance requirements for source and logs of changes to environments
- PCI/HIPAA Compliance etc.

Troubleshoot Operational Issues

- Identify recently changed resources to time an issue occurs

CloudTrail Concepts

Once configured CloudTrail logs all API events and delivers the log to an S3 bucket

CloudTrails are configured on a per region basis and a region can include global services

CloudTrails log files from different regions can be sent to the same S3 buckets

CloudTrail can integrate into SNS, CloudWatch, and CloudWatch logs to send notifications when specific API events occur



Logging Best Practices

Limit and control access to CloudTrail and CloudTrail logs

Configure logs to notify in the event of misconfiguration

Integrate with lifecycle policies to store for industry standard time frames

- HIPAA and PCI compliance are examples of requiring 6 years of log storage



Linux Academy

Amazon Web Services

AWS KMS (Key Management Service)



AWS KMS

Key Management Service is a region specific hosted service that makes it easy to create and control encryption keys on AWS which are used to encrypt data.

KMS uses Hardware Security Modules (HSMs) to protect the security and integrity of keys.

KMS not only integrates with other AWS services to automatically manage keys for protection but also allows you to generate and store your own keys within the KMS.

To ensure compliance and security monitoring requirements KMS integrates with key based permission policies, IAM policies and CloudTrail.



AWS KMS – Encryption Concepts

Key encryption is a class of encryption based on specific algorithms that create two different keys which are needed for a single encryption/decryption process also known as Asymmetric encryption which is what is supported by KMS.

Plaintext - Refers to the data that is not encrypted, for example a password.

Ciphertext – Refers to the encrypted plaintext



AWS KMS – KMS Concepts

KMS is used on a per region basis and is managed out of the IAM console

The screenshot shows the AWS KMS 'Create Key' interface. At the top, there are two buttons: 'Create Key' (blue) and 'Key Actions' (grey). Below them is a 'Filter' dropdown set to 'US East (N. Virginia)' and a search bar. A modal window is open over the main table, listing various AWS regions:

Region	Key ID	Status
US East (N. Virginia)	a38009bd-4e46-40cf-9585-9a63c6f49b2c	Enabled
US West (Oregon)		
US West (N. California)		
EU (Ireland)		
EU (Frankfurt)		
Asia Pacific (Singapore)		
Asia Pacific (Tokyo)		
Asia Pacific (Sydney)		
South America (Sao Paulo)		

The 'US East (N. Virginia)' row is highlighted with a blue background. The modal window lists the following regions:

- US East (N. Virginia)
- US West (Oregon)
- US West (N. California)
- EU (Ireland)
- EU (Frankfurt)
- Asia Pacific (Singapore)
- Asia Pacific (Tokyo)
- Asia Pacific (Sydney)
- South America (Sao Paulo)



AWS KMS

Customer Master Key (CMK) – A logical key that represents the top of a customer’s key hierarchy and is also assigned an alias (which can be used in place of the key ID) and an ARN (which includes the unique key ID)

- If another key is not specified then by default the CMK is used to encrypt the resources.
- CMK settings cannot be modified
- IAM permission can be granted to IAM users to “administrate” a key.
- Key policies can be created which state the users that can use the key
- The ciphertext includes information about what key used to encrypt the data
- Additional AWS accounts can be granted access to use a key

External Accounts

The following external accounts can use this key to encrypt and decrypt data. Administrators of accounts defined below can further restrict usage permissions on this key by creating IAM policies for their users and roles that reference the ARN of this key.

[Add External Account](#)

[Save Changes](#)

Key Rotation



AWS KMS – Use your own keys but store them on KMS

You can use a CMK to encrypt a key of your own creation. That key can be stored on AWS and doesn't have to be stored on a local environment. The key will be secure and can be accessed programmatically using the API. Decrypt the additional key for usage.

Benefits:

- Secure storage
- Central location and easy audit trail
- Easy key rotation



AWS KMS – Key rotation

If key rotation is enabled for a specific CMK

- KMS will create a new version of the backing key for each rotation
- The backing key is used to perform cryptographic operations.
- KMS will automatically use the latest version of the backing key to perform data encryption.
- To decrypt data KMS will determine which key (the old or new) that the data was encrypted with and it will automatically decrypt it with that correct CMK.
- To start fresh then change the CMK that your data encrypted tool points to.



AWS KMS – Core service design features

Durability - Designed to equal the highest durability services in AWS. Data encrypted under a key becomes irretrievable if the key is lost.

Quorum-based access - No single Amazon employee can gain access to a customer's master keys.

Access control - Access to keys is protected using existing policies in IAM.

Low-latency and high throughput - KMS will provide cryptographic operations at throughput suitable for use by other AWS services.

Regional Independence - AWS provides regional independence for customer data, in other words the key usage is isolated within an AWS region.



Linux Academy

Amazon Web Services Kinesis



AWS Kinesis – What is Kinesis?

Kinesis is a real-time data processing service that continuously captures and stores large amount of data to power real time streaming dash boards of incoming data streams.

Kinesis dashboards can be creating using the AWS provided SDKs and can create real-time dashboards, integrate dynamic pricing strategies, and also allows you to export data from Kinesis to other AWS services for storage. Including EMR, S3, RedShift, and Lambda.

Build dashboards or applications that react to the incoming data.



AWS Kinesis – Benefits

Real-time processing – Continuously collect and build applications that analyze the data as it's generated

Parallel Processing – Multiple Kinesis applications can be processing the same incoming data streaming concurrently

Durable – Kinesis synchronously replicates the streaming data across three data centers within a single AWS region and preserves the data for up to 24 hours

Scales – Can stream from as little as a few megabytes to several terabytes per hour



AWS Kinesis – When would you use Kinesis?

Gaming – Collect gaming data such as player actions and feed the data into the gaming platform, for example a reactive environment based off of real-time actions of the player

Real-time analytics – Collect IOT (sensors) from many sources and high amounts of frequency and process it using Kinesis to gain insights as data arrives in your environment

Application alerts – Build a Kinesis application that monitors incoming application logs in real-time and trigger events based off the data

Log / Event Data collection - Log data from any number of devices and use Kinesis application to continuously process the incoming data, power real-time dashboards and store the data in S3 when completed

Mobile data capture - Mobile applications can push data to Kinesis from countless number of devices which makes the data available as soon as it is produced.



AWS Kinesis – Workflow

Create a stream

Build producers to continuously input data into the stream

- Sensors
- Mobile devices
- Literally thousands of different inputs (more shards is how you scale)

Consumers consume the stream (concurrently)

- Real-time dashboards
- S3
- Any application can consume the incoming data
- Redshift (data warehouse)
- EMR

Kinesis keeps 24 hours of streaming data stored by default, but can be configured to store up to 7 days.



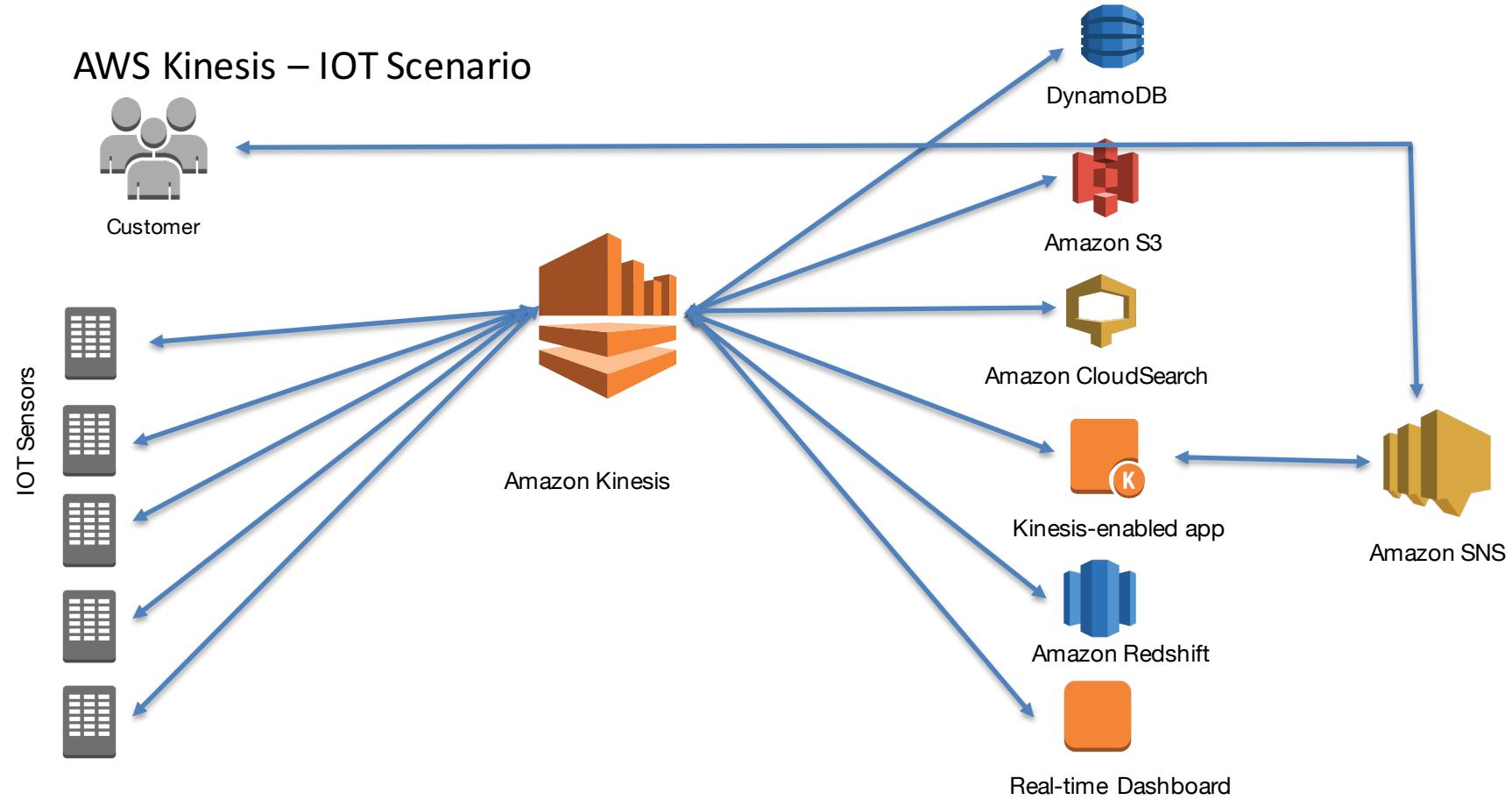
Linux Academy

Amazon Web Services

Kinesis Mobile IOT Scenario Example



AWS Kinesis – IOT Scenario





Linux Academy

Amazon Web Services

EC2 Section



Linux Academy

Amazon Web Services

Protecting Production Resources



Protecting Production Resources In EC2

In an AWS account which has several teams and applications available, there is a need to protect certain resources or allow access to only certain EC2 instances depending on the developer's team or the type of environment such as staging, dev, or production.

How can we best protect those resources?

How can we add additional layers of protection to those resources?



Protecting Production Resources In EC2

Ensure that proper tagging strategies have been implemented that identify production resources or at least the “type” of resource you are wanting to add an extra layer of protection to

Methods:

- Explicit deny on the “action/api” permissions not allowed on the tagged resource
- Grant allow on the “action/api” permissions allowed on the tagged resource

```
"Action": [  
    "ec2:StartInstances",  
    "ec2:StopInstances",  
    "ec2:RebootInstances",  
    "ec2:TerminateInstances"  
,
```



Specify the condition that should occur when this specific policy declaration should be enforced

```
"Condition" :{  
    "StringEquals" :{  
        "ec2:ResourceTag/env" :"production"  
    }  
}
```



Specify the resource type that the policy should apply on

```
"Resource": [  
    "arn:aws:ec2:region:aws-account-number:instance/*"  
,
```



Additional policies for adding extra layer of protection

- Add an IpAddress condition which specifies that the request should come from a specific IP address or CIDR block range
- Require that MFA has occurred recently (number of seconds since)



Scenario: How could we prevent developers who need access to terminate development instances from terminating production instances?

Hands-on example



Study Note: Remember not all “actions” are supported on resource level permissions. Because of this it is easier to use “deny” permissions such as deny starting, stopping, terminating instances that have a production resource tag.



Linux Academy

Amazon Web Services

Migrating Resources To Another Region



Migrating Resources To Another Region

- Part of having multi-region failover is understanding how to copy and replicate data from one region to another
- Part of designing properly is ensuring your resources are closest to their end users

Data that needs to be migrated:

- Databases running on EC2
- EC2 Instances / AMIs
- Auto Scaling Configurations
- EBS Volume Data
- Instance SSH Keys
- VPC and Internal IP address considerations
- Reserved Instances



Migrating Resources To Another Region: EC2 Configurations

Considerations:

- PEM keys are unique per region when you copy an AMI the authorized key will copy with it, to use your existing PEM key ensure you launch the AMI in the secondary region with the same PEM name but use existing PEM key name
- Use the CLI to export current Auto Scaling configurations and create new ones in the new region
- Launch a new ELB in the desired region which will be used during the DNS cutover
- Existing SSL certificates can be used on the new ELB since IAM stores the SSL certificates and is a global service
- Sell existing reserved instances in source region and/or purchase new reserved instances in the destination region
 - Reserved instances cannot be migrated to another region or even AZ



Migrating Resources To Another Region: AMIs and EBS Volumes

Considerations:

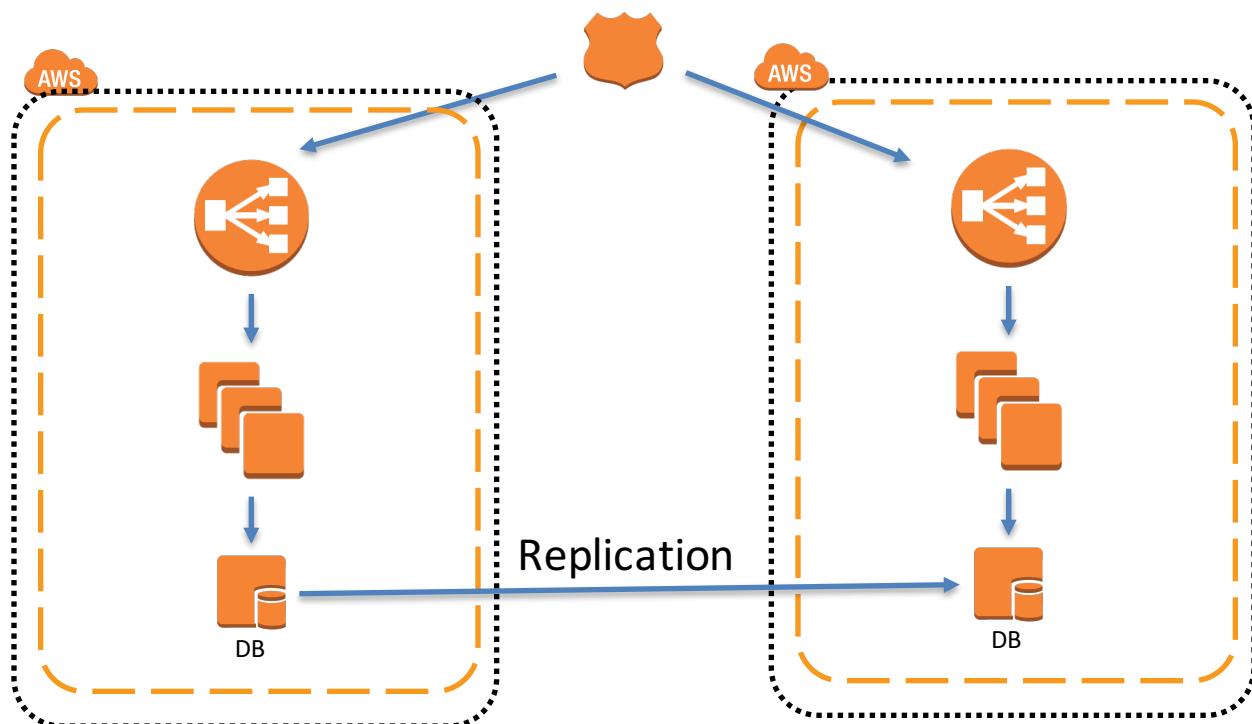
- Some snapshots require the suspension of I/O operation on the volume
- Multiple snapshots and EBS copy events can run in parallel reducing
- Snapshots only work on EBS backed instances/volumes
- Use AMIs to copy the EC2 instance, especially if it is backed by instance-store ephemeral
- Use “AMI copy” to copy the AMI from region1 to region2
- Not all AZ's support EBS optimized instance types ensure which ones do before migrating

Procedure:

- Snapshot volumes and use the snapshot copy feature to copy the snapshot to another region then launch the volume from the snapshot
- Use copy AMI feature to copy the AMI from current region to destination region



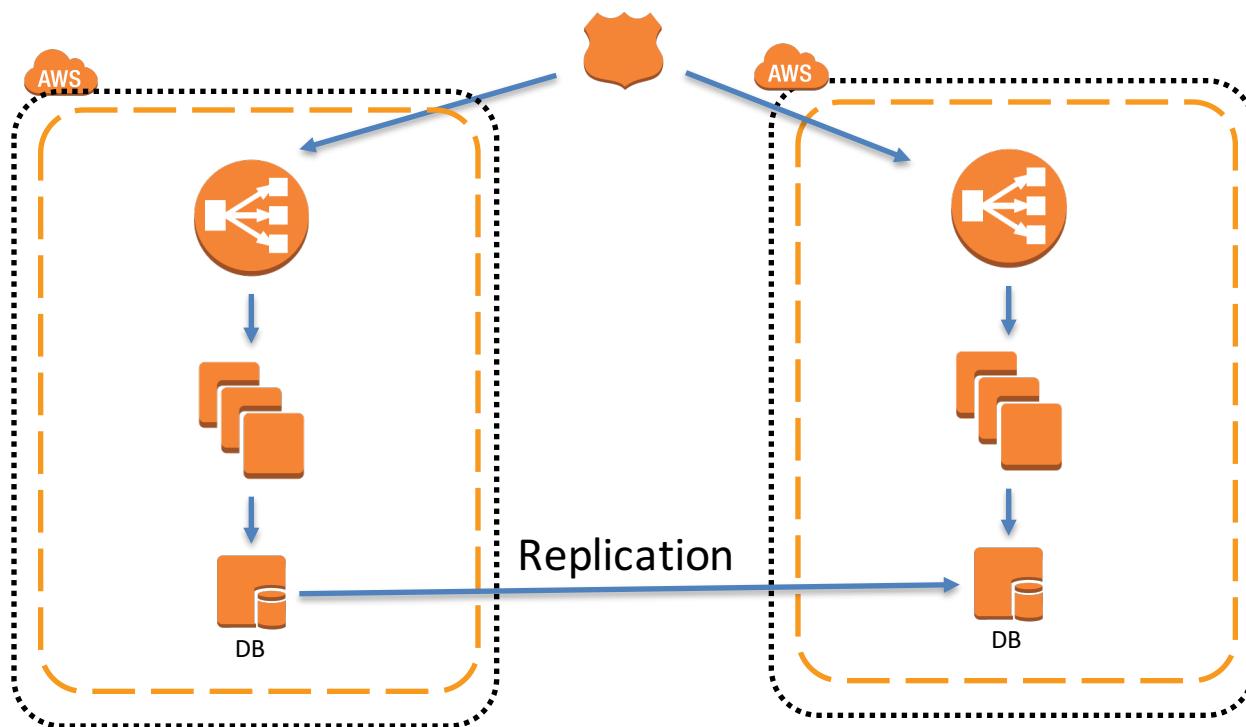
Migrating Resources To Another Region: Databases Running On EC2



- Create an AMI of the instance
- Use AMI Copy to region2
- Enable Replication region1 to 2
- Cutover DNS and application so that writes begin to occur on region2



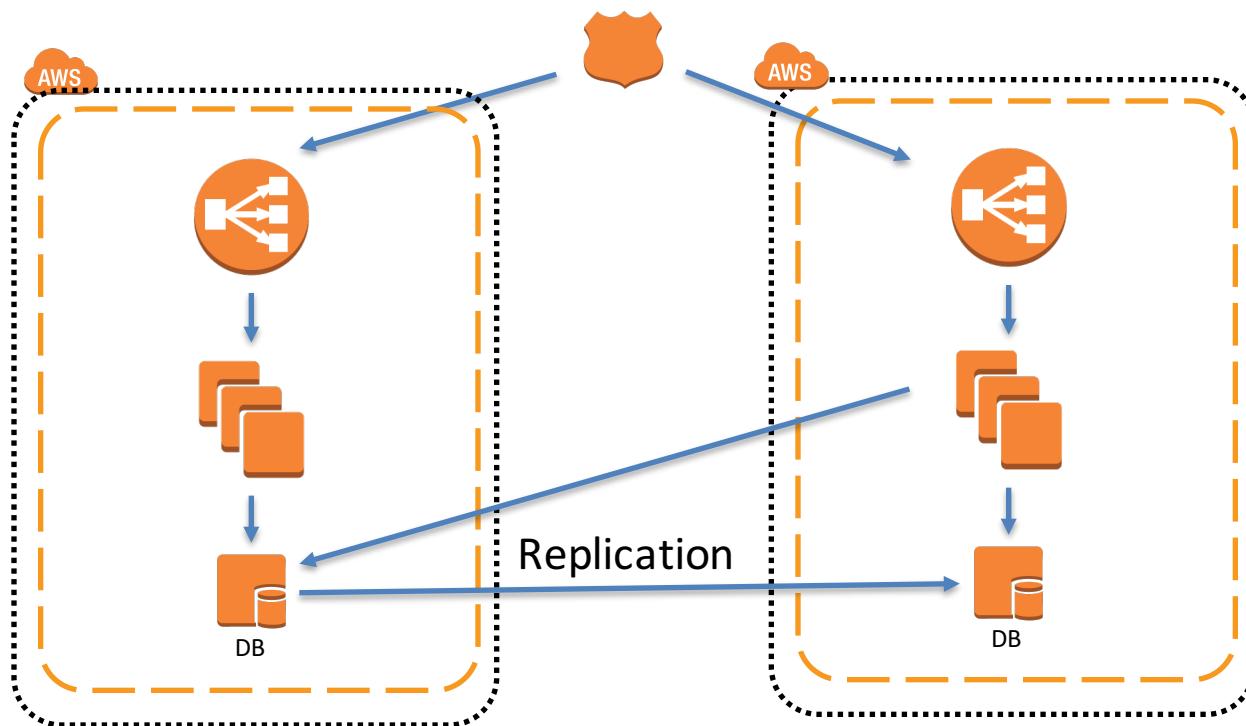
Migrating Resources To Another Region: Route53 Strategies



- If replicating data the Route 53 Weighted DNS record sets would be the preferred method
Start with low weighted sets on the destination region to ensure all configurations are place and gradually change the weighted load
Ensure that DB writes still occur on the master instance
After weights are increased cutover primary DB to destination region and cutoff the source region by removing the extra Route 53 record set



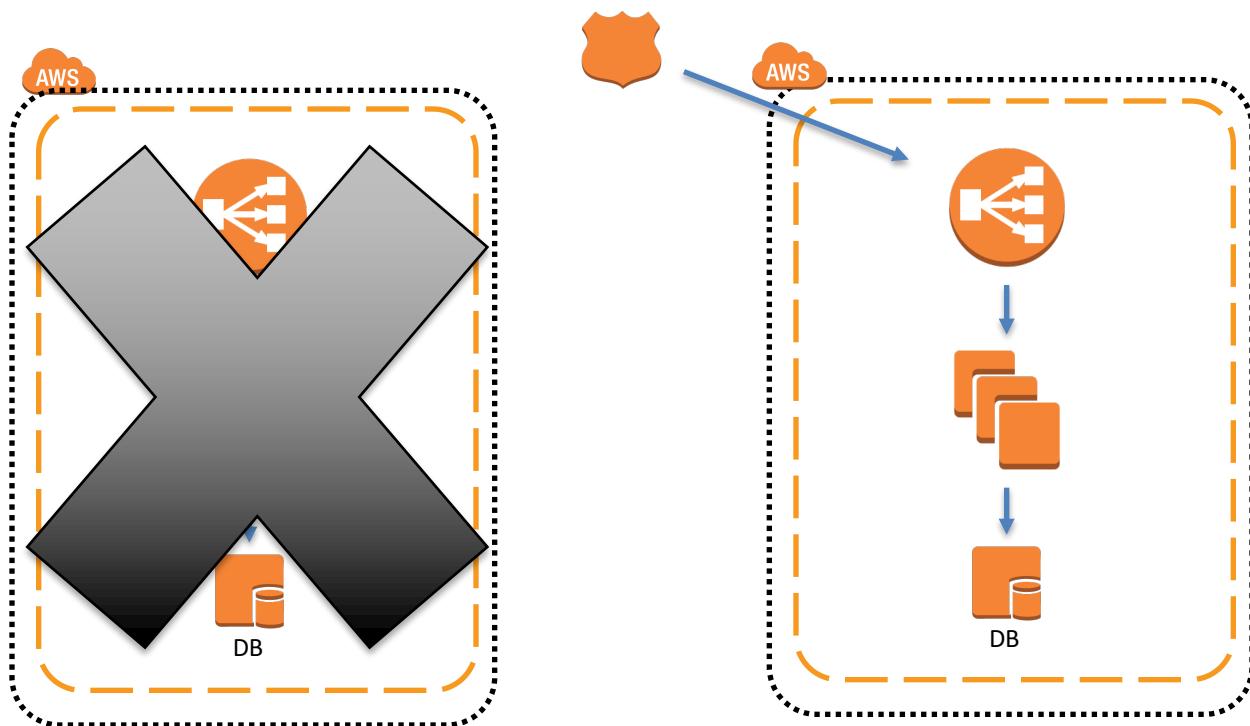
Migrating Resources To Another Region: Route53 Strategies



- If replicating data the Route 53 Weighted DNS record sets would be the preferred method
Start with low weighted sets on the destination region to ensure all configurations are place and gradually change the weighted load
Ensure that DB writes still occur on the master instance
After weights are increased cutover primary DB to destination region and cutoff the source region by removing the extra Route 53 record set



Migrating Resources To Another Region: Databases Running On EC2



- Create an AMI of the instance
- Use AMI Copy to region2
- Enable Replication region1 to 2
- Cutover DNS and application so that writes begin to occur on region2



Linux Academy

Amazon Web Services

EC2 Backup Strategies



EC2 Backup Strategies: AMIs

Instance-Store AMIs: Instance-Store AMIs rely on ephemeral storage; all software configurations and packages installed need to either be bootstrapped or stored on the AMI. To backup these instances it is acceptable to create AMIs frequently. If the data changes and needs to be stored, consider switching to EBS volumes.

Note: In EC2 you are generally backing up configurations with AMIs

EBS Backed AMIs: EBS backed AMIs can be backed up in one of two ways

1. **EBS volume snapshots:** Depending on the workload, suspension of I/O might be required. An AMI can be created from a “root” EBS volume
2. **AMIs:** An AMI will create a snapshot of the attached EBS volumes if configured correctly and the volumes will be restored upon launching the AMI



EC2 Backup Strategies: File Level Restore

1. Take frequent EBS snapshots (incremental)
2. To restore a file create an EBS volume from the desired snapshot
3. Attach the EBS volume to the EC2 instance at a different mount location
4. Browse the file system to the files needing to be restored
5. Copy from the volume to the regular production volume



Linux Academy

Amazon Web Services

Architecting For Performance



Instance Type	Usage
Storage Optimized	Large Data Stores
Memory Optimized	Applications that require more memory – DB – Qlikview (In memory) - Caching
Compute Optimized	Applications that require larger CPU processing such as Video Encoding – Batch Processing
General Purpose	Common applications that need an even mix of resources
GPU	Graphic manipulation – Game Streaming

<https://aws.amazon.com/ec2/instance-types/>



GPU instances are great for high parallel processing capability

Examples:

- Scientific calculations
- Engineering modeling
- Rendering applications
- Graphics applications
- Game streaming
- 3-D application streaming
- Other graphic work loads
- Remember GPU instances do not support SR-IOV

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using_cluster_computing.html#gpu-instance-current-limitations

Architecting For Performance: Burstable CPU Credits

What if you have a legacy application that cannot scale with auto scaling but has peak performance 10% of the time? How can we reduce costs while still being able to handle any increase in temporary load?

What about applications that are dev/test/staging environments that do not frequently run large amounts of data? How can we reduce costs but still have the required performance?

Burstable instances are perfect for workloads that do not use the full CPU often but casually need to burst.



Architecting For Performance: Burstable CPU Credits

- T2 instance types have “burstable” CPU performance
- Each instance has a “base line” performance but can “burst” to greater CPU usage if credits allow
- One CPU “credit” is equal to one vCPU running at 100% utilization for one minute
- One CPU “credit” is equal to one vCPU running at 50% utilization for two minutes, etc
- “Credits” are accrued when the instance uses LESS than it’s base level performance

Instance type	Initial CPU credit*	CPU credits earned per hour	Base performance (CPU utilization)	Maximum earned CPU credit balance***
t2.micro	30	6	10%	144
t2.small	30	12	20%	288
t2.medium	60	24	40%**	576
t2.large	60	36	60%**	864



Architecting For Performance: Storage

Instance store instances have storage that is physically attached to the host machine.

Instance store instances have faster i/o operations because of this. However, instance store data is ephemeral and will be deleted if the instance is stopped or terminated.



Architecting For Performance: Storage

Volume Type	Use Cases
General Purpose	Root/boot volumes, desktops, smaller databases, and non-prod workloads
Provisioned IOPS	Production work loads and databases
Magnetic	Infrequently accessed workloads and low cost requirements



Architecting For Performance: Storage

Volume Type	Limits
General Purpose	1GiB – 16TiB, 160MiB/s, baseline performance of 3 IOPS/GiB with burstable “credits”
Provisioned IOPS	4GiB – 16TiB, 320MiB/s, Up to 20,000 IOPS per volume
Magnetic	1GiB – 1TiB, 40-90MiB/s, 100 IOPS with burstable to hundreds of IOPS

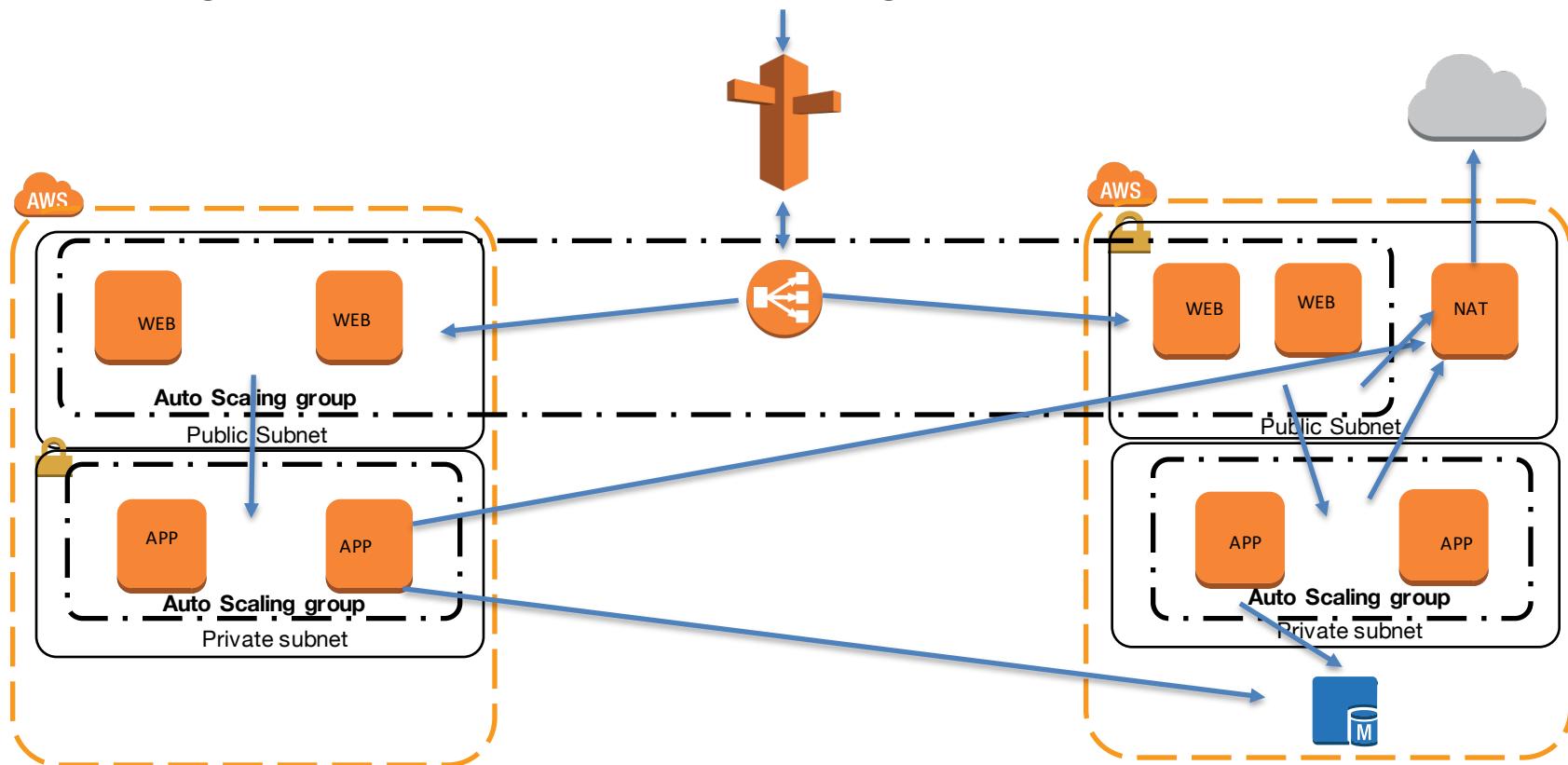
Architecting For Performance: Performance Design Patterns, Understand Bottlenecks

BCJC has a video transcoding application which accepts videos, processes the videos, and applies certain filters to the videos. The process includes uploading the video through an ELB to the web facing servers. When completed the web servers upload the video to the external location (or S3) through the NAT instance.

- Web facing servers do not have a public IP address but receive traffic from a public load balancer
- Use a NAT instance to upload the videos to the external location

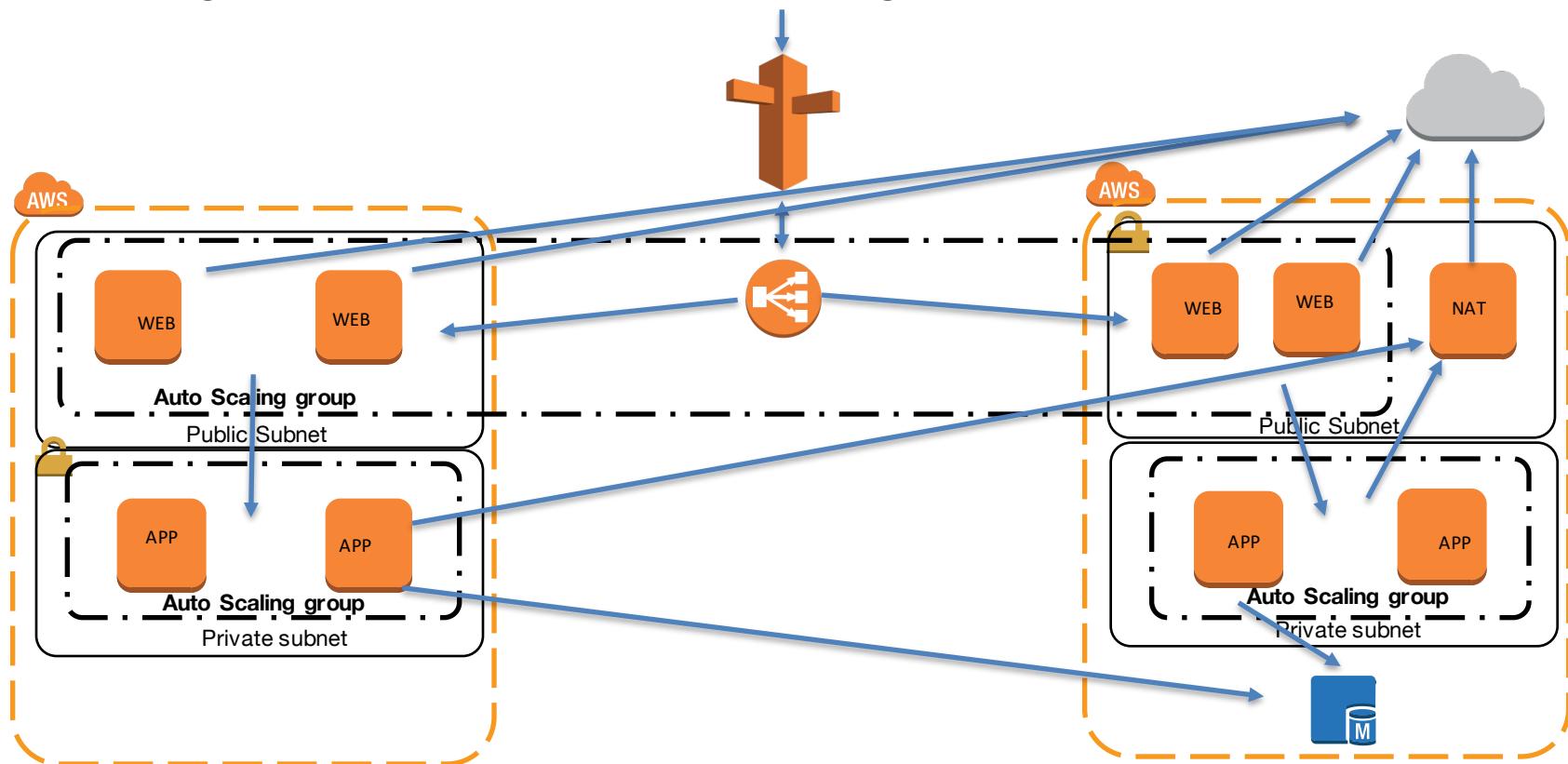


Architecting For Performance: Performance Design Patterns, Understand Bottlenecks





Architecting For Performance: Performance Design Patterns, Understand Bottlenecks





Architecting For Performance: Performance Design Patterns, Understand Bottlenecks

Solution:

- Add a public subnet layer to the ELB and enable auto scaling to assign public IP addresses so each instance can send the traffic rather than going through the NAT instance
- You can also create multiple NAT instances and assign one to each subnet but this also begins creating HA issues
- Increasing the instance size increases the bandwidth throughput



Linux Academy

Amazon Web Services

Increasing Performance With RAID Configurations



RAID: RAID Configurations

Configuration	Issues / Benefits
RAID 0	Need more I/O performance – Performance of the stripe is limited to the worst performing volume – Does not provide redundancy
RAID 1	Provides volume fault tolerance but no additional I/O

With RAID 0 you will get whatever additional throughput you provision on attached EBS volumes. Striping together two 20,000 volumes in RAID 0 will result in 40,000 IOPS I/O



RAID: Scenario

Problem: BCJC has an application with a need for 120,000 IOPS of write performance. However, EBS volumes can only provision a maximum of up to 20,000 IOPS each. How would you solve this situation?



RAID: Scenario

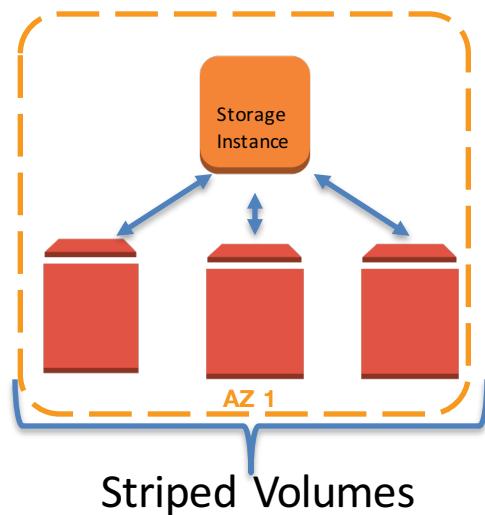
Problem: BCJC has an application with a need for 120,000 IOPS of write performance. However, EBS volumes can only provision a maximum of up to 20,000 IOPS. How would you solve this situation?

Solution: Stripe multiple EBS volumes together with RAID! For example, you can create a RAID 0 configuration for 6 20,000 IOPS volumes for 120,000 IOPS. Keep in mind your limitation will be bandwidth so EBS optimized and/or network optimized instances might be required.

Consider using RAID for storage services (NFS/CIFS) on AWS if S3 is not an acceptable solution



RAID: Needing higher I/O throughput

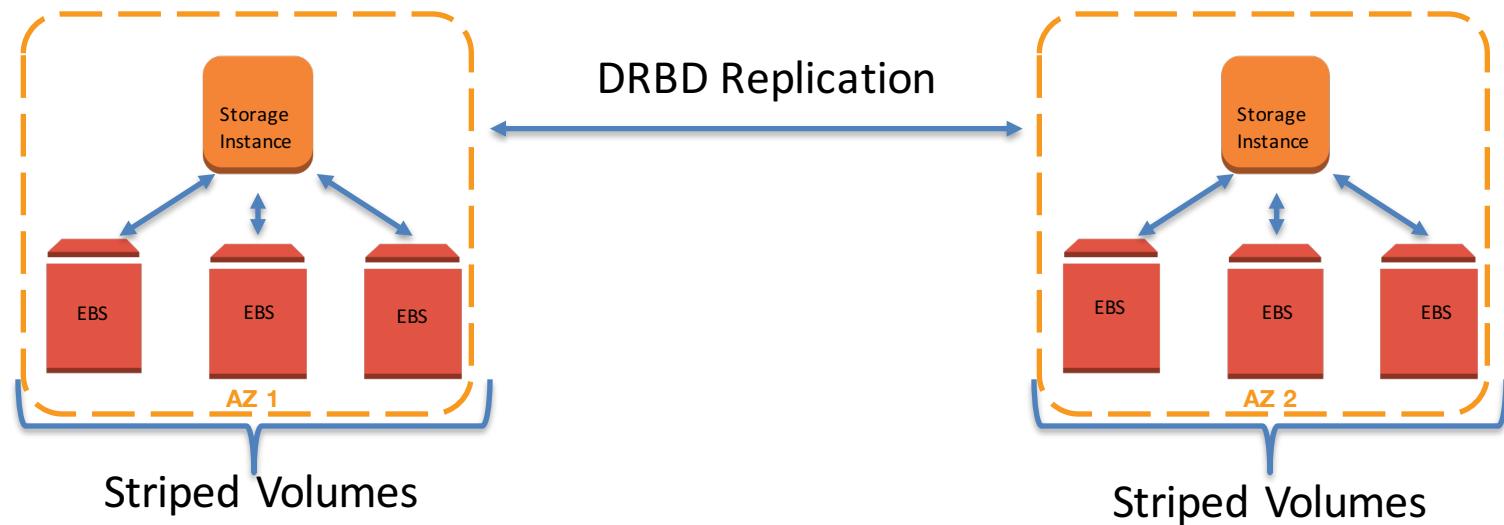


- Create durability by applying EBS snapshots
- What about high availability?



RAID: Needing higher I/O throughput

https://en.wikipedia.org/wiki/Distributed_Replicated_Block_Device





RAID: Problem

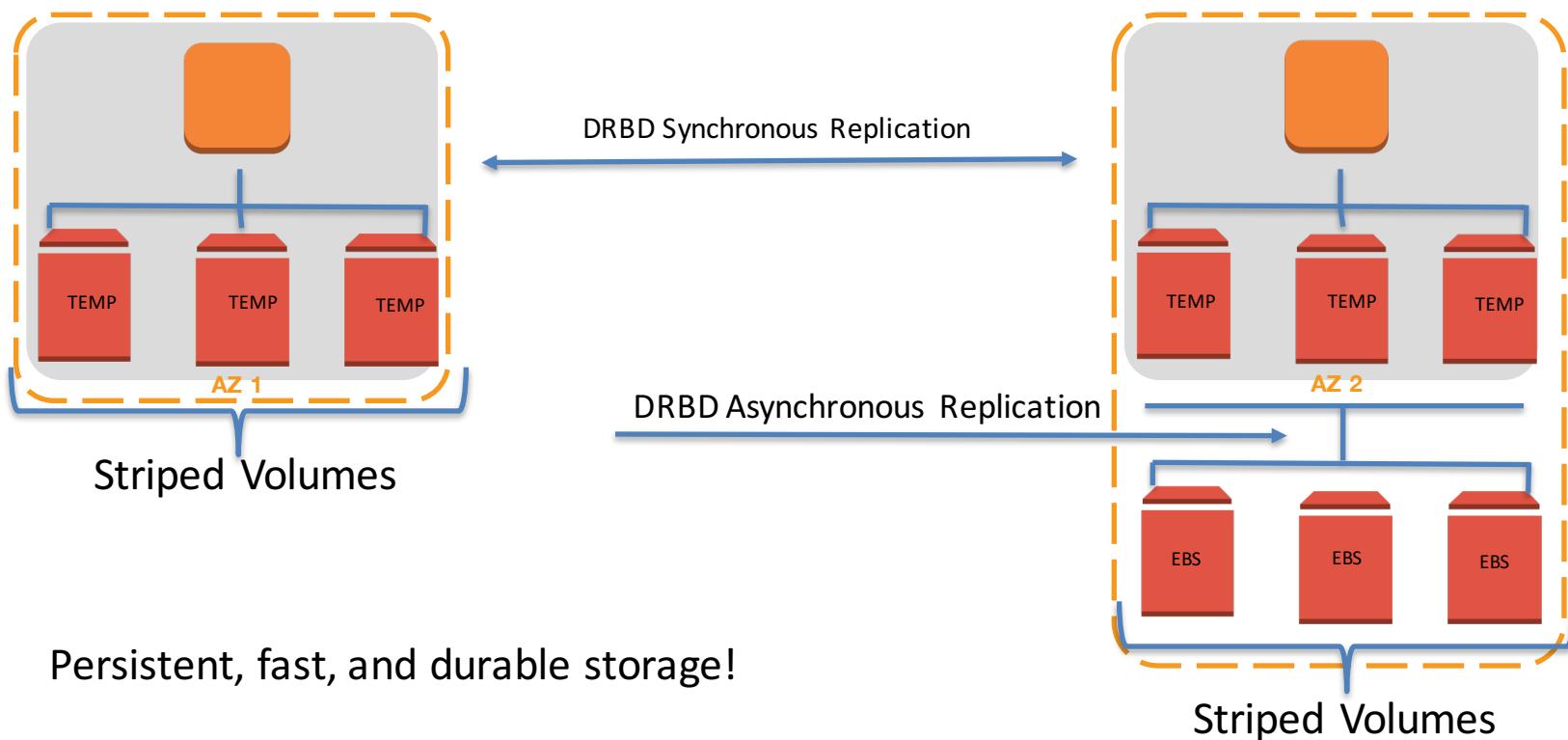
Problem: After 8-10 EBS volumes striped together your bottleneck becomes instance bandwidth. How can you get more throughput?

Solution: Use Instance-store backed instances, stripe the ephemeral storage devices attached for several hundred thousand IOPS depending on instance size

Instance Type	Instance Store Volumes
c1.medium	1 x 350 GB†
c1.xlarge	4 x 420 GB (1680 GB)
c3.large	2 x 16 GB SSD (32 GB)
c3.xlarge	2 x 40 GB SSD (80 GB)
c3.2xlarge	2 x 80 GB SSD (160 GB)
c3.4xlarge	2 x 160 GB SSD (320 GB)
c3.8xlarge	2 x 320 GB SSD (640 GB)
cc2.8xlarge	4 x 840 GB (3360 GB)
cg1.4xlarge	2 x 840 GB (1680 GB)
cr1.8xlarge	2 x 120 GB SSD (240 GB)
d2.xlarge	3 x 2000 GB (6 TB)
d2.2xlarge	6 x 2000 GB (12 TB)
d2.4xlarge	12 x 2000 GB (24 TB)
d2.8xlarge	24 x 2000 GB (48 TB)



RAID: Instance Store Instances With Ephemeral Storage





Linux Academy

Amazon Web Services

Multi-Region Architectures



Multi-Region Architectures

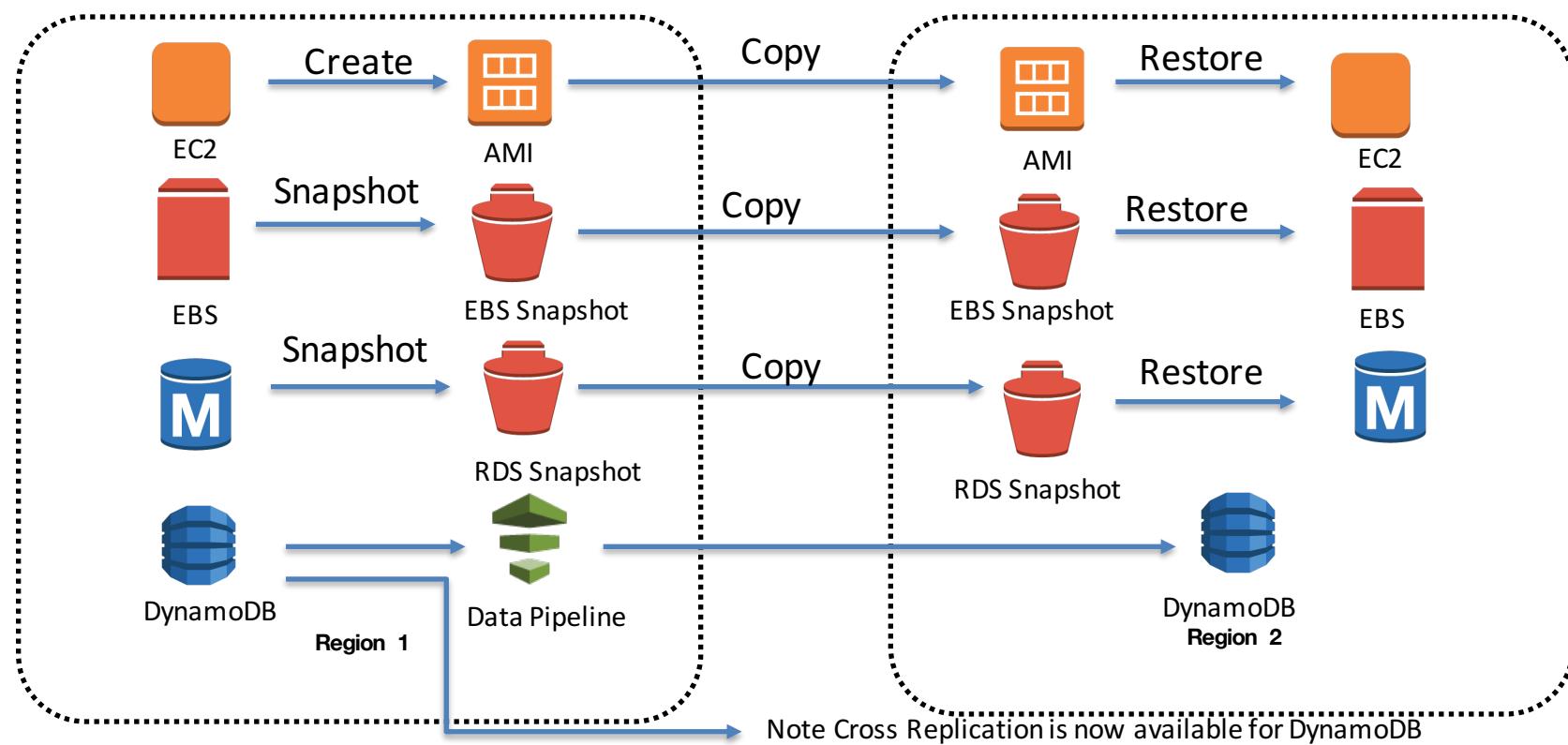
Multi-Region Architectures generally make use of Route 53 policies and complex policies to route traffic

- Active – Active multi-region designs
 - Latency based routing
 - Weighted based routing
 - GEO based routing
- Active – Passive multi-region designs (acceptable RTO/RPO)
 - Failover routing

Note: All policies should make use of health checks and for multi-region design you can often think Route 53 as a type of load balancer. Its job is to distribute traffic based off of some sort of “criteria” such as latency or weights.

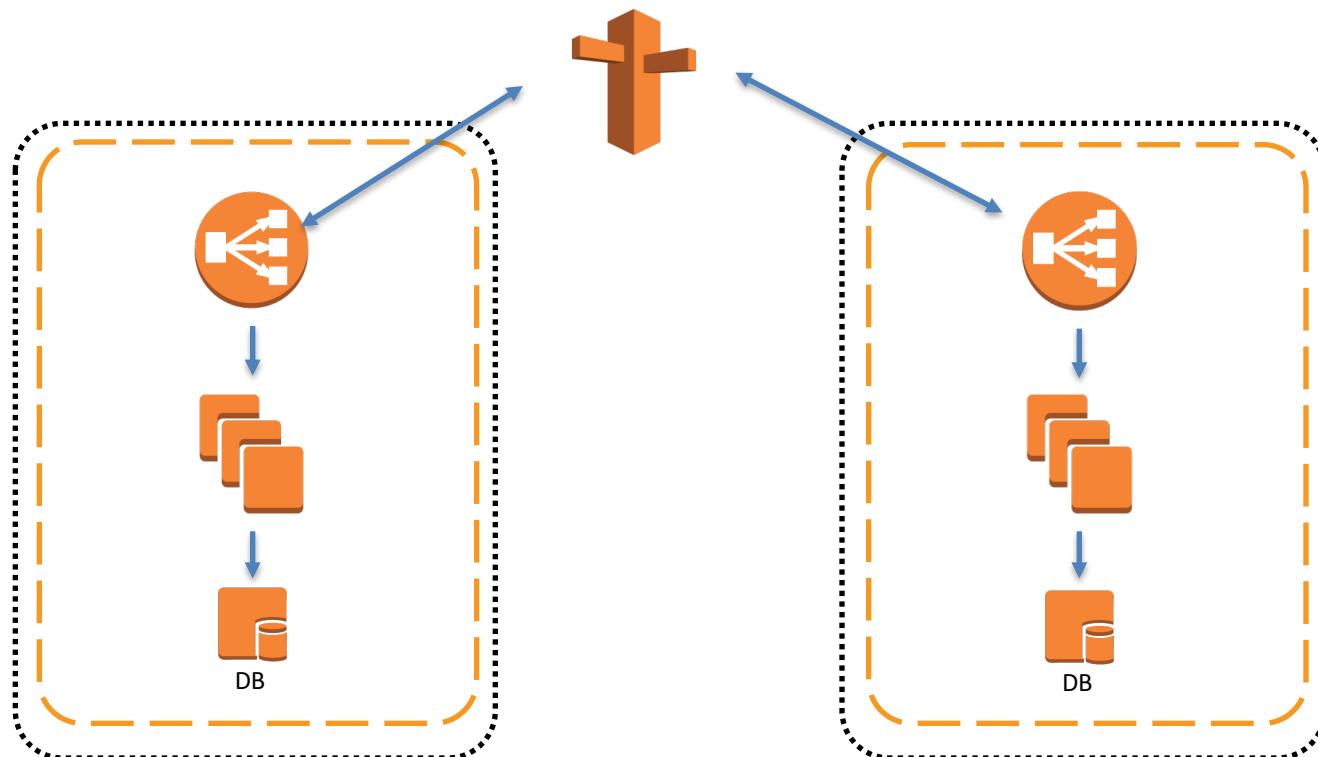


Multi-Region Architectures: First, copy the data





Multi-Region Architectures: Second, design the architecture for the region





Multi-Region Architectures: Third, Configure the routing policies

Latency Based Routing

- Route 53 will respond with DNS queries with resources that provide the best latency
- Used on EC2 instances or ELB
- Also works on private hosted zones

Geo Based Routing

- Respond with queries based off of users geographic location

Failover Routing

- Active-passive: If primary resource is not healthy then failover to secondary resource

Weighted Routing

- Send requests to record sets based off of weights

Weight for a given resource record set / sum of the weights for the resource record sets)

Multi-Region Architectures: Third, Configure the routing policies

Latency Based Routing

- Route 53 will respond with DNS queries with resources that provide the best latency
- Used on EC2 instances or ELB
- Also works on private hosted zones

Geo Based Routing

- Respond with queries based off of users geographic location

Failover Routing

- Active-passive: If primary resource is not healthy then failover to secondary resource

Weighted Routing

- Send requests to record sets based off of weights

Weight for a given resource record set / sum of the weights for the resource record sets)

Note: Hands-on examples in the video after this



Multi-Region Architectures: Complex routing – Nested record sets!

Problem: How can you send users to a geographic location or region based off of latency based routing or geographic based routing and then use weighted based routing among resources within that region?



Latency Based Routing Record
Name: domain.com
Region: eu-west-1
Set ID: 1
Evaluate target = yes

Latency Based Routing Record
Name: domain.com
Region: us-east-1
Set ID: 2
Evaluate target = yes

Weighted Resource Record Set
Name: eu-west-1-www.domain.com
Record Type : A
Value: 10.0.0.1
Set ID: 1
Weight: 15

Weighted Resource Record Set
Name: eu-west-1-www.domain.com
Record Type : A
Value: 10.0.0.2
Set ID: 2
Weight: 20

Weighted Resource Record Set
Name: us-east-1-www.domain.com
Record Type : A
Value: 10.0.0.1
Set ID: 1
Weight: 10

Weighted Resource Record Set
Name: us-east-1-www.domain.com
Record Type : A
Value: 10.0.0.2
Set ID: 2
Weight: 20

Health Check
Resource IP: 10.0.0.1

Health Check
Resource IP: 10.0.0.2

Health Check
Resource IP: 10.0.0.1

Health Check
Resource IP: 10.0.0.2



Latency Based Routing Record
Name: domain.com
Region: eu-west-1
Set ID: 1
Evaluate target = yes

Latency Based Routing Record
Name: domain.com
Region: us-east-1
Set ID: 2
Evaluate target = yes

Weighted Resource Record Set
Name: eu-west-1-www.domain.com
Record Type : A
Value: 10.0.0.1
Set ID: 1
Weight: 15

Weighted Resource Record Set
Name: eu-west-1-www.domain.com
Record Type : A
Value: 10.0.0.2
Set ID: 2
Weight: 20

Weighted Resource Record Set
Name: us-east-1-www.domain.com
Record Type : A
Value: 10.0.0.1
Set ID: 1
Weight: 10

Weighted Resource Record Set
Name: us-east-1-www.domain.com
Record Type : A
Value: 10.0.0.2
Set ID: 2
Weight: 20

Health Check
Resource IP: 10.0.0.1
FAIL

Health Check
Resource IP: 10.0.0.2
HEALTHY

Health Check
Resource IP: 10.0.0.1
HEALTHY

Health Check
Resource IP: 10.0.0.2
HEALTHY



Latency Based Routing Record
Name: domain.com
Region: eu-west-1
Set ID: 1
Evaluate target = yes

Latency Based Routing Record
Name: domain.com
Region: us-east-1
Set ID: 2
Evaluate target = yes

Weighted Resource Record Set
Name: www.domain.com
Record Type : A
Value: 10.0.0.1
Set ID: 1
Weight: 15

Weighted Resource Record Set
Name: www.domain.com
Record Type : A
Value: 10.0.0.2
Set ID: 2
Weight: 20

Weighted Resource Record Set
Name: us-east-1-www.domain.com
Record Type : A
Value: 10.0.0.1
Set ID: 1
Weight: 10

Weighted Resource Record Set
Name: us-east-1-www.domain.com
Record Type : A
Value: 10.0.0.2
Set ID: 2
Weight: 20

Health Check
Resource IP: 10.0.0.1
FAIL

Health Check
Resource IP: 10.0.0.2
FAIL

Health Check
Resource IP: 10.0.0.1
HEALTHY

Health Check
Resource IP: 10.0.0.2
HEALTHY



Linux Academy

Amazon Web Services High Performance Computing (HPC)



What is HPC?

HPC is the Aggregation of computing power to create much higher performance clusters and machines to perform large scientific, mathematical, and algorithmic based computations on data quickly.

- Financial computations
- Weather forecasting
- Anything that requires large amounts of compute usage



HPC On AWS – Instances

Specific instance types provide different types of compute power and networking speed. Depending on the amount of data needing to transfer in and out between nodes and the types of analysis required.

C4 instances – For compute heavy work loads

- Xeon E5-2666 v3 and up to 36 vCPUs of computing power
- EBS optimized by default for 500Mbps to 4,000 Mbps throughput to EBS

GPU Instances

- Used for 3D modeling and simulation (graphical heavy)
- NVIDIA GPUs



HPC On AWS

Placement groups:

- A placement group is a logical grouping of instances within a single Availability Zone. When using a placement group the application can take advantage of low-latency, 10Gbps network.



HPC On AWS

Placement groups: High throughput computing clusters using EC2 instances

- Instances launched as part of the group have high throughput network ability to each other
- AWS, when instances are launched at the same time, will attempt to locate the instances as physically close to each other as possible usually on the same host
- An already running instance cannot be added to a placement group
- Use the same instance type to help ensure the instances are located as close as possible. AWS groups physical hardware based off of instance type.
- If you receive a capacity error when launching an instance in a placement group, stop and restart the instances in the placement group, and then try the launch again.

HPC On AWS

Important: Auto Scaling can be used to launch instances in placement groups based off of CloudWatch metrics



HPC On AWS: SR-IOV (Enhanced Networking)

What is SR-IOV? SR-IOV is Single Root I/O Virtualization that creates enhanced networking abilities on instances ***which results in higher performance of packets per second, lower latency, and reduced jitter (jitter = noise on the wire)***

Supported Instance Types:

- C3, C4, D2, I2, M4, R3 (notice GPU instances are not listed!)

Supports only HVM virtualization and Amazon Linux has it on by default and in order to enable it the kernel module ixgbevf is required

Modinfo ixgbevf

Ethtool –l eth(n) to verify the kernel driver is being used will return driver: ixgbevf
X`



Linux Academy

Amazon Web Services

HPC Scenarios



HPC Scenarios

- Grid Computing (high throughput computing htc):
 - Locality is not a primary requirement
 - Work loads are more distributed
 - The size of the cluster can grow and shrink (auto scaling)
 - Often used with spot instances
 - Servers can be utilized over a wide area and even types of instances
 - Grid clusters should be designed for resilience
 - Are more often scaled horizontally
 - Loosely Coupled (Does not require tight communication between nodes)

Examples of Grid Computing:

- SETI@home
- 3D rendering jobs such as CAD or other GPU heavy work loads



HPC Scenarios

- Cluster Computing: Two or more instances connected together to support an application
 - Usually requires high node to node throughput
 - Most commonly assembled using the same type of instances
 - Usually uses placement groups or enhanced networking to satisfy the high network throughput requirement

Examples of Grid Computing:

- Weather computations and modeling
- Electromagnetic simulations
- Applications that require multiple nodes and low latency communication between the nodes



HPC Scenarios

Challenge: Knowing when to use specific architectures depending on the workload

- Does the workload require tight inter-communication between the nodes?
- Can a workload complete on a single node and benefit from auto scaling to handle increase in capacity?
- If a workload needs EC2 placement groups and SR-IOV enabled enhanced networking, then auto scaling and GPU instances are not a great solution and GPU instances cannot utilize SR-IOV enhanced networking
- Auto Scaling can be used to launch instances into placement groups



HPC Scenarios

- Grid computing workloads can benefit from high availability and resilience using tools such as auto scaling but a trade off is enhanced networking



Linux Academy

Amazon Web Services

Mitigating DDoS Attacks



DDoS Mitigation Strategies

Types of attacks:

- UDP Floods
- HTTP Floods (application attacks)
- SYN Flood (protocol attack)

Methods to Mitigate DDoS Attacks:

- Minimize The Attack Surface Area
- Scale to absorb the attack
- Safeguard exposed resources
- Learn normal behavior (IDS/WAFS)
- Create a plan for attacks

DDoS Mitigation Strategies: UDP Attacks

A UDP Attack occurs when UDP packets are sent to random ports on a host system or host systems. The hosts will attempt to look for applications listening on the UDP ports and, if no port is listening, it will respond with host unreachable. Increased number of requests will cause the system into forcing many ICMP packets which will eventually lead to the host being unreachable.

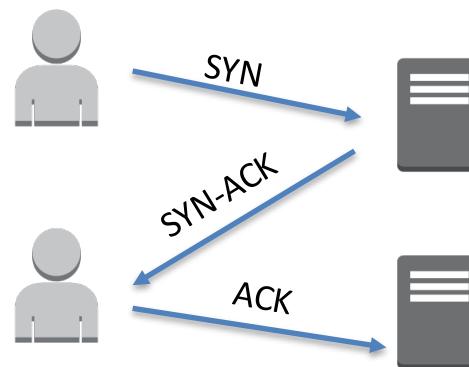
https://en.wikipedia.org/wiki/UDP_flood_attack



DDoS Mitigation Strategies: SYN Flood

Three-way TCP handshake:

1. Client requests a connection by sending a synchronized message to the server (SYN)
2. Server acknowledges the request by sending SYN-ACK back to the client
3. The client responds with an ACK and connection is then established



SYN: Synchronized packet

SYN-ACK: Synchronized Acknowledgement

ACK: Acknowledgement

“SYN Flood” occurs when the initiating client does not respond with an ACK (acknowledgement), causing the server to wait for an ACK leaving half-open connections on the server side reducing the available connecting resources.



DDoS Mitigation Strategies: Application Floods

Application floods using GET/POST requests occur against attacks on the layer 7 (application layer) of the network, often sending large amounts of HTTP GET/POST requests to overwhelm the application servers.



DDoS Mitigation Strategies

Minimize The Attack Surface Area

- Use ELB/CloudFront to distribute load to your applications
- Multi-Tier application architectures often provide layered protection against attacks

Scale to absorb the attack

- Enable Auto Scaling to handle increase load while you work to identify the source of attack

Safeguard exposed resources

- Use Route53 and aliases to hide the source IP of your resources

Learn normal behavior (IDS/WAFS)

Create a plan for attacks



DDoS Mitigation Strategies

CloudFront:

- CloudFront has built in abilities to absorb and deter DDoS attacks while still serving traffic to legit users. This is done as part of the CloudFront service and requires no additional configuration.
- CloudFront can scale to handle any increase in traffic which helps absorb attacks
- CloudFront uses filtering techniques to ensure that only valid TCP connections and HTTP requests are successful in passing through the edge locations
 - Solves UDP and SYN flood DDoS attacks



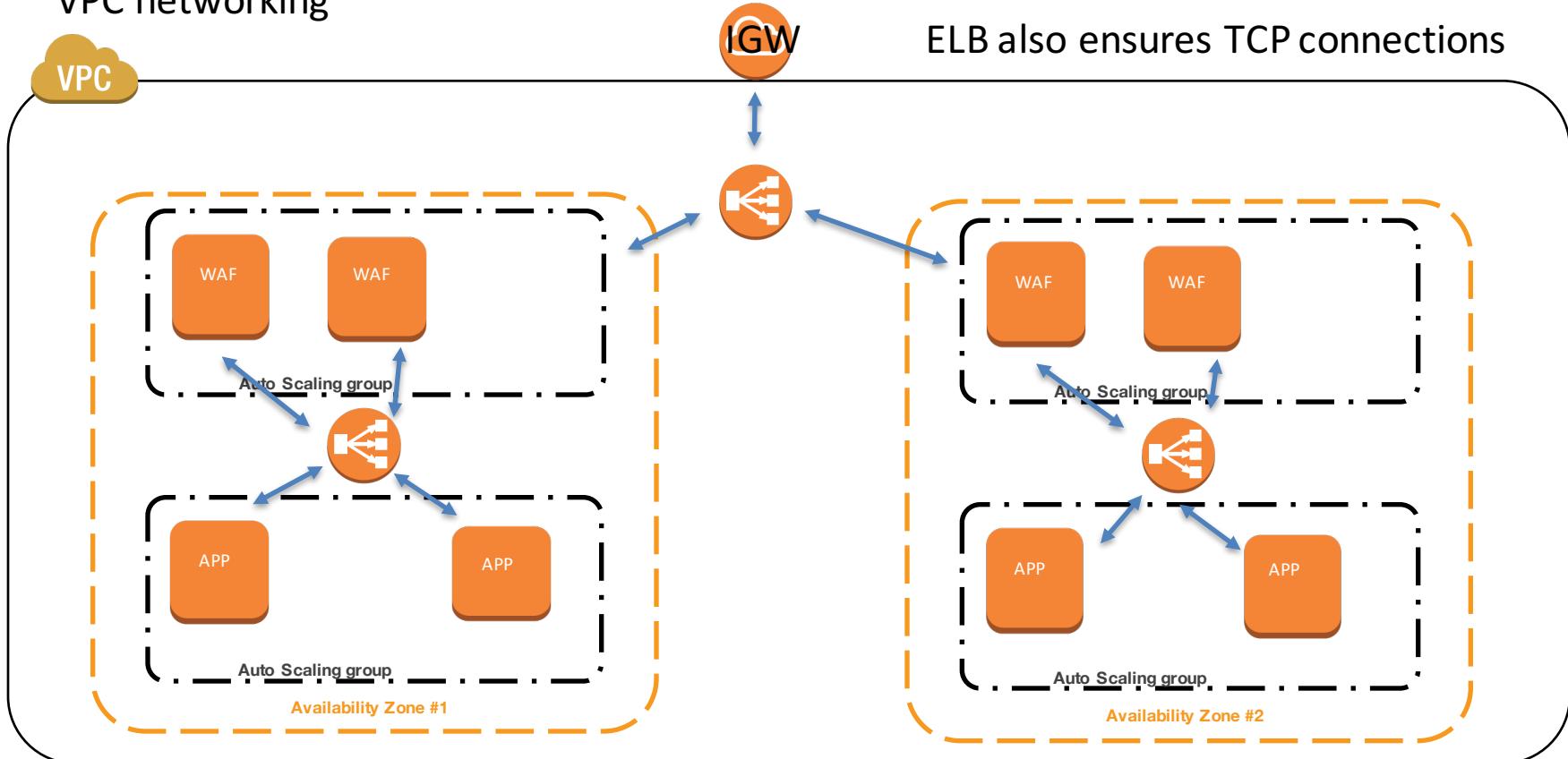
WAF (Web Application Firewall) controls input and shows what the traffic is doing and where it is coming from. Many WAFs have built in IDS (Intrusion Detection Systems) which analyze traffic data and looks for suspicious activity.

- Filters traffic and can identify/prevent injection attacks
- DDoS mitigation
- Malware protection
- Data loss prevention (identifies and traces data leaving your application)
- Detect suspicious activity and block/report the logging
- Used for greater insights into traffic flow for regulator reasons

WAFs can be part of the web server itself or it can sit in front of the webserver/ELB to filter the traffic and then forward to the application.

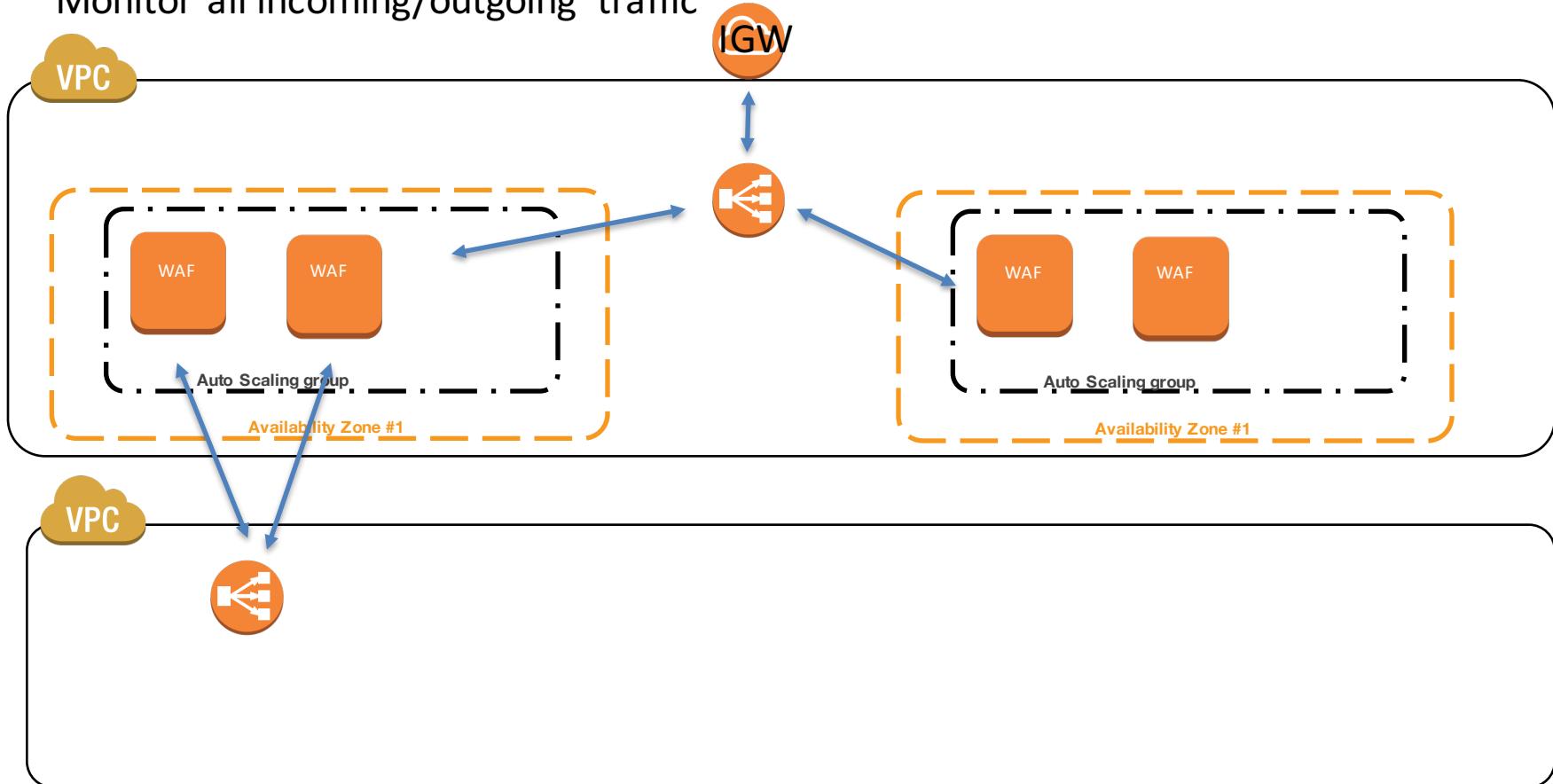


VPC networking





Monitor all incoming/outgoing traffic





Linux Academy

Amazon Web Services ELB Considerations



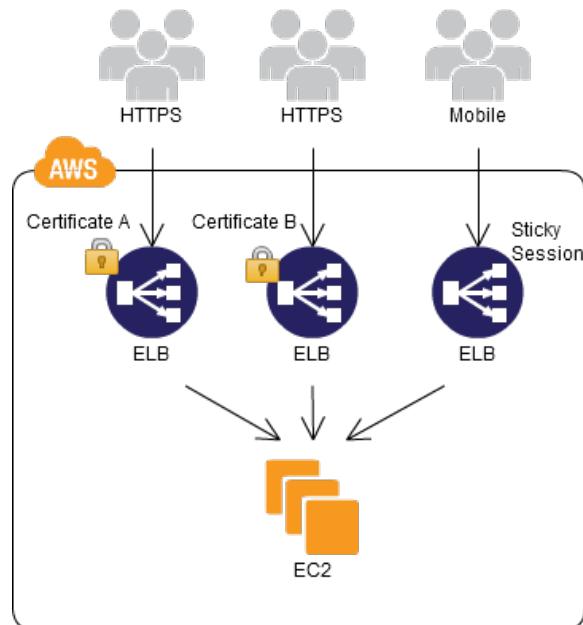
Elastic Load Balancer Considerations:

Scenario: BCJC's web application is multi-device compatible. Depending on the type of device, different ELB stickiness sessions and even SSL certificates are required. What is the best design pattern for presenting the different options to the arriving devices?

Example: Mobile devices being taken to a different ELB with different stickiness settings or desktop users going to different SSL certificates.



Elastic Load Balancer Considerations



Advantages:

- Different ELB behavior depending on the device
- Different SSL certificates for the same application



Understanding ELB Layer 4 vs. Layer 7 listeners

HTTP – Layer 7 request

HTTPS – Layer 7 request

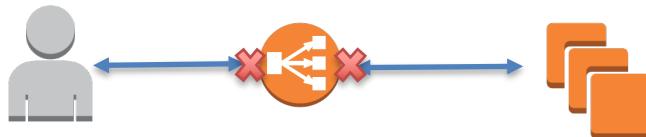
TCP (Ports 1- 65535) – Layer 4 request

SSL (Secure TCP) Layer 4 request

Layer 4 “forwards” the request to the backend instances

Layer 7 “terminates” and parses header information and makes a unique request

When a request is made to a load balancer, the load balancer intercepts the request and creates a new request on behalf of the client.





Understanding ELB Layer 4 vs. Layer 7 listeners

HTTP/HTTPS are layer 7 application requests. The ELB parses the header information when TCP termination occurs between the client and the ELB. It creates a new request and forwards it to the EC2 instances as if it was making the request itself.

ELB maintains an open TCP connection from the ELB to the backend instances when using HTTP/HTTPS layer 7.

TCP will not modify any header information and instead forwards the exact header information received from the client to the ELB.

- What happens if your application does not respond with common HTTP codes?
- Layer 4 does not work with ELB session stickiness



Understanding ELB Layer 4 vs. Layer 7 listeners

Edit listeners

The following listeners are currently configured for this load balancer:

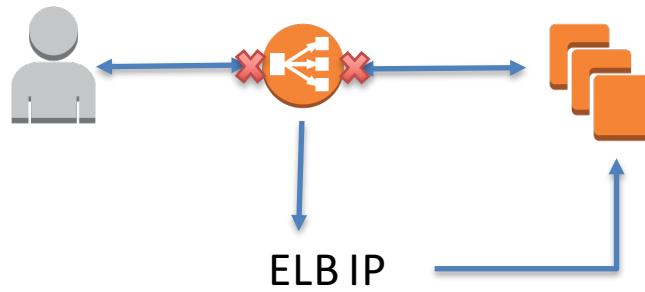
Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
Choose a protocol	80	HTTP	80
HTTP	80	HTTP	80
HTTPS (Secure HTTP)	80	HTTP	80
✓ TCP	80	TCP	80
SSL (Secure TCP)	80	TCP	80

Add



Forwarding Client IP Addresses To The EC2 Instances Behind The ELB

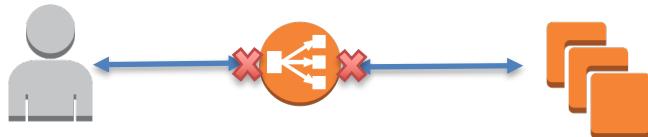
- When the ELB uses TCP to make the request to the EC2 instance on behalf of the client, the ELBs IP address will be sent to the EC2 instances and logged instead of the clients (think HTTP access logs)
- How can we forward the clients IP address?





Forwarding Client IP Addresses To The EC2 Instances Behind The ELB

- When the ELB makes the request to the EC2 instance on behalf of the client the ELBs IP address will be sent to the EC2 instances and logged instead of the clients (think HTTP access logs)
- How can we forward the clients IP address?
 - Use the CLI to configure **proxy Protocol** on the ELB; proxy protocol is used to carry connection information from the client making the request to the destination EC2 instances



Note: This only works with TCP configurations- NOT HTTP/HTTPS listeners



Forwarding Client IP Addresses To The EC2 Instances Behind The ELB

Problem: How do you get the client IP address when using the layer 7 HTTP/HTTPS listener on the load balancer since Proxy Protocol is only supported on TCP listener setup?

Solution: Modify your application code to send another header along with the request to the load balancer. The header needs to be X-Forwarded-For request header and will be passed through the ELB to the server with the clients IP address (if you add the clients IP address to the header in the code)

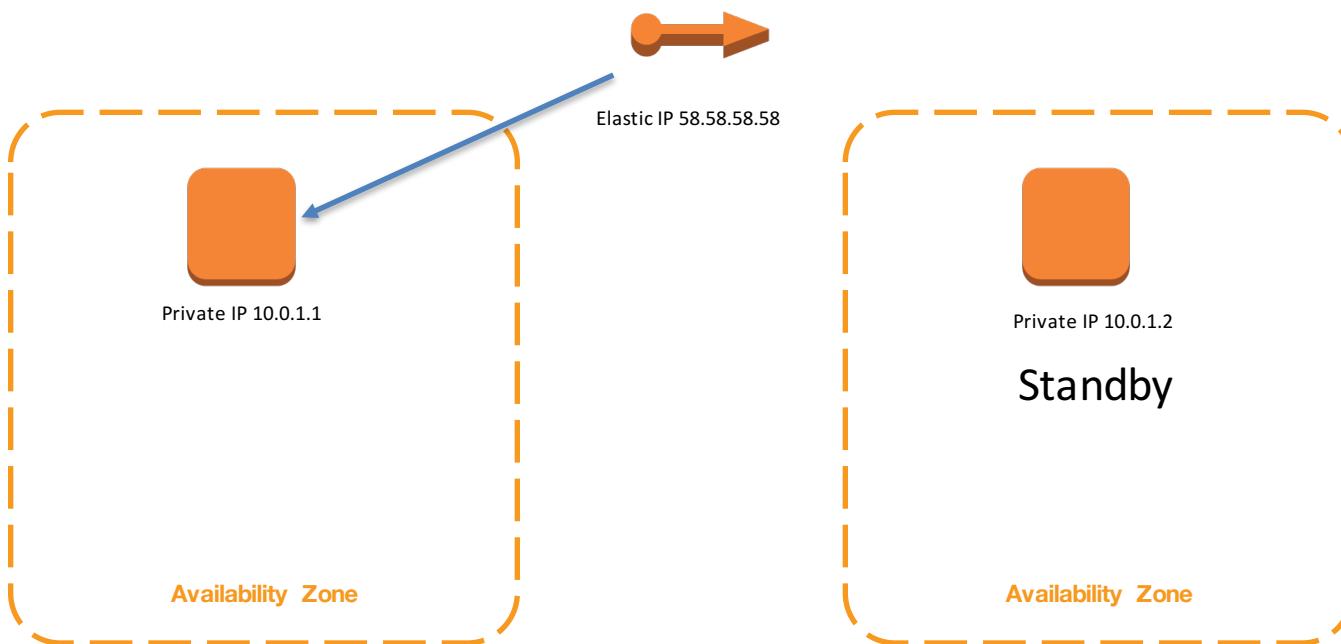


Linux Academy

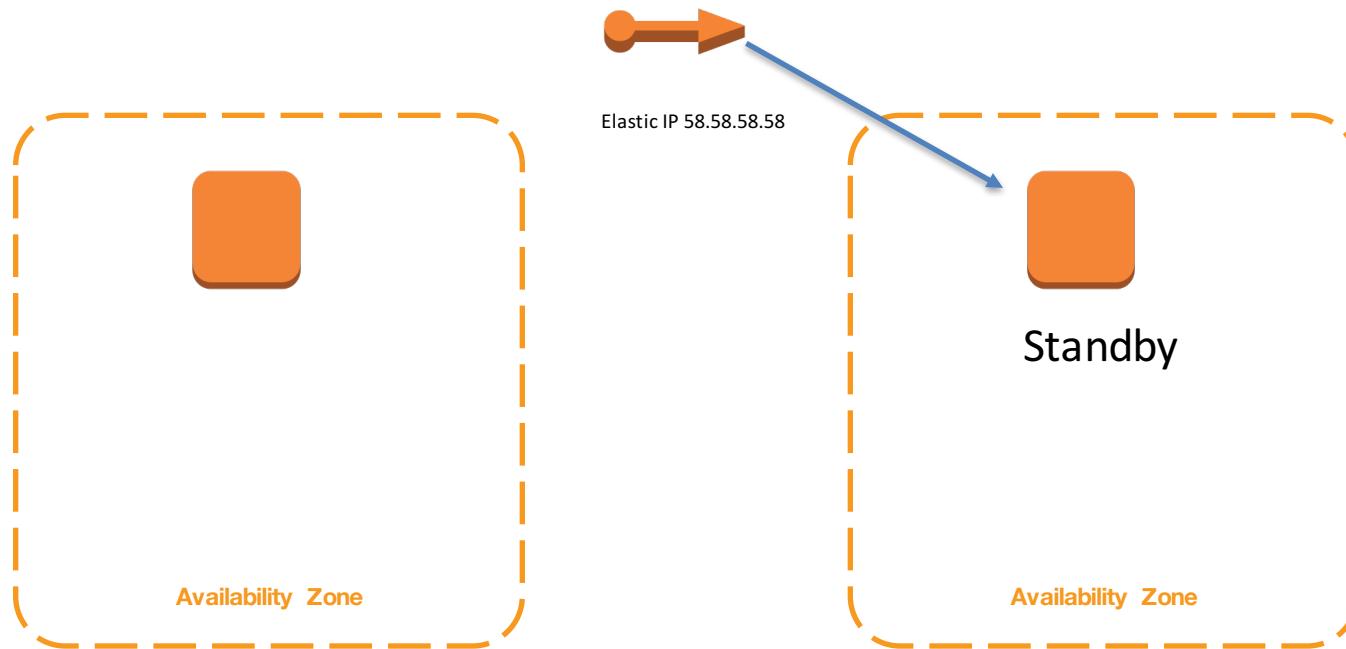
Amazon Web Services

Supporting Legacy Applications

Supporting Legacy Applications: Floating IP



Supporting Legacy Applications: Floating IP





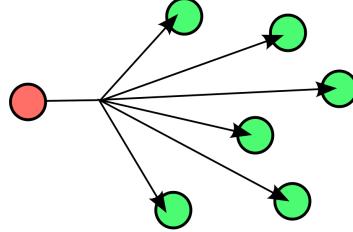
Supporting Legacy Applications: Floating IP

- Same concepts apply if you use an ENI to assign a static Private IP address
- In the event of failover disassociate the ENI and assign it to another instance
- Floating IP is also a solution if your software is licensed by MAC address

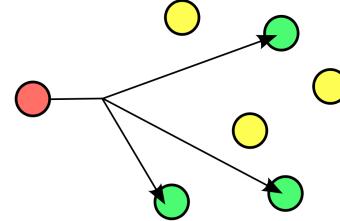


Supporting Legacy Applications: Multicast applications

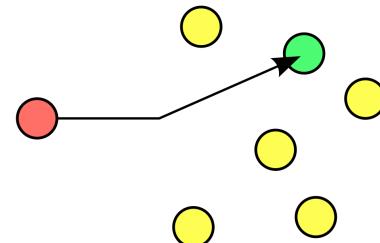
What is multicast? Multiple networks over the same network



Broadcast



Multicast



Unicast



Supporting Legacy Applications:

BCJC has an application that is built to work only on the same subnet using multicast type setup. How can BCJC design this application to be highly available across availability zones?

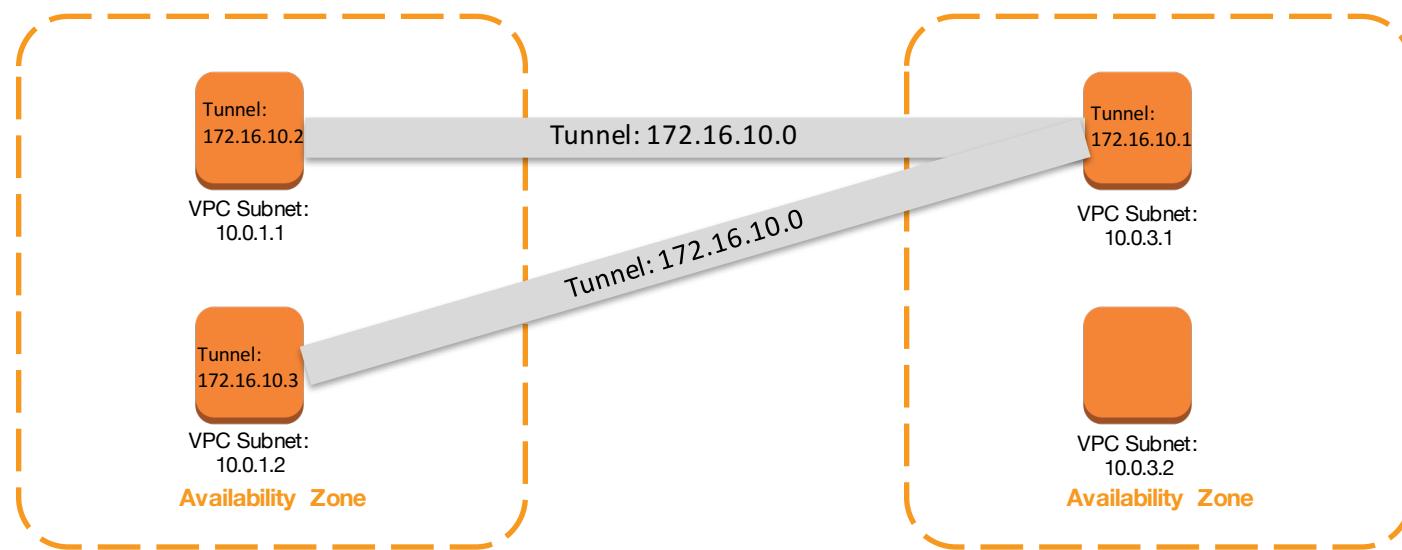
Multicast is NOT supported on AWS, why?

1. You cannot manage multiple subnets on a single interface on AWS
2. A subnet can only belong to one availability zone



How would you deploy legacy applications that require multicast?

- Create a Virtual overlay network that runs on the OS level of the instances
- VPC is unaware of what is happening
- Tunnel and a virtual network on the Operating System level of the EC2 instances
- The Virtual network CIDR ranges MUST be different than that of the VPC and the subnets are independent of the VPC
- Tunnels are typically created using different software applications such as
 - GRE or L2TP tunnel types
 - OpenVPN or Ntop's N2N application software to create the tunnel types





Linux Academy

Amazon Web Services Virtual Private Cloud



Linux Academy

Amazon Web Services Network Monitoring



VPC Network Monitoring Protecting Data Integrity

BCJC has hired third party contractors to work on applications that integrate with existing regulatory requirement (credit card data) data in BCJC's environment. While BCJC trusts the developers, there is an audit requirement to know what data, activity and the source for each is occurring in your environment or what data is leaving. The developers need full access to the AWS environment in order to perform the development tasks appropriately. What is best method and design to implement this type of security?

- Basically, how to know if a contractor is stealing the data, what the data is, and when/where it occurred
- Also know what commands are being issued in your environment and filter out bad potentially dangerous activity before it occurs in your environment.
- Allow developers “admin” access to the AWS tools



VPC Network Monitoring Protecting Data Integrity

- Create TWO AWS accounts
- Create a proxy for all data incoming/outgoing of the developer AWS account to the primary AWS account that can monitor or block the traffic



VPC Network Monitoring: Intrusion Detection Systems / Intrusion Prevention Systems

- Understand the limitations traditional intrusion detection systems allow you to put the system into promiscuous mode which allows for “sniffing” of traffic on your network that is intended for other machines/instances. This is a limitation of AWS and the hypervisor has it disabled so it will not deliver any traffic to instances that is not specifically addressed to the instances. Thus, ***promiscuous mode is not allowed.***
- Intrusion detection can work inline or by monitoring logs. There are a lot of log capabilities in AWS that we can use and then analyze for intrusion detection such as CloudTrail and S3 logs.
- Intrusion prevention actually identifies and “drops” suspect packets and this type of setup requires inline configuration

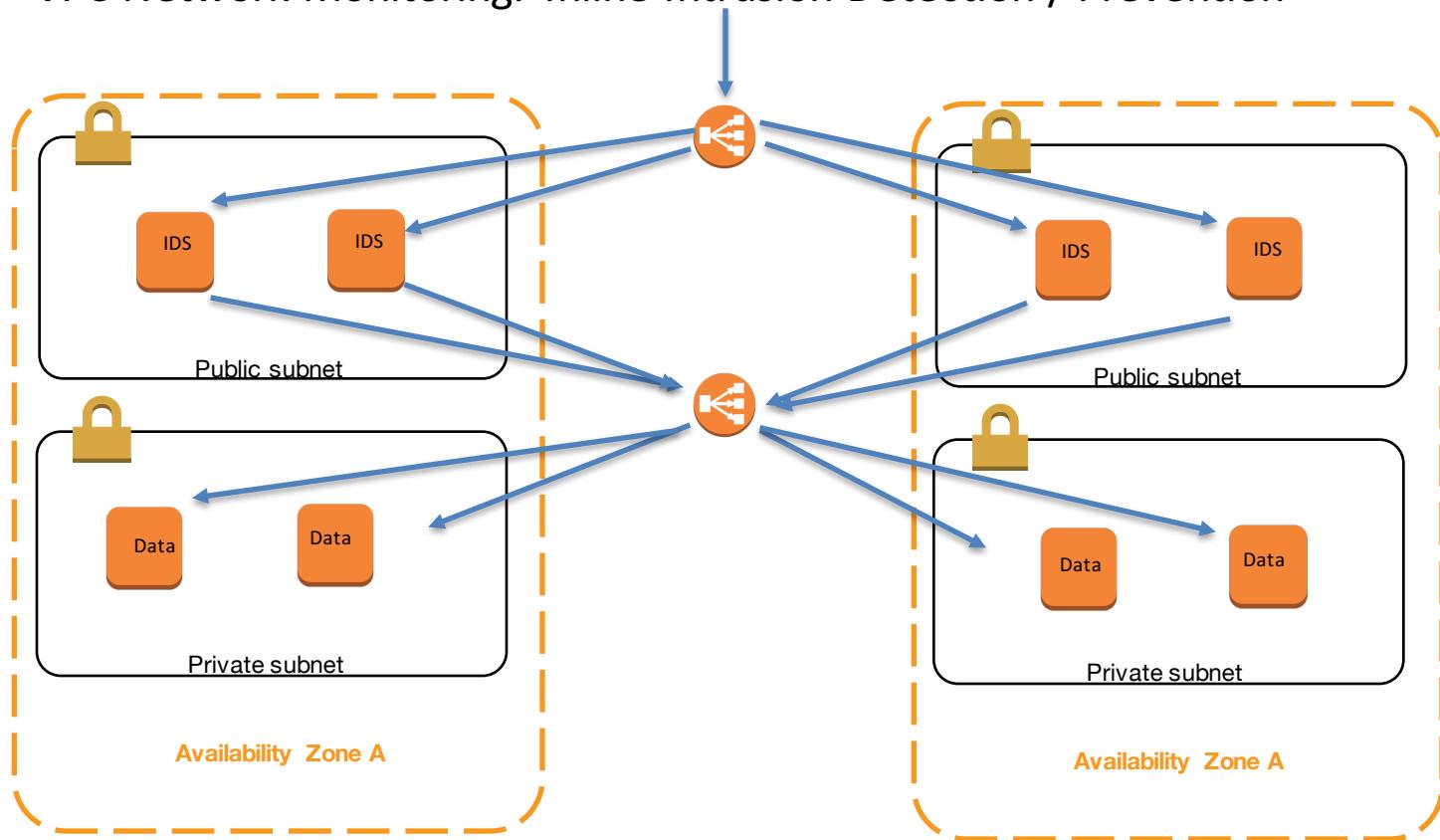


VPC Network Monitoring: IPS / IDS

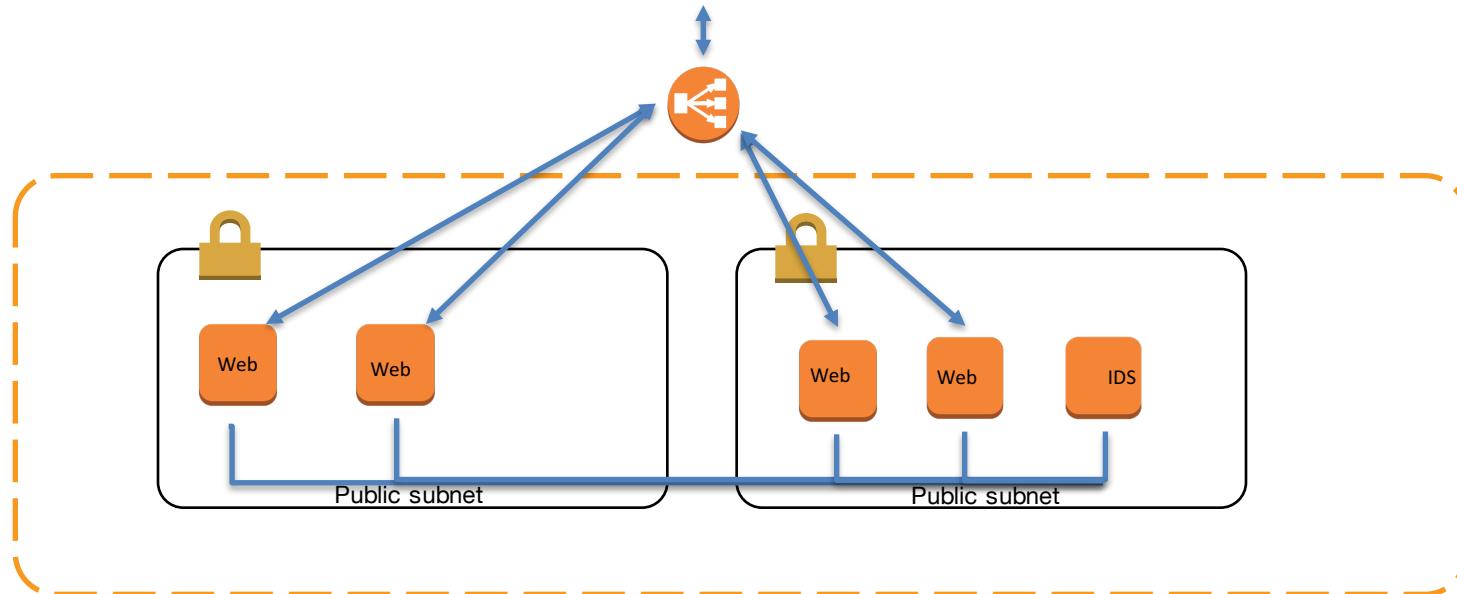
- Place an IDS inside of your cluster and allow your EC2 instances to send “copies” of the traffic to the instances for “monitoring” only.
- Place IDS software on your EC2 instances that deliver your primary “front end” application
- Place an IDS/IPS inline that automatically blocks/allows traffic to the destination instances.
 - Use your own inline monitoring in your AWS environment
 - Use a third party service to send traffic to an IDS/IPS provider that then redirects the traffic back to your application architecture



VPC Network Monitoring: Inline Intrusion Detection / Prevention



VPC Network Monitoring



An agent installed on the EC2 instances will send copies of the network traffic received on the EC2 instance to the IDS system.



Linux Academy

Amazon Web Services

Extending On-Premise Networks With VPN

Creating VPN Connections

VPN (Virtual Private Network) connections are used to extend on-premise data centers to AWS. Other uses include providing secure IPSec connections from on-premise computers/servers to AWS.

VPN connections create **private** connections to a VPC, giving on-premise machines access to internal VPC resources such as private IP addresses and Internal load balancers.

Key is to understand how to create VPN connections and how networking occurs with VPN connections

- Most corporate companies have hardware routers that are used to create VPN connections to the VPC and currently only hardware routers are supported by the VPC VPN option
- Software VPN such as OpenVPN can be configured on an EC2 instance

Creating VPN Connections: Key knowledge

- Understand how to create a hardware VPN connection
- Understand how to configure subnet route discovery between on-premise and VPC CIDR blocks



Creating VPN Connections: Steps For Configuring a hardware VPN

1. Create a VPG (Virtual Private Gateway) This is AWS side of the VPN connection
 - Create the VPG and attach it to the VPC; only one can be attached at a time

Create Virtual Private Gateway ×

A virtual private gateway is the router on the Amazon side of the VPN tunnel.

Name tag ?

Cancel **Yes, Create**



Creating VPN Connections: Steps For Configuring a hardware VPN

2. Create a customer gateway, this is the physical device or software on the client side of the connection

- Requires a public IP address to the on-premise router and an ASN number IF your enabling dynamic routing
 - Dynamic routing used with BGP will automatically discover on-premise and VPC CIDR blocks and create routes for the traffic to communicate between them
 - Static routes are required if BGP is not enable and requires a manual configuration

Create Customer Gateway ×

Specify the Internet-routable IP address for your gateway's external interface; the address must be static and can't be behind a device performing network address translation (NAT). For dynamic routing, also specify your gateway's Border Gateway Protocol (BGP) Autonomous System Number (ASN); this can be either a public or private ASN (such as those in the 64512-65534 range).

Name tag	<input type="text" value="data-center-a"/>
Routing	Dynamic (dropdown)
IP address	<input type="text" value="72.129.186.157"/>
BGP ASN	<input type="text" value="65000"/>

Cancel Yes, Create



Creating VPN Connections: Steps For Configuring a hardware VPN

3. Create a VPN connection in the VPC

- VPN connection is where the configuration for either static or dynamic routes is configured

Create VPN Connection

Select the virtual private gateway and customer gateway that you would like to connect via a VPN connection. You must have entered the virtual private gateway and your customer gateway information already.

Name tag i

Virtual Private Gateway ▼

Customer Gateway (radio button) Existing (radio button) New
 ▼

Specify the routing for the VPN Connection [\(Help me choose\)](#)

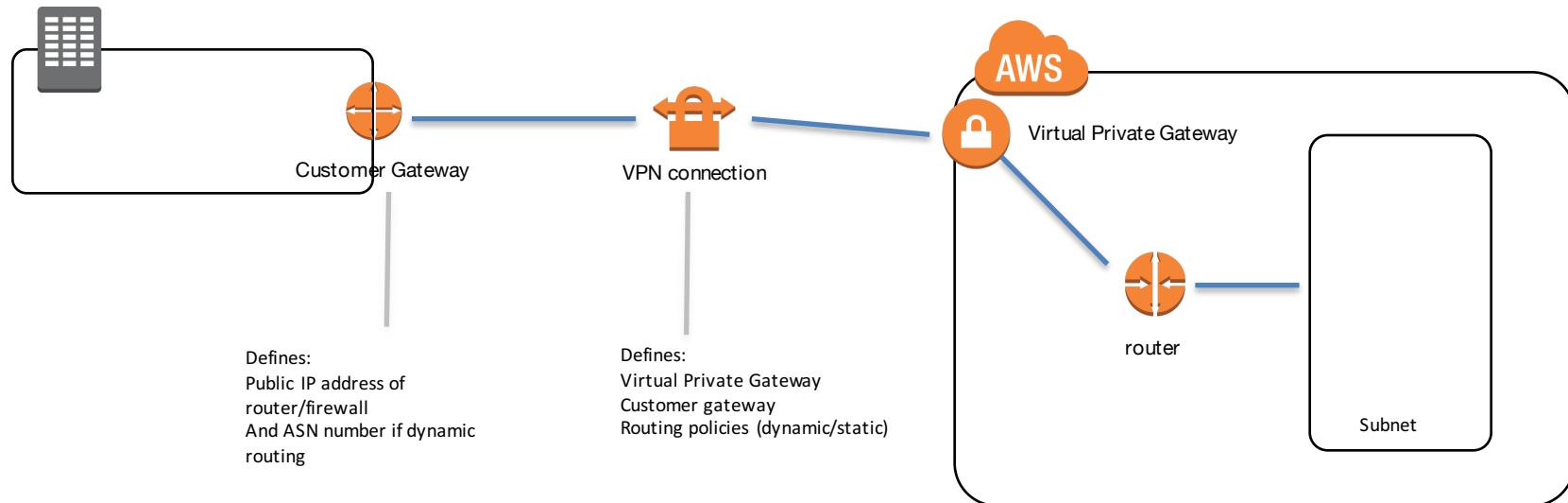
Routing Options (radio button) Dynamic (requires BGP) (radio button) Static

VPN connection charges apply once this step is complete. [View Rates](#)

[Cancel](#) Yes, Create



Creating VPN Connections: Steps For Configuring a hardware VPN



Creating VPN Connections: Steps For Configuring a hardware VPN

Creating redundancy

- Each VPN has two tunnels associated with it that can be configured on the customer router
- The single point of failure then becomes the single customer router
 - Create a second customer gateway and a second on-premise router for configuration



Linux Academy

Amazon Web Services Security Zones

Security Zones

What if you're running multiple applications in an AWS environment?

- Separate by creating multiple VPCs one for each zone (if this high level separation is allowed and the apps do not need to communicate)
- For apps/instances that need communication, use segmentation tools available to ensure only traffic required is flowing in and out of zones
 - Security Groups
 - NACLs
- Segment environments based off of CIDR block ranges and create NACL rules that allow traffic to specific subnets/security groups based off of those CIDR block ranges; this ensures inter-zone communication is allowed from only specific locations



Linux Academy

Amazon Web Services

Understanding AWS IP Subnet Reservations



The first 4 and last 1 IP addresses of a given subnet are not available due to AWS reservations of the IP addresses for networking purposes.

Summary: AWS reserves 5 IP addresses of each subnet for networking purposes.

	Addresses	Available
/28	16	11
/27	32	27
/26	64	59
/25	128	123
/24	256	251
/23	512	507
/22	1024	1019
/21	2048	2043
/20	4096	4091
/19	8192	8187
/18	16384	16379
/17	32768	32763
/16	65536	65531



Linux Academy

Amazon Web Services

AWS Direct Connect



Why Direct Connect?

Reduce network costs

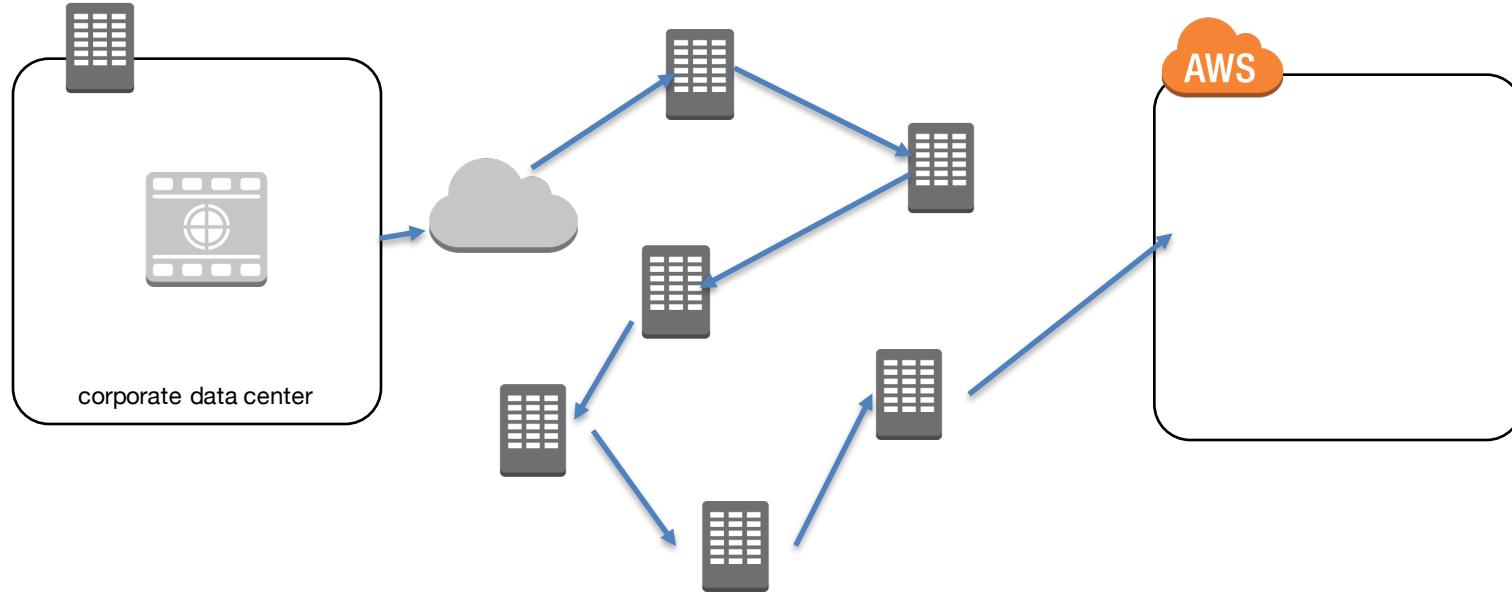
- Reduce bandwidth commitment to corporate ISP over public internet
- Data transferred over direct connect is billed at a lower rate

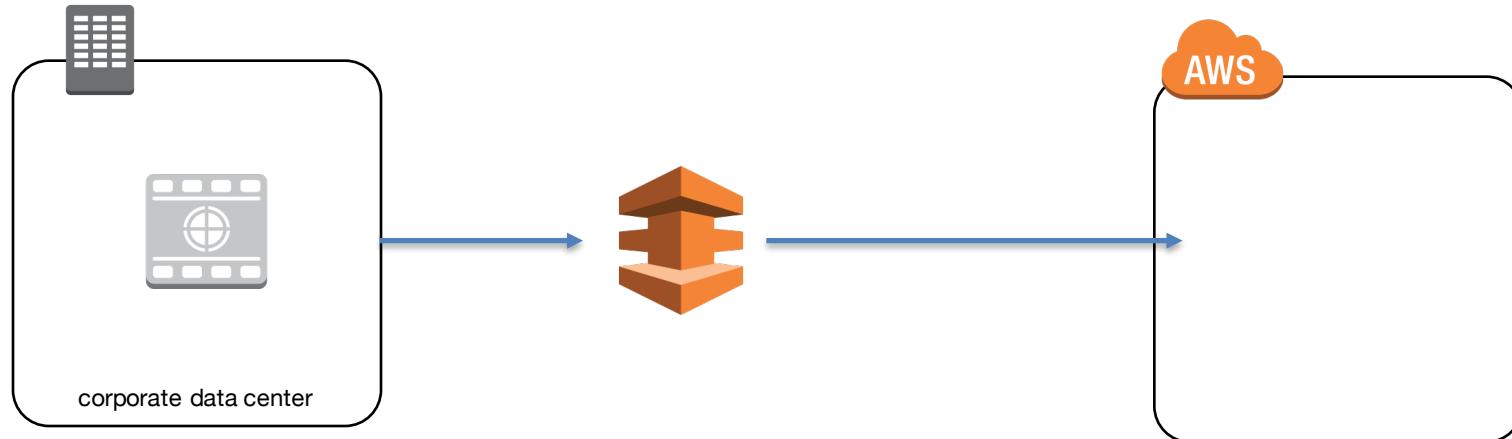
Increase network consistency

- Dedicated private connections reduce latency rather than sending the traffic via public routing

Dedicated private network connection to on-premise

- Connect the direct connect connection to a VGW in your VPC for a dedicated private connection from on-premise to VPC
- Use Multiple VIF (Virtual Interfaces) to connect to multiple VPCs





Does not require hosting any router/hardware at the Direct Connect Partner location, only requires a Direct Connect location and a participating backbone provider.



Using the Direct Connect service to connect to AWS you provision:

Private Virtual Interfaces: Interfaces with an Amazon Virtual Private Cloud (VPC) with automatic route discovery using BGP and requires a public or private ASN number

- Can only communicate with internal IP addresses inside of EC2
- Cannot access public IP addresses as Direct Connect is NOT an internet provider
- This is a dedicated private connection which works like a VPN
- Use two Direct Connect connections for active-active or active-failover availability
- Can also use VPN as a backup to direct connect connections
- Create multiple private VIFs to multiple VPC's at a time



Using the Direct Connect service to connect to AWS you provision:

Public Virtual Interfaces: Use Direct Connections to AWS and connects to public AWS endpoints for any AWS service such as DynamoDB or Amazon S3

- Requires public CIDR block range
- Still has consistent traffic as it is sent over your dedicated network to the Direct Connect partner at the partners connection to AWS



Cross-network Connection (Cross Connect) – Physical connection between your network and the Direct Connect authorized partner which then handles the routes and connections to AWS networks.

An AWS Direct Connect location provides access to the AWS region it is associated with. It does not provide access to other AWS regions. However, there are methods to connect to additional AWS regions discussed in the next lesson.



Linux Academy

Amazon Web Services

AWS Direct Connect Accessing A Remote AWS Region



A Direct Connect into an AWS Partner Direct Connect provider will only connect to the closest region or associated AWS region to the provider.

What if you're creating multi-region design and have a need for a more reliable network connection?

- Create a public virtual interface to the remote regions public endpoints and use VPN over the public virtual interface to protect the data

Note: While you will not have a private direct connect connection your data will still utilize AWS backbone networks for a better connection to the remote region. By creating a VPN you are creating your own private network to internal AWS VPC resources.



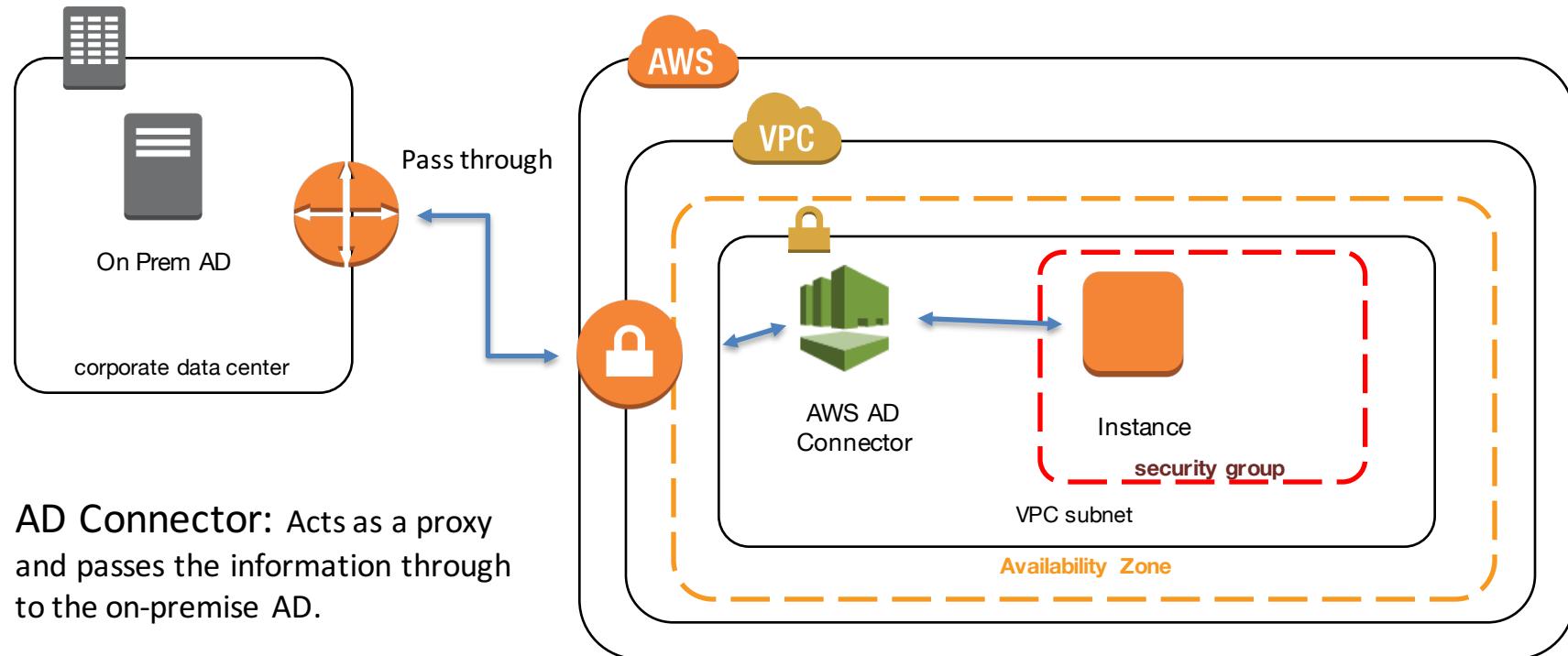
Linux Academy

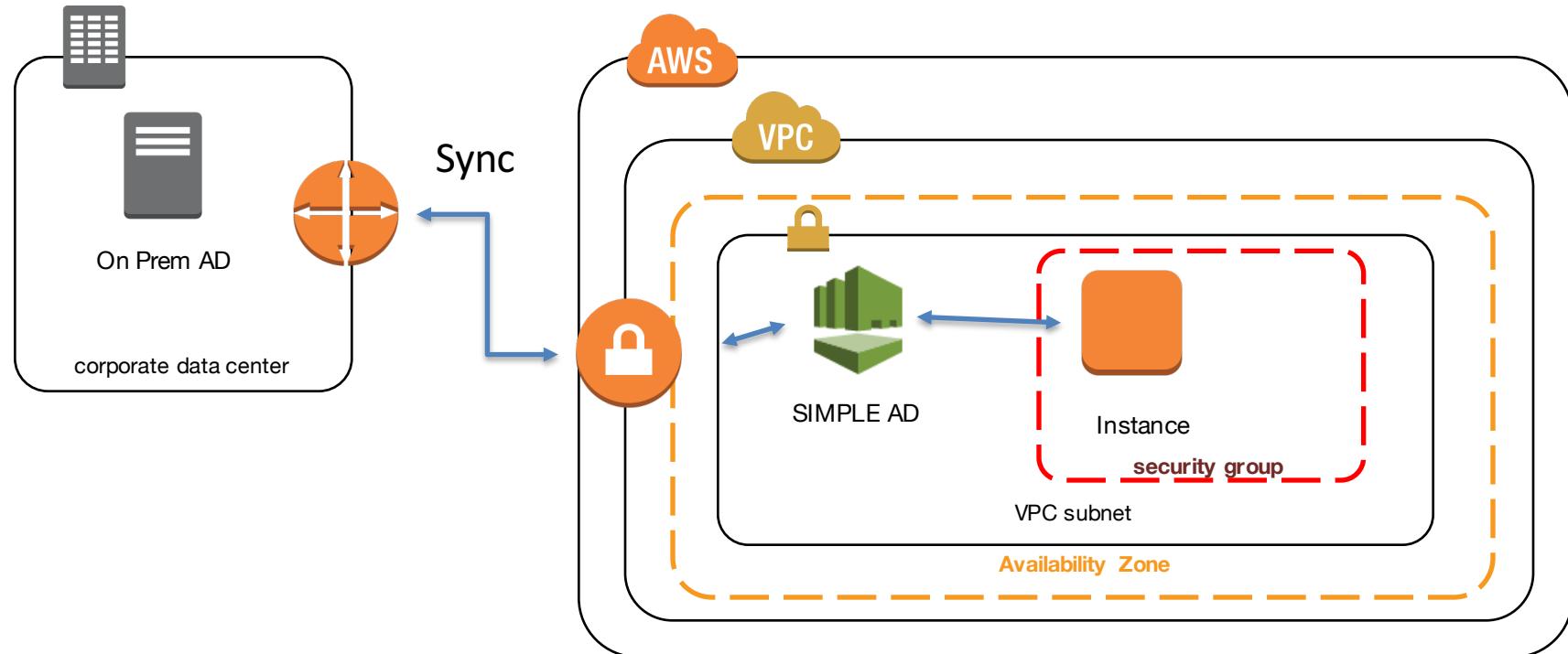
Amazon Web Services

Hybrid Data Center With Directory Service

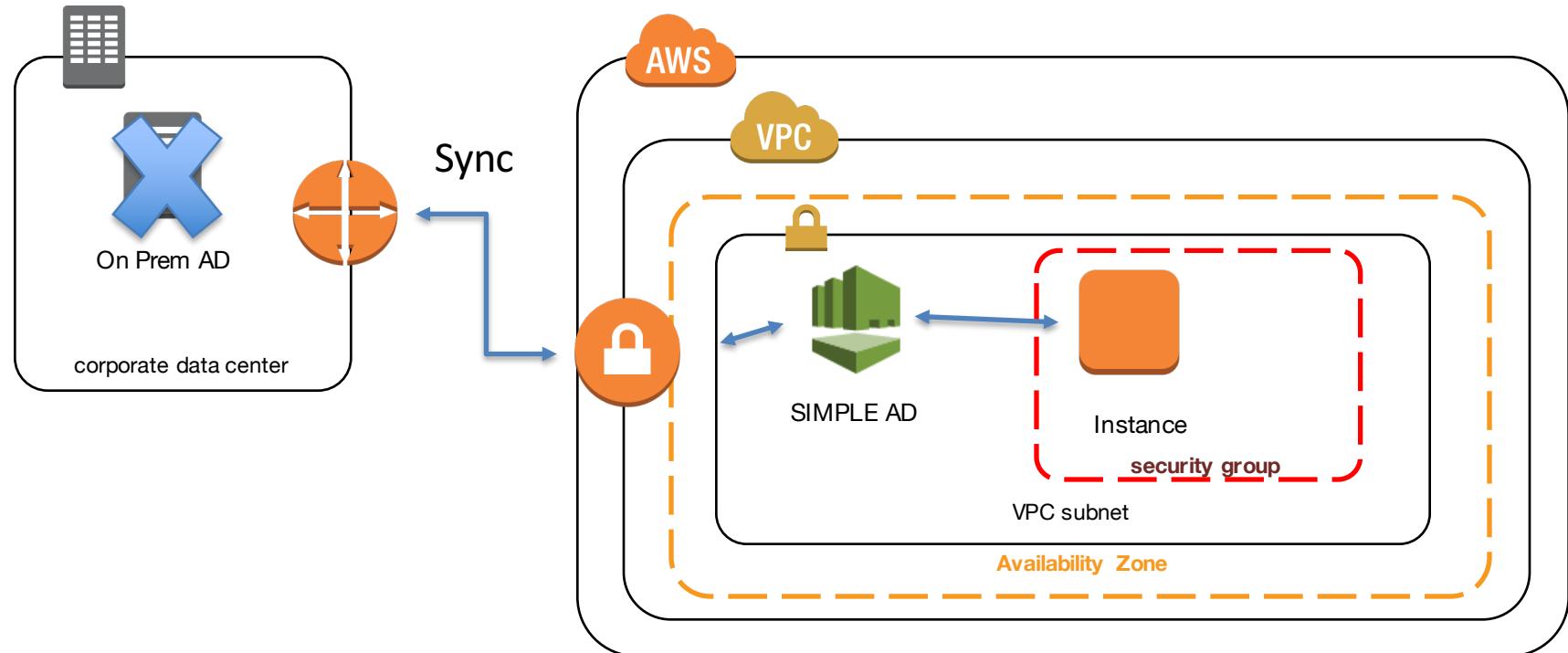


- Different than temporary credentials
- This is giving apps access to active directory in the AWS environment or on-premise environment
- As your applications migrate to amazon, you might want those apps to access the AD roles/accounts
- BCJC wants to connect on-premise to the cloud using existing credentials; in other words expose existing AD to the cloud
 - AWS Directory Service
 - Ad connector (essentially a hosted proxy service, no caching) instances on AWS that need access to on-premise AD will proxy through the AD connector down to the on-premise AD server; nothing is stored on the connector
 - Simple AD – a fully hosted AD on amazon; you would setup another master controller and « sync » to on-premise and then maybe eventually move fully on-premise





Simple AD: A full Active Directory Service which, in order to use on-premise credentials, you would setup active directory sync



Simple AD: Remove the AD sync to on premise and instead use simple AD as your AD on premise as well



Linux Academy

Amazon Web Services

Amazon ElastiCache

Amazon ElastiCache

ElastiCache is an in-memory hosted caching solution provided by AWS.

ElastiCache supports two types of caching engines at this time.

Memcached – Common caching appliance which uses a DB source such as mariaDB/MySQL as the persistent storage and fills frequently accessed objects inside of in-memory memcache.

Redis – Redis acts more like a replacement for the DB server and instead maintains its own persistence and is used for certain types of application functions.

Note: Each caching engine provides different methods for high availability, backup, usage, and migration.

Amazon ElastiCache: When to cache?

Cache data that is “static” and is also frequently accessed

- Profile data

Storing infrequently accessed data doesn’t equate to cost savings or much performance savings but will fill up your available cache memory

Cache expensive queries or slow queries with joins that run across multiple tables, these are considered hardware intensive and expensive.

Cache data is “stale” it doesn’t change frequently and would require flushing for new data to appear.

- Redis caching engine is a little different as it uses the in-memory storage for actual data storage and only writes persistence to snapshots or data files frequently



Amazon ElastiCache: When to cache?

Is the query being made against the database slow or expensive?

- Large join showing the results of comments on a wordpress thread

Is the resulting data frequently accessed?

- Social media profile or even a course listing

Is the data “static” or does it change frequently?

- Video count on Linux Academy front page



Amazon ElastiCache: Caching Strategies

Lazy Loading

- Application attempts to receive data from the cache nodes
- If no data is available then the cache nodes return null
- Application receives the data from the database (disk based db)
- Application then updates the cache
- Only requested data is cache, so the cache is not filling up the memory with non-requested data and taking resources
- Node failures aren't a huge issue because if a node fails the request just goes to the DB

Lazy loading can be expensive if there is a cache miss. This is important in determining if an item is infrequently accessed and should be cache or not. If it is infrequently accessed it will be less expensive to just read from the DB and bypass cache.



Amazon ElastiCache: Caching Strategies

I

- Ensures data is never stale and is always up to date (does not require expiration)
- Each DB write involves two steps, write to db and write to cache can become expensive by increasing latency
 - Good strategy for applications that do not have a lot of writes
- Downsides:
 - Lots of data is stored in memory that may not be frequently accessed
 - If a node is spinning up it could miss writing and cause missing data

Amazon ElastiCache: Caching Strategies

Adding TTL: Essentially, cache expires after the TTL (Time To Live) which can be applied to both lazy loading and write through to manage cache resources.

- Number of seconds until a key expires (caching is a key:value store)

Note: Anytime you access data from in-memory storage, it is ephemeral but is MUCH faster than reading from a disk. Remember, the type of data you cache depends on the caching engine your using, the use case, and what it takes to load the data into cache.



Linux Academy

Amazon Web Services

Amazon ElastiCache: Memcached



Amazon ElastiCache: Memcached

Memcached is a more traditional caching mechanism which is placed in front of a DB source.

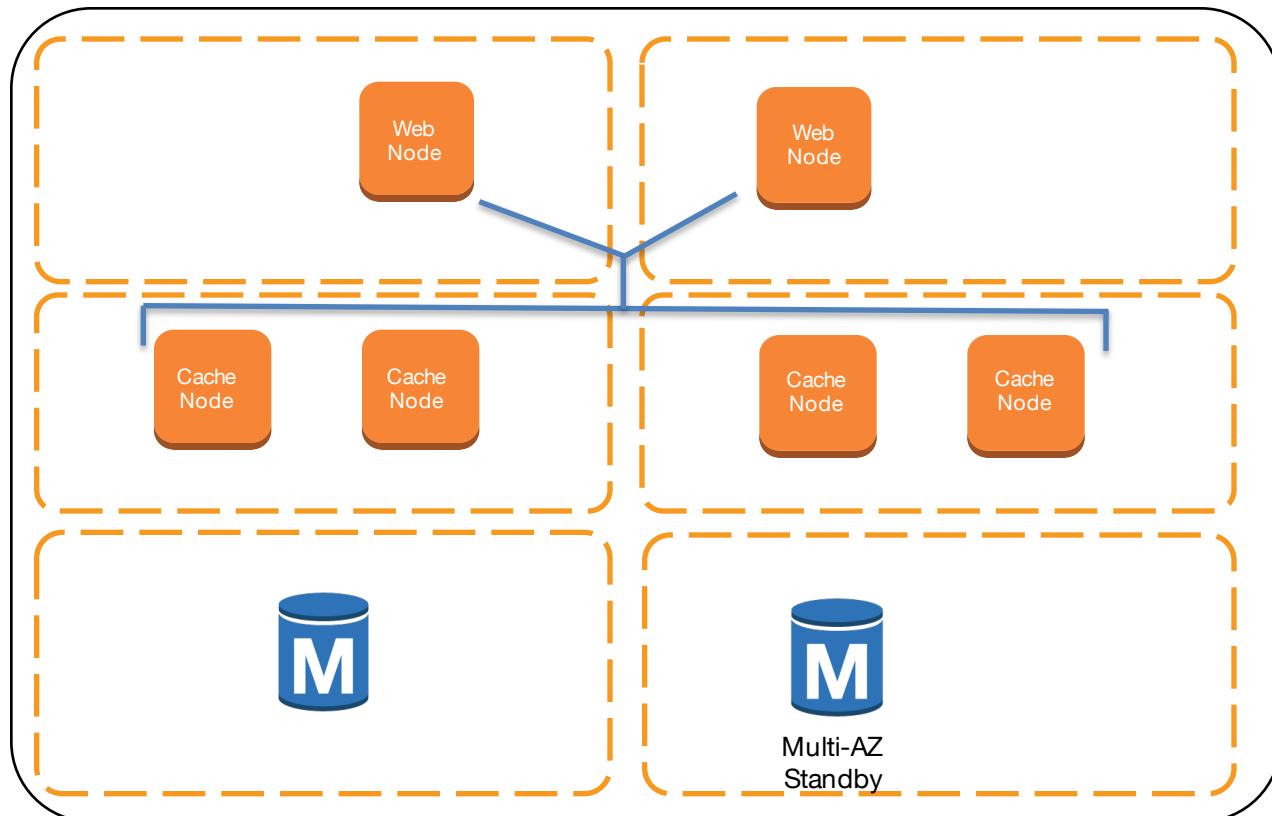
- Does not manage its own persistence
- Can be run in a cluster of nodes
- Does not have backup abilities
- Scales by adding more nodes to the cluster

Populate cache:

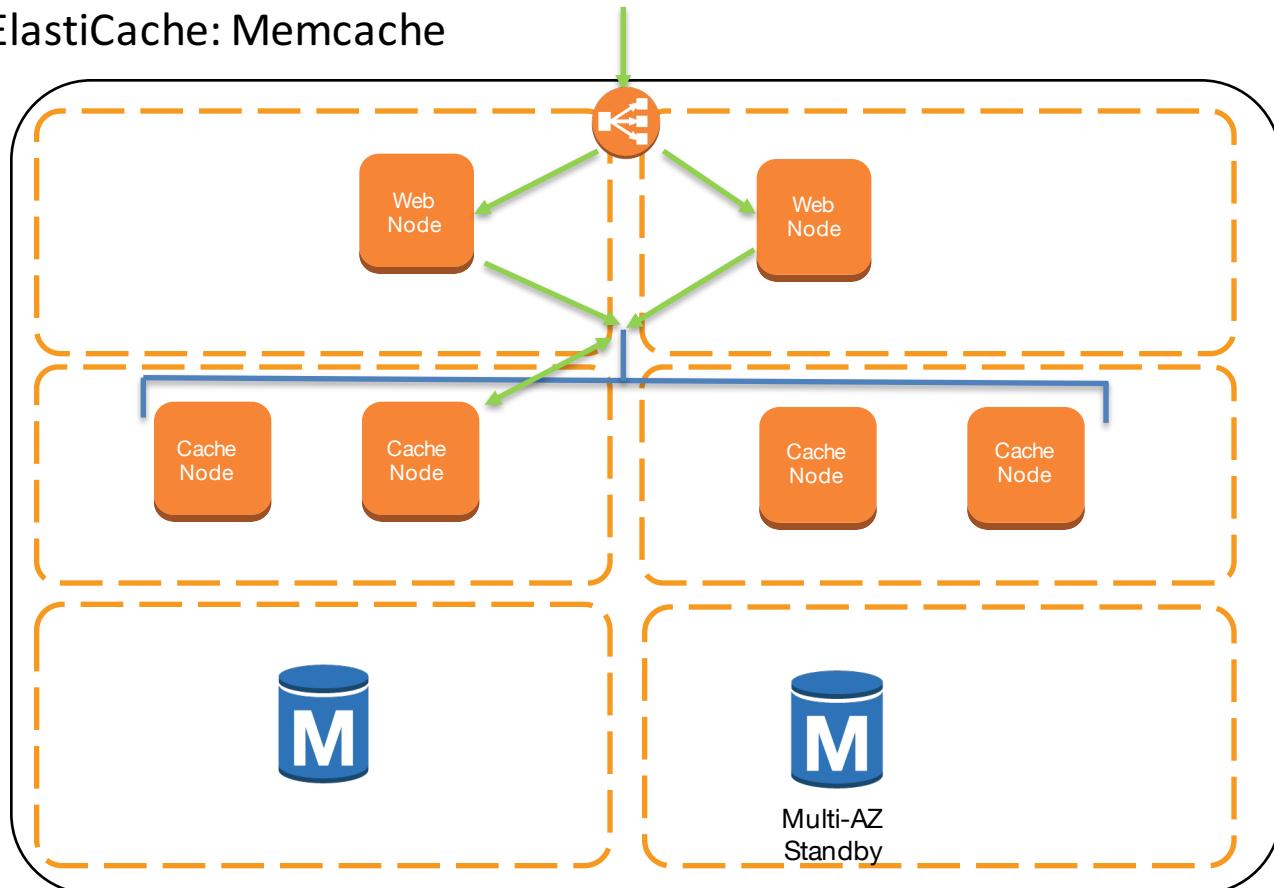
- Write through
- Lazy loading



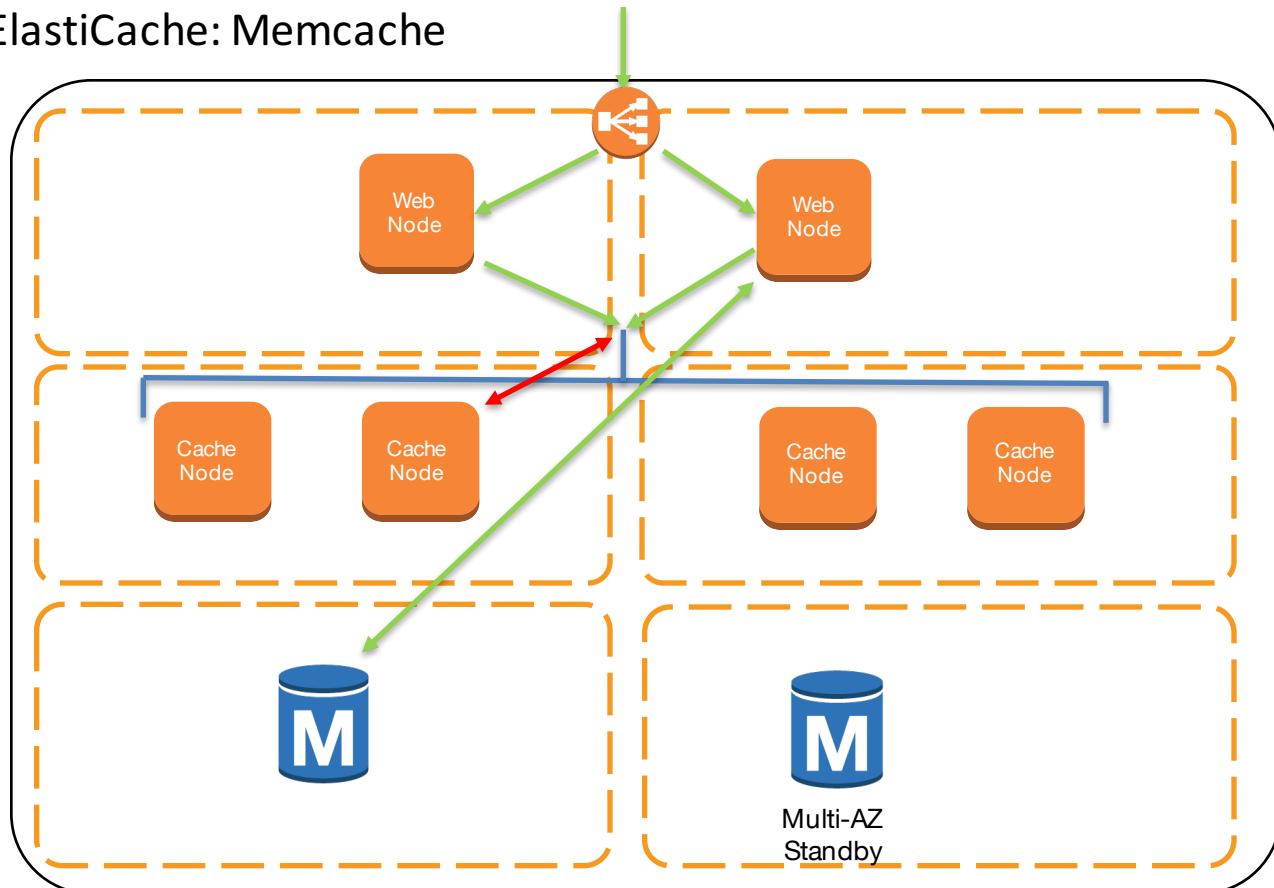
Amazon ElastiCache: Memcache Lazy loading example



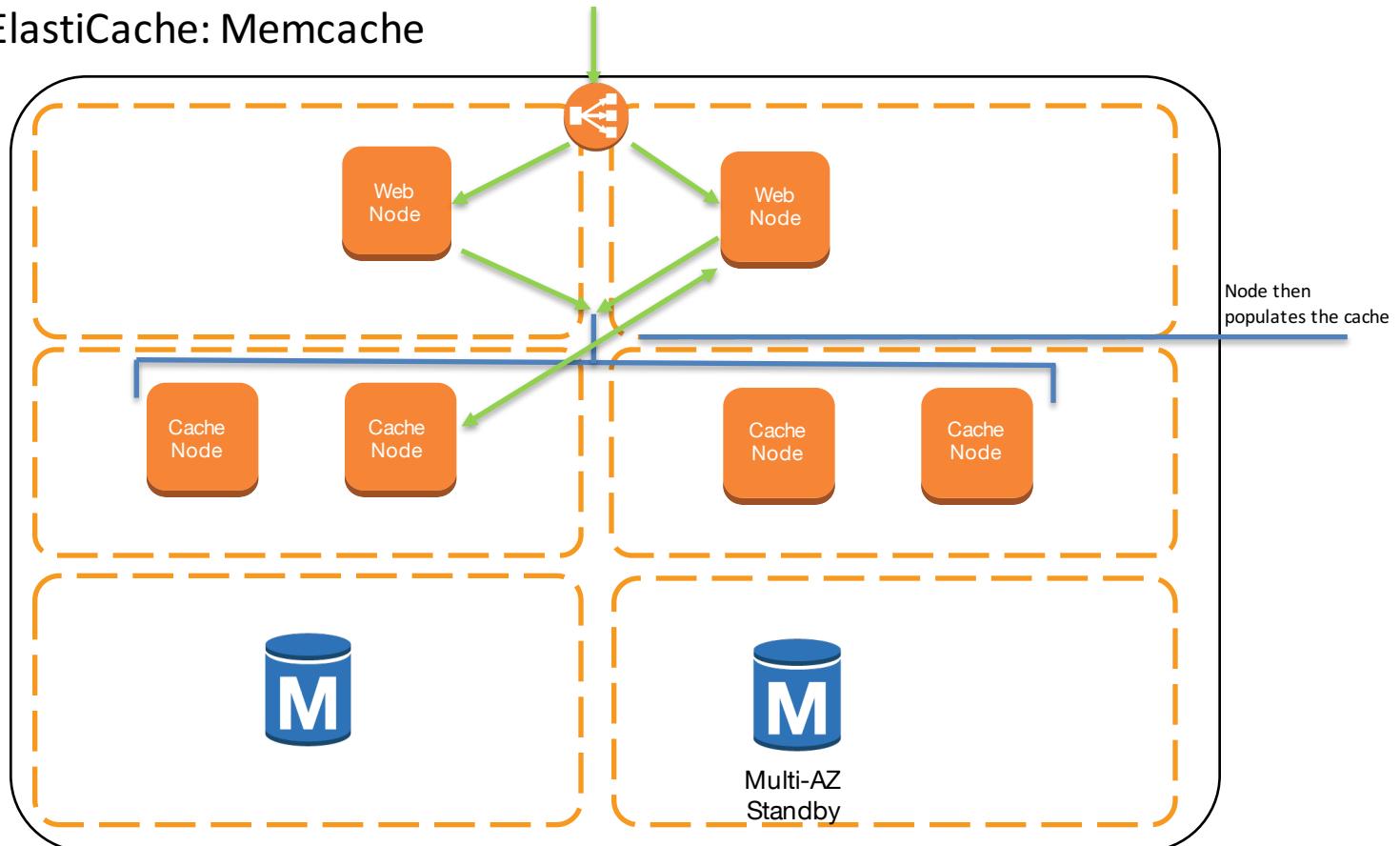
Amazon ElastiCache: Memcache



Amazon ElastiCache: Memcache



Amazon ElastiCache: Memcache





Amazon ElastiCache: Memcached

- If you need to scale the nodes in a cluster up or down to a different instance type, you must create a new cluster with the new node instance type
- Purchase reserved nodes to reduce costs -> not good for spot
- Can scale by adding on-demand nodes for times of increase in demand
- Every node in the cluster is the same instance type
- Memcached supports auto discovery, client programs automatically identify all nodes in a cache cluster
- Improve fault tolerance by locating nodes in multiple availability zones



Amazon ElastiCache: Memcached

- Memecached is a region only service there is no method for “migrating” ElastiCache clusters to another region other than firing up a new cluster and letting it populate in another region
- In a multi-region design, have an ElastiCache cluster in each region populating data from the local/regional DB server
- Memecached is a great solution for storing “session” state in applications this will make web servers stateless which allows for easily scaling



Amazon ElastiCache: Memcached Backups

Memcached uses a database as its persistent storage in the event of a node failure cache misses will make requests to the backend DB to populate the cache engine.

Note: This can cause an increase load on your SQL server to mitigate this load use more nodes in a cluster so a loss of a node does not equate to a substantial increase in database load on your backend database store.

When “events” occur to clusters notifications can be configured to be sent to SNS topics for automation and notification



Linux Academy

Amazon Web Services

Amazon ElastiCache: Redis



Amazon ElastiCache: Redis

Redis caching engine is substantially different than memcached. Redis provides persistent storage options instead of using a DB such as MySQL or MariaDB.

Redis uses:

- Small enough data sets that can be stored in-memory
- Need a persistent key store or caching engine that provides persistence
- Automatic failover to a backup node in case of node failure
- Backup and restore capabilities
- Leader boards
- Data with intense calculations and frequent changing data



Amazon ElastiCache: Redis persistence

Redis is often used as a replacement of some DB servers which in a memcached cluster are what allows for persistence. To apply persistence to a cluster in the event of a reboot, enable Redis Append Only Files (AOF)

- Disabled by default
- Will write all commands that change cache to an “append-only” file
- If a node is rebooted and the memory is cleared then when Redis caching engine starts the AOF is loaded through the commands in the AOF file and the cache is available again.



Amazon ElastiCache: Redis

Scaling Redis

- Scales similar to RDS scaling to increase capacity for writes you need to increase instance size
- Redis also supports clusters of read replica groups
- To increase the size of a Redis node
 - Take a snapshot of the node
 - Launch a new instance with instance type based off of the snapshot
 - Can also launch a new cluster and “seed” it from a snapshot



Amazon ElastiCache: Redis backups

Redis is the only caching engine “currently” that supports backups on ElastiCache.

Automatic Snapshots – Backups are taken on a daily basis, select a snapshot window and time limit, if failure occurs on a cluster then the cluster can be resorted from the most recent snapshot

Manual Snapshots – Can be taken at anytime and are not subject to the “retention limit” of automatic backups

Snapshots can be exported into an EC2 managed environment

Redis snapshots can be copied but cannot be copied to another region they can only be “copied”



Amazon ElastiCache: Use cases

- Leader boards
- Session state data
- Recommendation data
- Hootsuite session state example and why it's good for failover



Linux Academy

Amazon Web Services

Amazon Redshift



Amazon Redshift: Overview

Fully managed petabyte scale data warehouse used for storing large amounts of data for business intelligence applications.

Redshift runs in a single AZ IF the AZ supports Redshift clusters

Redshift nodes are continuously backed up to Amazon s3 and in the event of a failed drive in the cluster redshift will re-replicate the data from the failed drive and replaces the nodes as needed

Redshift nodes are all within the same availability zone and cluster is not available in multiple availability zones at one time



Amazon Redshift: Overview

Redshift distributes the query from the “leader” node in parallel across all the cluster’s compute nodes.

The compute nodes work together to execute the queries and return the data back to the leader node which then organizes the results and sends it back to the client requesting the data from the cluster.

Amazon Redshift: Scaling

Small single node clusters and scale up to larger multi-node clusters as demand changes

Change instance type of the cluster node

- Type of instance determines the total storage

Adding additional nodes

- Queries are sent in parallel to the replica nodes from the primary node from within a cluster so the data is distributed across all available nodes within a cluster
- To scale it is as simple as adding more nodes to cluster as long as the instance type of the cluster is still within operation requirements
- When adding nodes Redshift manages all the data distribution and load balancing of the data from within the cluster to the new nodes

Amazon Redshift: Changing the node type of a cluster

Considerations: Any change made to the cluster requires that enough resources be provisioned to managed the amount of current storage on the cluster or the process will fail

- Multi-node to single-node | single-node to multi-node
- Adding nodes
- Changing node type

Resizing a cluster:

- All connections are terminated and the cluster is restarted in read-only mode, any transaction that was not completed will be rolled back
- A new cluster is started (by Redshift) and uses the original (source) cluster as a data source to populate the new cluster
- The new cluster is in read-only mode until the resize is completed
- End point is updated and old cluster terminates all connections

Amazon Redshift: Costing

Storage is provisioned as part of the node as long as a cluster is running AWS will charge the nodes. Spot instances are not an option when working with Amazon Redshift due to the nature of the type of application

On-demand: on-demand instances can be added for scaling a node or temporary redshift clusters can also rely on on-demand

Reserved instances: To reduce costs for nodes that will maintain a continuous running state then purchase reserved instances to reduce the cost of the nodes

- Must be proper instance type
- Must be in the proper region/availability zone for the reserved pricing to apply

Amazon Redshift: Costing

BCJC is running a Redshift cluster for a petabyte scale data warehouse application. BCJC anticipates the cluster running 6 nodes of the ds1.xlarge instance type 24x7x365. Currently BCJC has purchased 9 reserved instances that match the proper availability zones and instance type.

- BCJC will be charged the discounted rate for the 6 running nodes
- BCJC will also pay the discounted rate for the additional 3 nodes reserved even though the cluster is only running 6 nodes



Amazon Redshift: Backups

Data on Amazon Redshift needs to be backed up with Redshift data snapshots

- Point-in-time snapshots are stored on Amazon S3 for durability (done by Redshift)
- Automatic and manual snapshots are available
- Redshift can restore data from a snapshot by launching a new cluster and importing the data from the snapshot

Amazon Redshift: Snapshot Region Copy

Snapshots can be copied from one region to another region (if the region supports Redshift)

- Manual Copy: Manually copy a snapshot from one region to another
- Automatic Copy: Redshift will automatically copy a snapshot from one region to another
retention period for the destination region can also be configured so automated snapshots can be removed after the retention period

Note: Snapshot copying does incur data transfer costs from one region to another

Amazon Redshift: Restoring From a Snapshot

A restore from a snapshot will contain the following information

- Number of nodes
- Type of nodes
- Cluster configuration
- Data included in the DB's of the cluster



Linux Academy

Amazon Web Services

CloudFront Key Concepts And Overview

CloudFront Key Concepts And Overview

Dynamic Content & Whole Site CDN – CloudFront is not just a “static files only” CDN anymore. When you enable “forward query strings” these will now be forwarded to the origin (if the origin supports it S3 does not) which allows the CDN to cache static pages such as word press posts that pull from a database. We’ll learn in whole site CDN how to configure this to ensure if the dynamic content changes it doesn’t stay cached.

Media Streaming – CloudFront allows you to stream media on-demand, Adobe RTMP streaming distributions as well as streaming origins such as WOWZA EC2 instances.

Invalidation – CloudFront will cache the last requested item until either the TTL on the item expires, the object is invalidated, OR the TTL is set to zero and the last modified header has not changed

Custom SSL – By default CloudFront provides a xxxx.cloudfront.net URL. With this comes an SSL certificate associated with the cloudfront.net domain. If there is a requirement to use a custom domain i.e linuxacademy.com you must provision and configure your own SSL certificate in IAM and associate it to your CloudFront distribution.

Custom Error Messages – CloudFront allows you to respond back with custom error message/pages. I.E 404 not found page.



HTTP Methods: Core benefits are allowing you to use CloudFront for all website actions

DELETE – no caching

GET - caching

HEAD - caching

OPTIONS - caching

PATCH – no caching

POST – no caching

PUT – no caching

What does this mean?

1. If you upload an option using PUT it is not cached on the origin even though the upload process uses the closest origin. The origin acts only as a proxy back to AWS which does in fact reduce latency and speeds up the upload process.
2. Delete request will delete the object but not remove it from cache, invalidating the cache is still required.



Linux Academy

Amazon Web Services

Dynamic Content With CloudFront

Dynamic Content With CloudFront

- Use one CDN for an entire website rather than one for just static files.
- Use custom origins and origin rules to determine what part of the website requests go to an origin. For example, images to go S3 but dynamic content goes to a specific EC2 instance.
- Whole site CDN works with uploads as well, up to 20GB. The edge location acts as a proxy for the uploaded object to the origin with the speed of an AWS backed network rather than open internet. This will increase site performance even with uploads!
- Use 0 TTL for dynamic content



Dynamic Content With CloudFront

Scenario: BCJC is consulting for a company that runs their current application entirely all on-premise. However, they are expecting a big boost in traffic tomorrow and need to figure out a way to decrease the load in order to handle the scale. Unfortunately, they cannot migrate their application to AWS in the time period required. What could they do to their current on-premise application to help offload some of the traffic and scale to meet the demand expected in 24 hours?

Whole site CDN! CloudFront allows you to specify custom origins including on-premise servers and sources. Configure static resources in the CDN as well as dynamic content and enable query string forwarding.

Dynamic Content With CloudFront

How does dynamic caching work? What if the dynamic content has changed?!

- Create a custom origin for your dynamic content
- Enable forwarding of query strings
- Set the TTL to 0 **IMPORTANT!** What does a TTL of 0 do?
 - It will cache the content even though the TTL is set to zero
 - When a request is made it will make a GET request to the origin with an **“If-Modified-Since”** header to determine if there is new data in the origin if there is then the new data is requested and cached else the current data is served from the origin



Dynamic Content With CloudFront

Device Detection: Send users different content based on the type of device that makes the request to the CloudFront origin which is based off of the User Agent header.

Geo Targeting: Serve content specific to an individual country by using CloudFront Geo targeting; URL stays the same content sent is different.

How it works: Essentially AWS now records this information and sends it as part of the request. Your code on the application server can process the data and return customized content based off of the information.

Query Strings / URL parameter forwarding

i.E <http://domain.com/videodownloads?current=5> (forwards current = 5 to the server)



Linux Academy

Amazon Web Services

CloudFront Reporting



CloudFront Reporting

Access Logs: Shows details of every request made to your CloudFront origin. Can integrate with EMR for log analysis.

Log data:

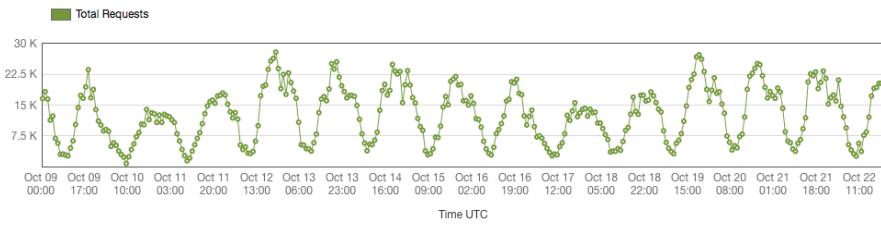
- Object requested
- Date and time of request
- Edge location serving the request
- Client IP address
- HTTP Referrer
- HTTP User Agent

Access logs are sent to and stored in Amazon S3 buckets

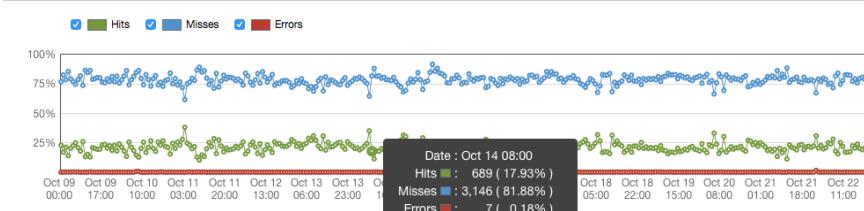


CloudFront Reporting: Cache Statistics

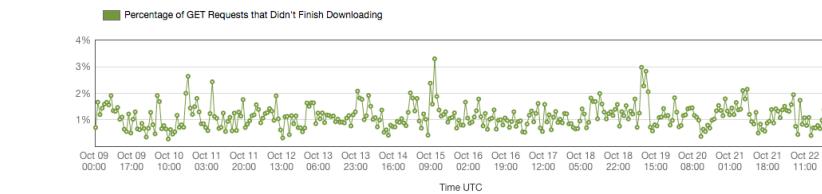
Total Requests (Millions | Thousands | Not Scaled) [Show Details](#)



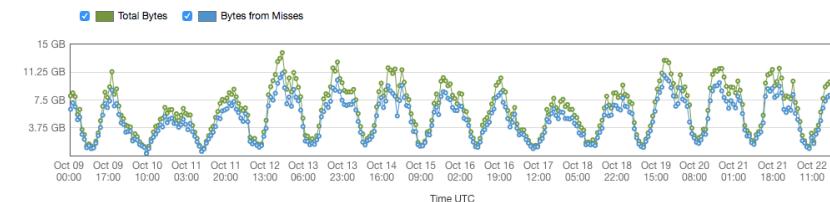
Percentage of Viewer Requests by Result Type [Show Details](#)



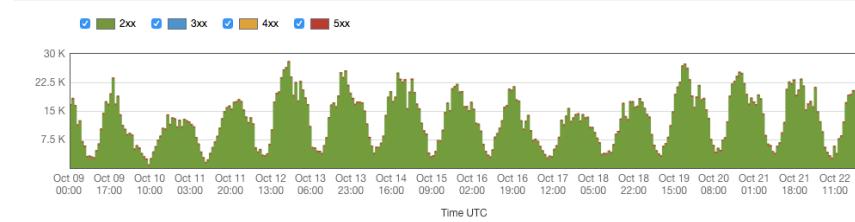
Percentage of GET Requests that Didn't Finish Downloading [Show Details](#)



Bytes Transferred to Viewers (Gigabytes | Megabytes | Kilobytes) [Show Details](#)



HTTP Status Codes (Millions | Thousands | Not Scaled) [Show Details](#)





CloudFront Reporting: Additional Reports & Analytics

Popular Objects: Shows the most requested objects from the CDN distribution

Top Referrers: Shows the URL that made the most requests to the CDN distribution

Usage: Number of HTTP / HTTPS requests, Data transferred By Protocol, Data Transferred By Destination (From CloudFront To The Users / From CloudFront To The Origin)

Viewers:

- Devices
- Browsers
- Operating Systems
- Locations



Linux Academy

Amazon Web Services

CloudFront Security

CloudFront Security

Private Content

- Signed URLs: Provide URLs with expire dates to limit access to content
- Signed Cookies: Signed cookies are new and are an extremely flexible tool in terms of limiting content. You can limit content without limiting access to the URL. For example: if a user is logged into a site, you can issue a signed cookie that verifies they have permission to access certain parts of the site. If streaming HLS files from CloudFront you can also create signed cookies that will be validated each time an HTTP request is made to an HLS chunk. Essentially, providing secure streaming!



CloudFront Security

Geo Restriction: A CloudFront setting that allows you to specify which countries your CDN will deliver to

Edit Geo-Restrictions

Geo-Restriction Settings

Enable Geo-Restriction Yes No



Restriction Type Whitelist Blacklist



Countries



AF -- AFGHANISTAN
AX -- ALAND ISLANDS
AL -- ALBANIA
DZ -- ALGERIA
AS -- AMERICAN SAMOA
AD -- ANDORRA

Add >>

<< Remove



Linux Academy

Amazon Web Services

CloudFront Security: Forcing HTTPS To The Origin

CloudFront Security: Forcing HTTPS To The Origin

If the origin is S3 then requests made to the CloudFront distribution will forward as the protocol that was originally made. I.E if the client request was HTTPS it will forward to the origin as HTTPS

Custom Origins: Custom origins have the option to forward as HTTP only or “Match Viewer” which means if the client request is HTTPS then the request is made from CloudFront to the custom origin as HTTPS.

Origin Settings

Origin Domain Name	<input type="text"/>	
Origin Path	<input type="text"/>	
Origin ID	ELB-wowza-b-1100773316	
Origin Protocol Policy	<input checked="" type="radio"/> HTTP Only <input type="radio"/> Match Viewer	
HTTP Port	80	
HTTPS Port	443	



Linux Academy

Amazon Web Services

CloudFront Performance



CloudFront Performance Considerations

- Increase performance by increasing the number of requests that are cache hits instead of cache misses
- Use CloudFront to upload objects, the edge location will proxy the data back to the origin location going over the AWS backend network
- Increase minimum TTL and maximum TTL so items are cached longer (if they are not frequently changing)

How does Cloud Front React in the event of high load and multiple simultaneous requests?

In case of increase in simultaneous requests CloudFront, will wait for the first request to finish before processing the second request.



Linux Academy

Amazon Web Services

CloudFront Video Streaming



CloudFront Video Streaming

Video streaming on CloudFront is a very useful tool as you can use the CDN to stream video around the world. The key is understanding how to stream different types of video and how to secure access to the video if required.

On-Demand Streaming

Pre-Recorded Media Streaming

Live Streaming



CloudFront Video Streaming

Video streaming on CloudFront is a very useful tool as you can use the CDN to stream video around the world. The key is understanding how to stream different types of video and how to secure access to the video if required.

On-Demand Streaming: On-Demand streaming is configured on web CloudFront distributions.

Smooth: To enable Microsoft smooth streaming, create a web distribution and on the custom origin select “Enable smooth streaming”

Progressive Downloads: Progressive download is the process of transferring digital media files (HLS/MP4) from a CloudFront origin to a client over HTTP/HTTPS



CloudFront Video Streaming

Streaming of pre-recorded media, usually MP4 files over the Adobe Streaming RTMP protocol. This is actual video streaming and not video download and requires a video streaming distribution when creating a new CloudFront distribution.

Live Streaming: Use CloudFront CDN with a streaming server origin such as WOWZA media server to stream live events. Live event streams will send chunks of data that can be cached in a “delay” by the CDN so live requests are being served via CloudFront and limited streams are being sent to the streaming origin such as WOWZA EC2 instances. To configure this setup you would use a web CDN and NOT an RTMP CDN.

Note: Keep in mind there is no “streaming switch” other than enabling smooth streaming on CloudFront distributions. This means understanding what type of media should be streaming from what type of CloudFront distribution is important.



Linux Academy

Amazon Web Services

Amazon Elastic Transcoder

Amazon Elastic Transcoder

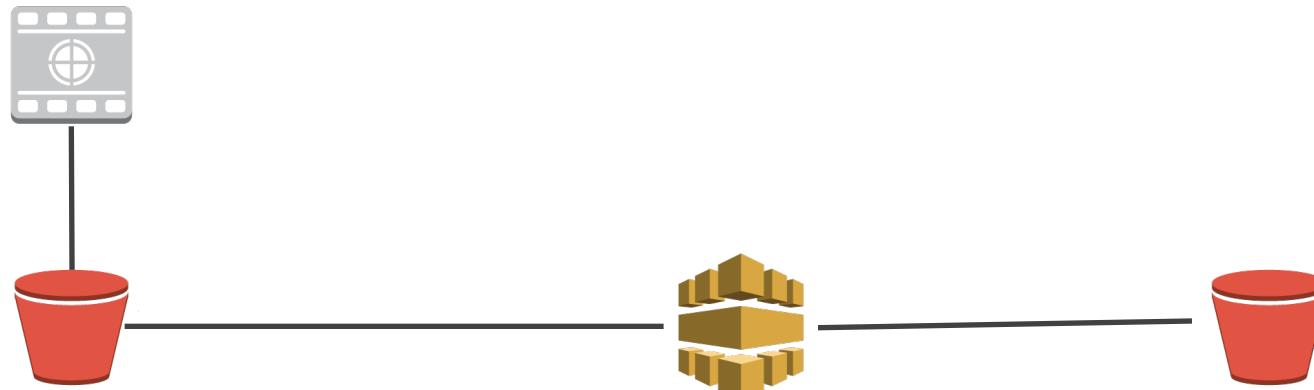
Elastic Transcoder is a fully managed AWS application service that works out of specific regions

Elastic Transcoder is used to convert media files stored on AWS S3

- Different formats (mobile available, i.e HLS)
- Different Quality levels
- Different Resolutions
- Apply Captions
- Create MP3 files from video files
- Add watermarks to videos

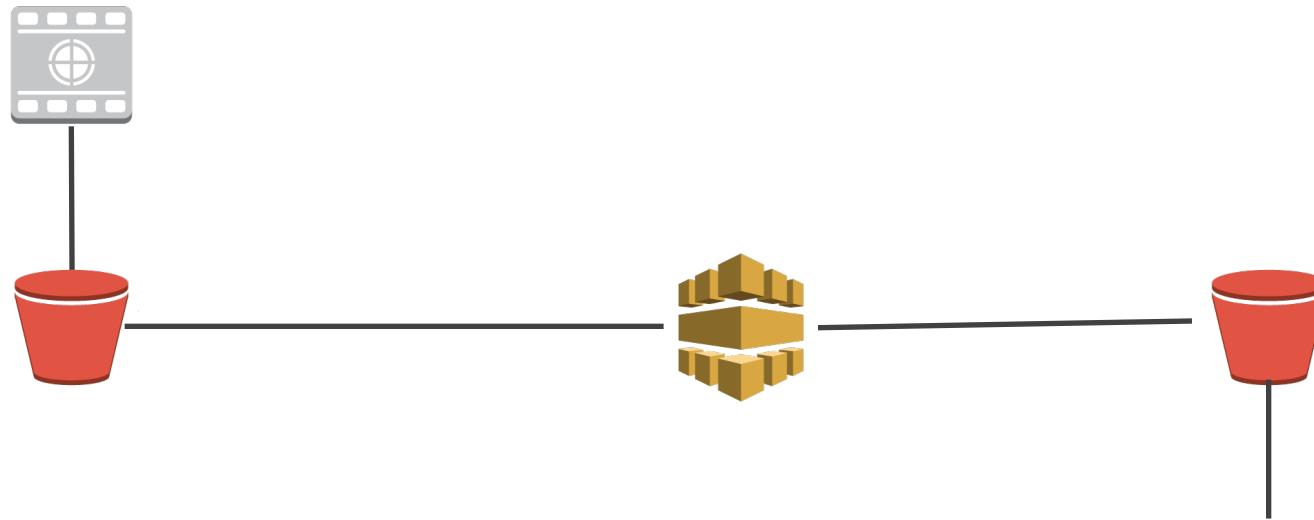


Amazon Elastic Transcoder





Amazon Elastic Transcoder



The result S3 bucket that receives the transcoded file is a prime example of when you can use RRS. The transcoded files are easily reproduced from the “source” video bucket.

Amazon Elastic Transcoder: Components of Elastic Transcoder

Jobs: A job is called via the API when you specify the type of encoding, video settings, and pre-sets for videos you want to create. A single job can create up to 30 output video types.

Pipelines: Pipelines are where the jobs are submitted. Pipelines handle each job in the order in which they are submitted to the pipeline. The pipeline is where the destination and source buckets are configured for the outputted files. All jobs in a pipeline can be temporarily stopped just by “pausing” the entire pipeline.

Presets: Pre-built templates for transcoding files into one format or another.

Notifications: Integrate into SNS for automation and job updates

**Create New Pipeline**

A pipeline is a queue for your transcoding jobs. You can have more than one pipeline per AWS account.

Pipeline Name ⓘ

Input Bucket ⓘ

IAM Role ⓘ

Elastic Transcoder previously created a default IAM role for this AWS account. [View the policy.](#)

[Update default role](#)

Configuration for Amazon S3 Bucket for Transcoded Files and Playlists

Bucket ⓘ

Storage Class ⓘ

[+ Add Permission](#)

Configuration for Amazon S3 Bucket for Thumbnails

Bucket ⓘ

Storage Class ⓘ

[+ Add Permission](#)



A job contains all of the information that Elastic Transcoder needs to transcode one media file into another format. When you create a job, it's automatically added to the pipeline that you specify.

Pipeline ⓘ

Input Key ⓘ The name of the file that you want to transcode. If the filename includes a prefix, for example, cooking/lasagna.mpg, include the prefix in the key.

Output Key Prefix ⓘ The value, if any, that you want Elastic Transcoder to prepend to the names of all files that this job creates, including output files, thumbnails, captions, and playlists. If you specify a value, it must contain a / somewhere after the first character to simplify Amazon S3 file management.

Decryption Parameters None Enter Information ⓘ Decrypt the Input:
If the input file is encrypted, enable this box and specify the settings that Elastic Transcoder needs to decrypt them.

- AES Cipher-Block-Chaining with PKCS7 padding
- AES Counter Mode
- AES Galois Counter Mode

Description Key: The Base64 encoded encrypted key that you want Elastic Transcoder to use to decrypt the input.
Description Key MD5: The Base64 encoded MD5 digest of the encrypted key that you want Elastic Transcoder to use for a key checksum.
Description Initialization Vector: The Base64 encoded initialization vector that you want Elastic Transcoder to use to decrypt the input.

Output Details (1 of 1)

Preset ⓘ The preset that you want to use for this job. The preset determines the audio, video, and thumbnail settings that Elastic Transcoder uses for transcoding.

Output Key ⓘ The name of your output file. When not used as a pattern, such as for segmenting, you should include an appropriate extension such as .mp4, .ts, .webm, .ismv, .mp3, .ogg, .oga, .flac, .mpg, .gif, .f4v, or .mxf. You may also include a / in your example "outputs/movie.mp4".

Encryption Parameters None Enter Information ⓘ Encrypt the Output:
If you want the output files encrypted, enable this box and specify the settings that Elastic Transcoder needs to encrypt them.

- Server-side encryption with Amazon S3 managed keys
- Server-side encryption with AWS KMS managed keys
 - If you don't specify an AWS KMS key, Amazon S3 will use the default service key
- AES Cipher-Block-Chaining with PKCS7 padding
- AES Counter Mode
- AES Galois Counter Mode

Encryption Key: The Base64 encoded encrypted key that you want Elastic Transcoder to use to encrypt the output.
Encryption Key MD5: The Base64 encoded MD5 digest of the encrypted key that you want Elastic Transcoder to use for a key checksum.
Encryption Initialization Vector: The Base64 encoded initialization vector that you want Elastic Transcoder to use to encrypt the output.

Available Settings Clip Captions ⓘ

+ Add Another Output

**▼ Summary**

ARN arn:aws:elastictranscoder:us-east-1:765783612490:preset/1440617823137-rls7cz
Name Optimized 720p
Preset Id 1440617823137-rls7cz
Description
Container mp4

▼ Video

Codec H.264
Codec Options InterlacedMode:Progressive,MaxReferenceFrames:4,Level:4,ColorSpaceConversionMode:None,Profile:baseline
Maximum Number of Frames Between Keyframes 300
Fixed Number of Frames Between Keyframes true
Bit Rate auto
Frame Rate 30
Video Max Frame Rate
Max Width 1280
Max Height 720
Sizing Policy ShrinkToFit
Padding Policy NoPad
Display Aspect Ratio auto

Watermarks

Id	Max Width	MaxHeight	Sizing Policy	Horizontal Align	Horizontal Offset	Vertical Ali
-----------	------------------	------------------	----------------------	-------------------------	--------------------------	---------------------

▼ Audio



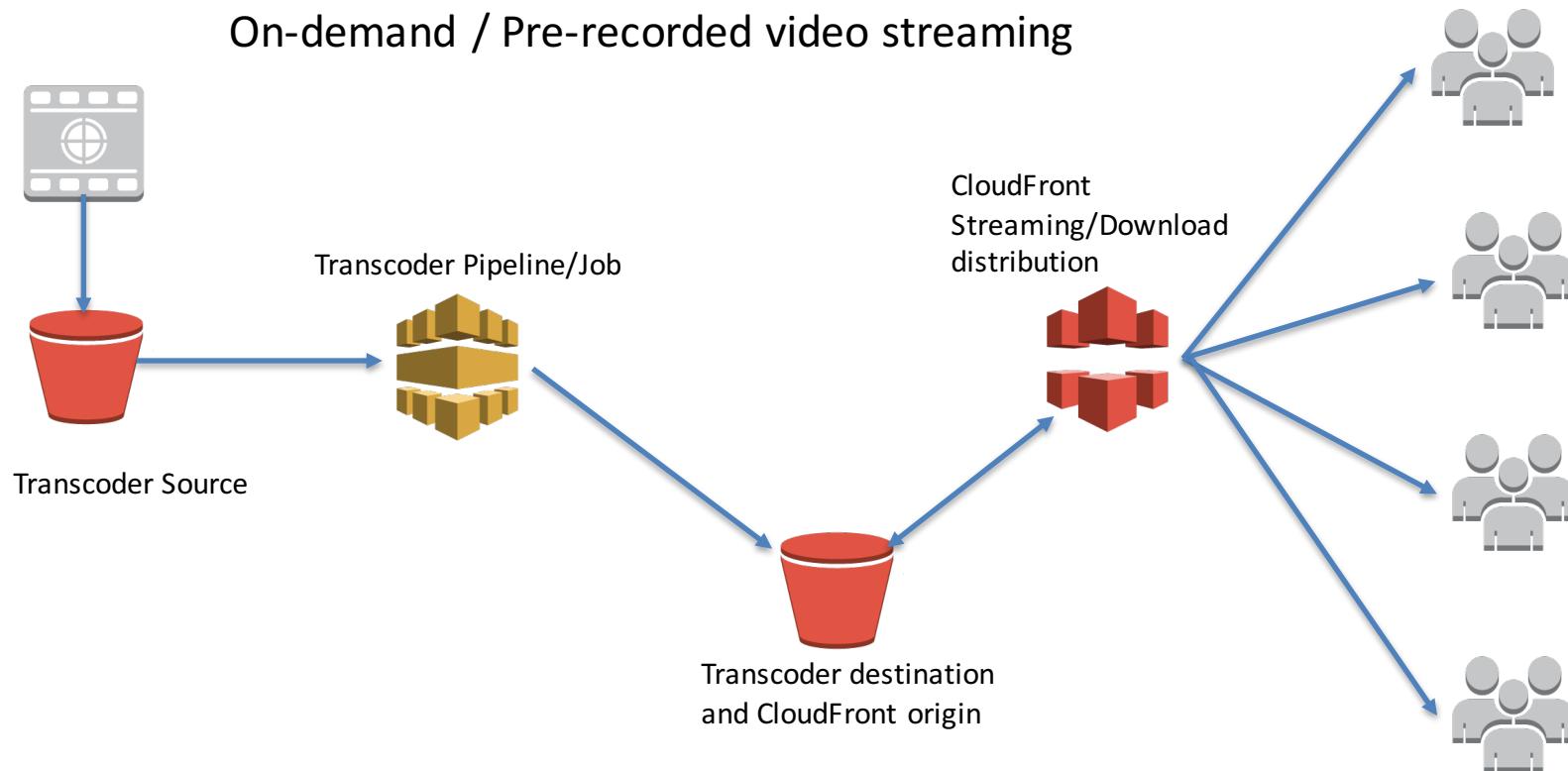
Linux Academy

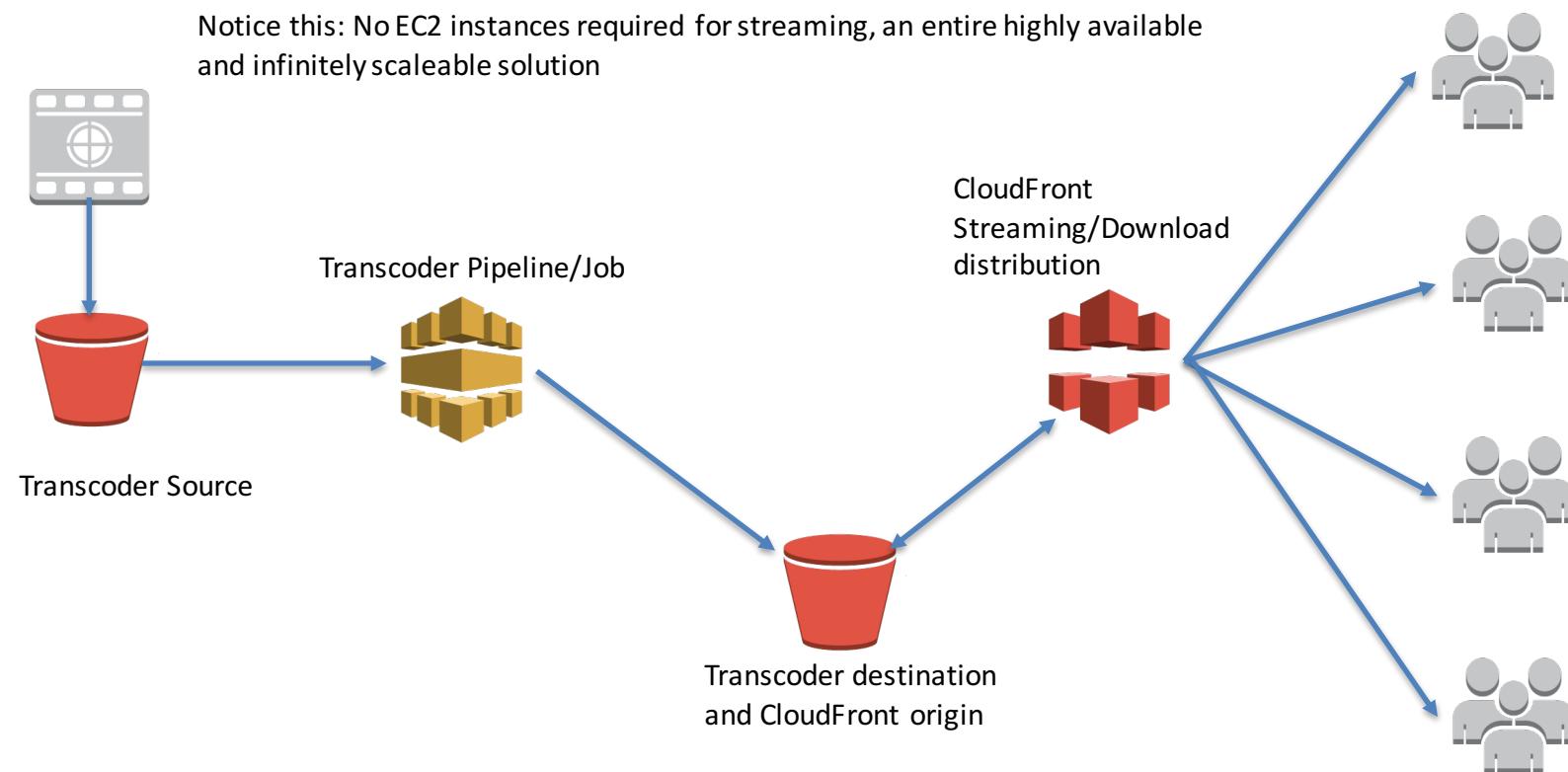
Amazon Web Services

Streaming With S3, CloudFront, And Transcoder

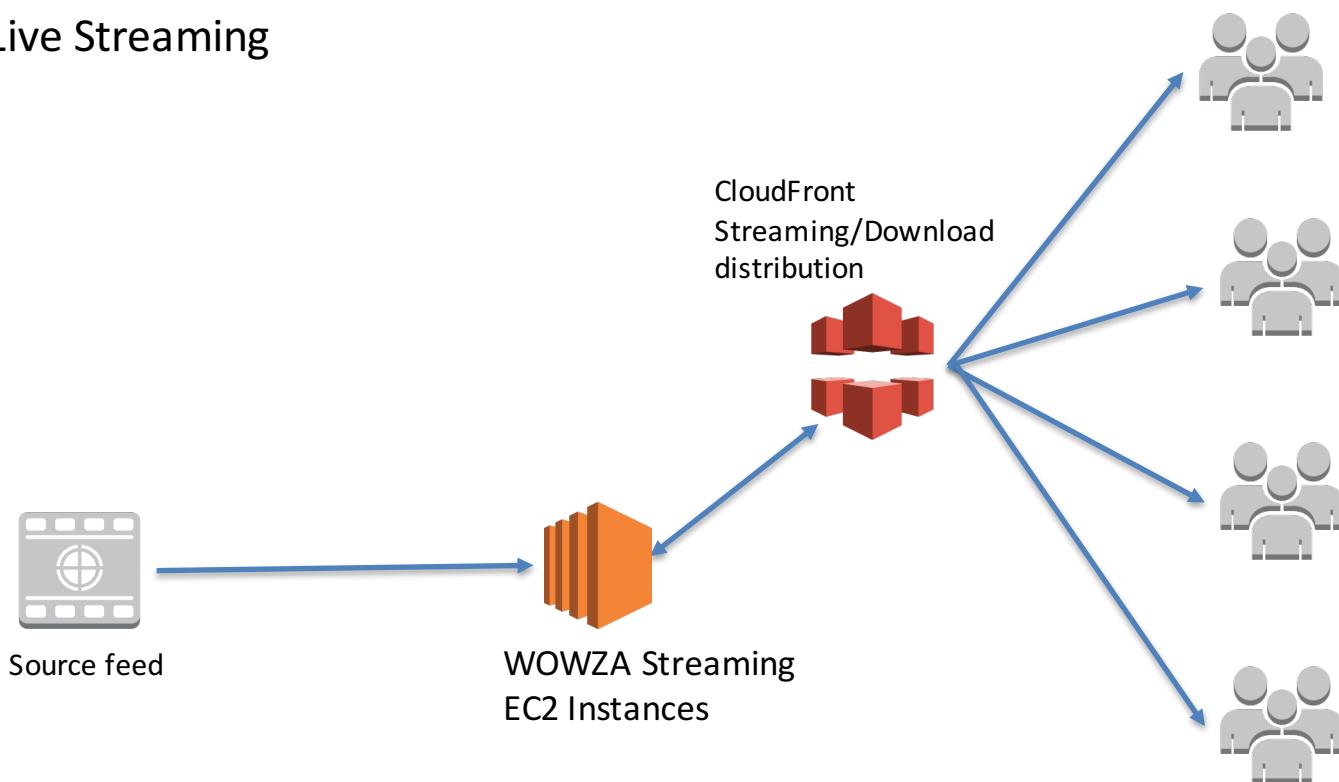


On-demand / Pre-recorded video streaming





Live Streaming





Securing Your On-Demand Videos

Encryption at rest

- Use AWS KMS to decrypt source data and encrypt resulting output
- Use origin access identity on your CloudFront distribution so content is only able to be served via CloudFront, NOT S3 URLs



Securing Your On-Demand Videos

What about streaming protection?

Signed URLs:

- Streaming RTMP data from a streaming distribution (Signed cookies are not supported)
- Signed URLs for progressive download, security hole because it makes the file available for download for as long as the video is available
- If a client does not support cookies

Signed Cookies

- Providing access to multiple files, for example, chunk files of HLS, the signed cookies will be “checked” for each served chunk
- Does not require “custom signed” URLs the URL link can stay the same



Linux Academy

Amazon Web Services

AWS Data Pipeline



AWS Data Pipeline

AWS Data Pipeline is used for automating the transfer and/or transformation of data

Examples:

- Migrating DynamoDB tables to another region (can also do this with DynamoDB streams) also known as importing/exporting DynamoDB data
- Exporting RDS tables
- Taking data from an S3 bucket, for forming ETL (extract transform and load), and uploading to another resource such as RDS, DynamoDB, Elastic Map Reduce, etc.
- Processing data using EMR with Hadoop streaming



Data pipeline key benefits:

- Automate the movement of data between services
- Move data to and from services and transform data
- Set preconditions to “tasks” before the pipeline starts
 - I.E Wait until log files are delivered to S3 before starting the pipeline that sends them to EMR for processing
- Can run a pipeline:
 - Once
 - Defined number of times
 - Run on activation
 - Run indefinitely
 - Run repeatedly within a date range



Data pipeline use case examples

Export and Import DynamodDB tables for backup/restore across regions



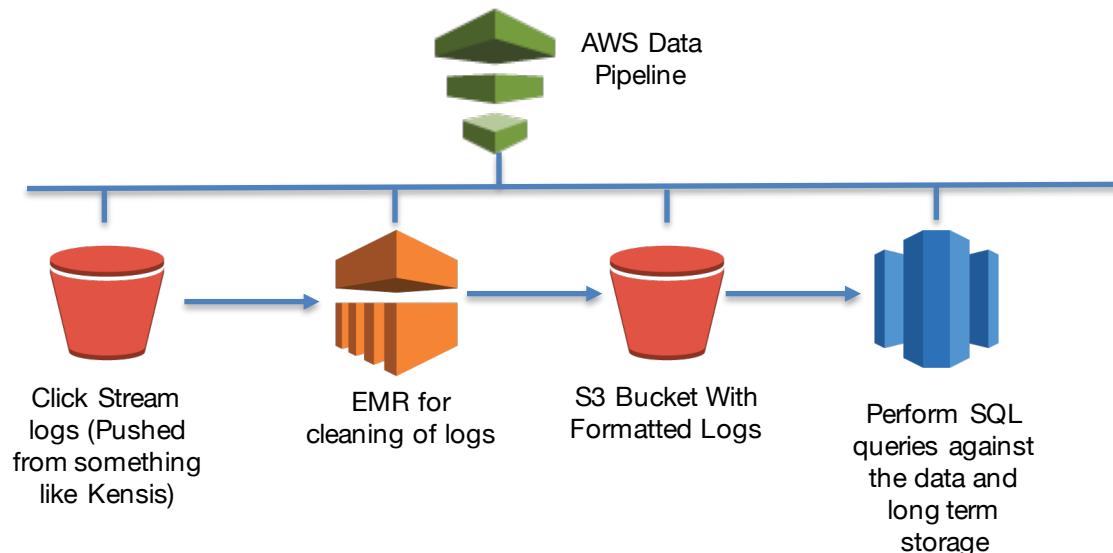
Data pipeline use case examples

Importing data to Redshift:

- Bulk copy data from DynamoDB or S3 to a new or existing Redshift table.
- To move data from RDS to Redshift first move it to S3 as part of the pipeline
- Run SQL queries on the data that is stored within Redshift data and those query results can be stored in a new table or modified in the existing table



Data pipeline use case examples: Clickstream analysis (diagram)





Technical Overview:

Task Runners: An application “polling” the pipeline for tasks to perform and performs that task

- Can be launched by pre-built data pipeline templates
- Can be added to EC2 instances or an on-premise server!

Data Nodes: The location and type of data the pipeline uses as input and output

- DynamoDBDataNode
- MySqlDataNode
- RedshiftDataNode
- S3DataNode



Technical Overview:

Activities: This defines what is suppose to be done by the pipeline. Pre-built activities are available in data pipeline but you can also write custom scripts to perform custom tasks

- Copy data from one location to another
- Run an EMR Cluster
- Run a Hive query within an EMR cluster
- Run a Pig script on EMR cluster
- Copy data to and from Redshift tables
- Run custom shell commands
- Run a SQL query on a supported database



Technical Overview:

Databases: Supported databases

- JDBC database
- RDS Database
- Redshift database



Technical Overview:

Preconditions: An assertion that must be true in order for the pipeline to run.

You can create a custom pre-condition with a script or use a data pipeline precondition

- DynamodDBDataExists: Checks for data within a specific DynamoDB table
- DynamoDBTable Exists: Checks to see if a DynamoDB table exists
- S3KeyExists: See if an S3 key exists (object)
- S3PrefixNotEmpty

User Preconditions

Exists: Checks to see if a data node exists

ShellComandPrecondition: Executes a Linux bash command



Technical Overview:

Resources: Computational resources which performs the specified pipeline activity.

- Ec2Resource
- EmrCluster



Data pipeline Cost Considerations:

Data pipeline utilizes EC2 instances for both EC2 based resources and EMR based resources

- Purchase reserved instances to reduce the cost of EC2 based on usage
- Use spot instances for the “task” EMR nodes
 - The data is persistent because the core nodes in an EMR cluster are on-demand so risk of them being terminated due to being outbid is eliminated

Trade offs of reduced costs with spot instances:

- If you’re using spot nodes, then it can take longer for the pipeline to start due to waiting on successful bidding
- A pipeline can fail and have to be retried if spot instances are outbid at anytime



Linux Academy

Amazon Web Services

RDS Overview + Security



AWS RDS

AWS Professional Requirements:

- Understand security design
- Understand multi-region RDS environments and architectures
- Understand hybrid on-premise to AWS architectures
- Understand how to scale RDS instances

This lesson focuses on security for RDS



AWS RDS: Security

Encryption (data at rest): Can be enabled on RDS instances to encrypt the underlying storage. By default, this will also encrypt snapshots as they are created and no additional configuration needs to be made on the client side for this to work.

If encryption is enabled:

- Keys are managed by AWS KMS
- Logs are encrypted
- Snapshots are encrypted
- Backups are encrypted
- Read replicas are encrypted
- Once created, the key used cannot be changed
- If the key is lost, then the DB can only be restored from a backup
- Encryption can only be specified at instance creation time
- Cross region replicas and snapshot copy does not work since the key is only available in a single region



AWS RDS: Security

RDS Databases that support at rest encryption on RDS:

- MySQL
- Oracle
- SQL Server
- PostgreSQL
- MariaDB



AWS RDS: Security

Transparent Data Encryption (TDE): Automatically encrypts the data before it is written to the underlying storage device and decrypts when it is read from the storage device.

This is a native feature of:

- Oracle: Requires key storage outside of KMS and integrates with CloudHSM for this
- SQL Server: Requires a key but is managed by RDS after enabling TDE



AWS RDS: Security

Encrypted Connections (SSL): SSL end points can be used (and should be used) to connect from the SQL client (the app making the SQL connection) to the RDS instance.

An SSL certificate is created when the RDS instance is created.



Linux Academy

Amazon Web Services

MySQL / MariaDB on RDS



AWS RDS: MySQL + MariaDB

AWS Professional Requirements:

- Understand multi-region RDS environments and architectures
- Understand hybrid on-premise to AWS architectures
- Understand how to scale RDS instances

MariaDB is a fork of MySQL

AWS RDS: MySQL + MariaDB

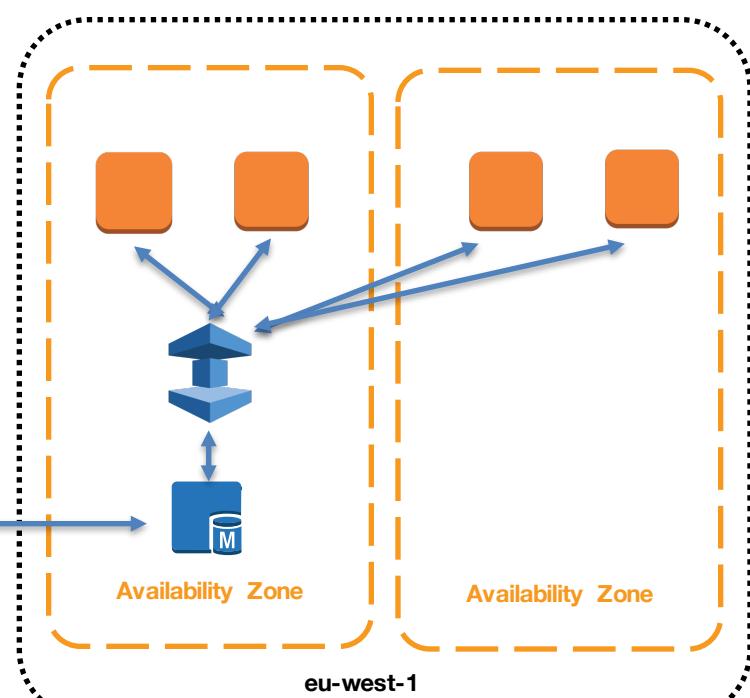
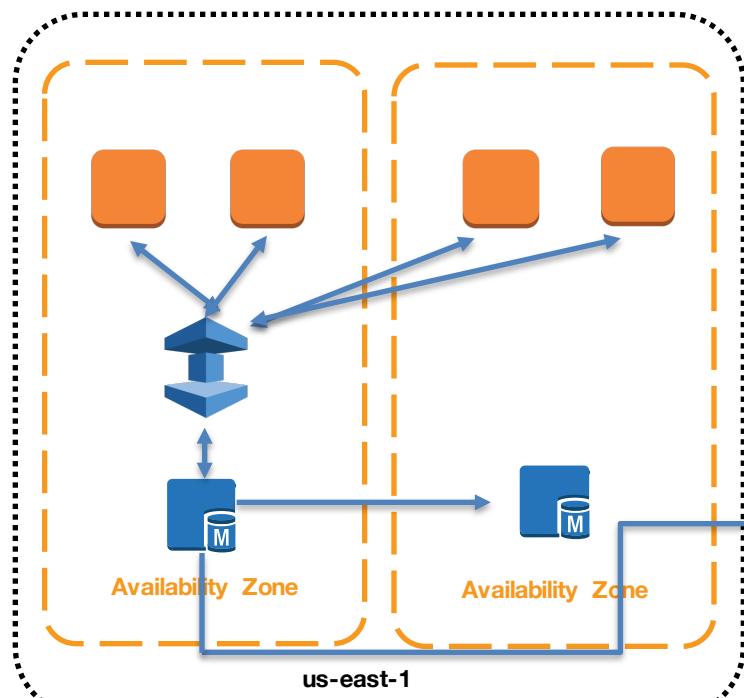
Read replicas: Read replicas can be created in any region supporting RDS. Data can be asynchronously replicated using the MySQL native replication features from the master instance to any read replica (slave) instance within any region.

- Improves disaster recovery (reduces the RTO and RPO of an application)
- Helps with data migration from one region to another
- Allows RDS “reads” to scale out globally (writes still need to happen on the master)
- Reduces load against the “master” database by sending read traffic to read replicas
- Still best practice to use caching in front of the read replicas depending on your update requirements



AWS RDS: MySQL + MariaDB

Database queries





Replication as a disaster recover or data migration mechanism

Replication with MySQL can be used to export data to an on-premise network

- Configure the RDS MySQL instance
- Configure the MySQL DB instance on RDS to be the replication source
- Use mysqldump and transfer the database from RDS to the on-premise MySQL
- Start replication to the instance running external to RDS (it is set as the slave)
- After the export is completed stop replication



Replication as a disaster recover or data migration mechanism

Replication for MySQL can also be configured from on-premise to RDS

- Set the source MySQL instance to read-only
- Determine the binlog location
- Use mysqldump to copy existing database to RDS
- Make the source writeable again
- Configure the security group to allow for your external IP address to communicate with the instance
- Create the MySQL replication user and grant permissions
- Configure the RDS instance to be a replica by using the `mysql.rds_set_external_master` command at the command line of the RDS instance
- Issue the `mysql.rds_start_replication` command on the replication RDS instance



Replication as a disaster recover or data migration mechanism

- On-premise to RDS backup (using AWS as a failover)
- RDS MySQL to another region with read replicas
- Multi-AZ failover for synchronous replication
- MySQL replication for importing data to the cloud (also use mysqldump/mysqlimport)

Note: Replication for MySQL server only works on MySQL 5.6.13 or later



Linux Academy

Amazon Web Services

RDS: Oracle DB



AWS RDS: Oracle DB

AWS Professional Requirements:

- Understand multi-region RDS environments and architectures
- Understand hybrid on-premise to AWS architectures
- Understand how to scale RDS instances

AWS RDS: Oracle DB

Available Oracle DB's on RDS:

- Oracle EE: Oracle Enterprise Edition
- Oracle SE: Oracle Standard Edition
- Oracle SE One: Oracle Standard Edition One

Databases not supported:

- Oracle RAC: A cluster database with shared cache architecture

Oracle RAC: This can run on EC2 instances even though multicast is required; you can use VPN (Ntop N2n) to create a tunnel between the nodes. Placement groups would be required since it is a cluster service. Data guard service can be used to extend high availability to the RAC design.

AWS RDS: Oracle DB

Importing databases into RDS Oracle instances:

- Import small databases with Oracle SQL Developer
- Import large databases with Oracle Data Pump
 - Import data from an Oracle EC2 to RDS Oracle DB instance
 - Import from database on Oracle DB instance to another Oracle DB instance
 - Import data from a local on-premises DB to an RDS Oracle DB instance

Oracle does not support:

- Cross region replication on RDS



AWS RDS: Backing Up Oracle DB Using RMAN and AWS

RMAN (Recovery Manager) is a backup and recovery manager included in all Oracle versions from Oracle 8 on.

- With on-premise Oracle servers, use RMAN to backup data to Amazon S3 as part of a hybrid environment
- With RDS based Oracle servers, use DBS snapshots for point-in-time snapshots
- RMAN can also be used with Oracle EC2 instances



Linux Academy

Amazon Web Services

RDS: MSSQL



AWS RDS: MSSQL

AWS Professional Requirements:

- Understand multi-region RDS environments and architectures
- Understand hybrid on-premise to AWS architectures
- Understand how to scale RDS instances



AWS RDS: MSSQL

- Supports point-in-time automatic backups and manual snapshot backups
- Supports Multi-AZ deployment options which uses the SQL Server Mirroring native to SQL server for high-availability and failover
- Read replicas are NOT supported on SQL server -> scaling will require increasing instance size
- Multi-region disaster recovery and backups will require using import/export tools provided by SQL server
- In order to have multi-region disaster recovery you can copy a snapshot from RDS to another region



AWS RDS: Importing MSSQL data

NOT as easy as the other open source technologies since MSSQL server is not supported on RDS.

1. Turn off all applications talking to the database
2. Disable key constraints
3. Disable backups
4. Create an empty table for each table being imported from on-premise to RDS
5. Export your tables and databases into “flat files” it’s a built in function as part of MSSQL server studio (on-premise)
6. Import those flat files into the new MSSQL RDS instance

Note supported MSSQL functions:

- Restore data from file
- FILESTREAM



Linux Academy

Amazon Web Services

AWS CloudSearch



AWS CloudSearch

AWS CloudSearch is a fully hosted solution provided by AWS. CloudSearch is used for indexing documents and information contained within the documents for search within an application.

CloudSearch provides search features similar to Apache SOLR and CloudSearch is powered by SOLR.

Search features include:

- Full text search
- Boolean search
- Prefix search
- Range Search
- Term boosting (assign higher importance to specific key words)
- Faceting (essentially “drill down” and “filter” searches)
- Highlighting (highlights all items found on a page based off of the search)
- Autocomplete Suggestions



AWS CloudSearch

Document types that can be indexed by CloudSearch

- CSV
- PDF
- HTML
- Excel
- PowerPoint
- Word
- Regular Text

AWS CloudSearch

CloudSearch can be used to search DynamoDB tables

- When updates to DynamoDB data occurs send the updates to CloudSearch
- Periodically send the updates to CloudSearch

Note: The CloudSearch data is indexed within cloud search. If changes occur to indexed items they will need to be re-uploaded to CloudSearch for indexing.

AWS CloudSearch

Scaling:

- CloudSearch will automatically scale based off of the increase in data and search load on the nodes
- You can manually scale out to additional nodes in the interface if there is an anticipation of increase in search traffic
- Multi-az is available by the “click of a button” to automatically add high availability
- The core of CloudSearch is just running software on EC2 instances so there are costs associated with those nodes which is the cost of CloudSearch



Linux Academy

Amazon Web Services

Amazon Elastic Map Reduce



Elastic MapReduce

EMR by AWS is a hosted version of the Apache Hadoop Clustering software

EMR is a fully managed and highly scalable service by AWS

EMR is used frequently for batch processing type applications and can easily integrate into Data Pipeline or is used for processing incoming data

Common use cases for EMR include

- Social and Mobile Data analysis
- Log analysis
- Clickstream analysis
- Sentiment analysis



Hadoop Components

HDFS (Hadoop Distributed File System)

- Distributed file system that aggregates the file storage of cluster data across nodes within the cluster
- Multiple copies of the data are stored across the nodes to ensure durability of the node data
- Data can be easily accessed in parallel since it is stored across multiple nodes thus increasing performance

MapReduce: Is the programming model used for processing large data sets written in Java and core map



Hadoop Data Services

Hive: A data processing component that uses an SQL “type” query language to analyze and process data. It creates a simple mechanism for processing extremely large data sets .

Pig: Is used for writing actual MapReduce programs (core programs are written in java and provide the fastest speed but are complex and large to write).



EMR is used for processing large data sets using a simple programming language to find “trends” in given data.

EMR works as a cluster that is made up of several components

- Master node: The “master” node manages data distribution to the additional core/slave nodes in the cluster to perform the labor intensive
- Core node: Stores data on HDFS from tasks run on the nodes and are managed by the master notes
- Task node: Task nodes are managed by the master node, no data is stored and HDFS is not on the task nodes. Task nodes perform data tasks and send the result data back to core nodes for HDFS storage/output
- EMRFS: Can be used instead of HDFS for storage/output on Amazon S3



EMR Spot Instances: Spot instances can be used as part of an EMR cluster to reduce costs. However, how and when you choose to use spot instances is determined based off of the cluster's use case and needs.

1. Is the cluster temporary or will it be long running?
2. Is it data critical or are the results urgent?
3. Is cost more important?
4. Is it a combination of one or more above?

Depending on the architecture requirements a combination of spot and on-demand instances between master, core, and task nodes can be used for EMR clusters.

- Is the application a test application?
- Does the data for processing exist on a persistent data store?
- Can the application handle termination in the middle of MapReduce jobs?



Linux Academy

Amazon Web Services

Deployment Concepts On AWS



Linux Academy

Amazon Web Services

OpsWorks Deployment And Concepts



AWS OpsWorks – Application stack configuration management using (Chef) and pre-built recipes by AWS or your own custom recipes.

- Pre-built stacks and layers using AWS resources including AWS deploy Chef recipes
- Ability to bring your own custom recipes from a GIT repository for customization of each layer
- Useful for grouping application components together in different components called “layers”. Layers are used to determine which recipes are deployed on the associated instances. An instance can belong to multiple layers.
- Simplify management of large scale multi-tier applications and facilitate continuous integration DevOps principles



Lifecycle events cause specific recipes to be run on all associated instances

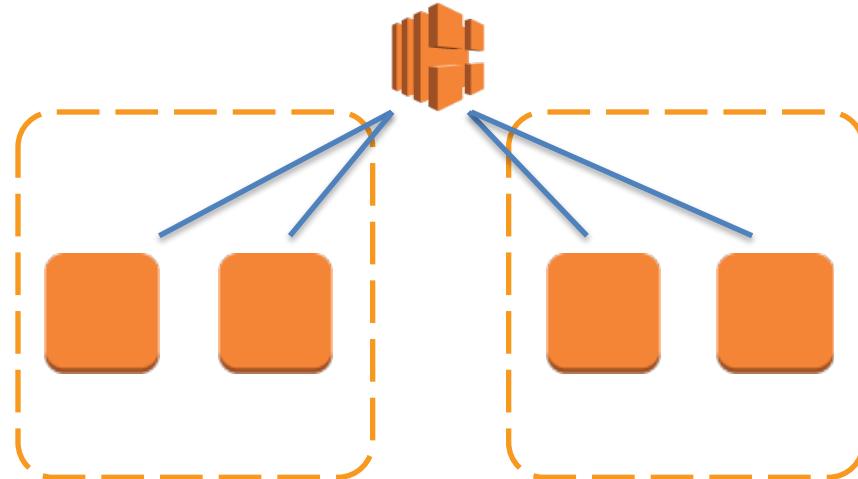
- Setup: Recipes that are executed once an instance is booted
- Configure: Recipes that are executed when “configuration” events
- Deploy: Recipes that are executed with the deploy command
- Undeploy: Recipes that are executed when undeploy is run
- Shutdown: Recipes that are executed when instances are shutdown and before the instance is terminated



Types of code deployments In OpsWorks

Rolling Deployment

- Roll out the code to individual or subset of instances one at time
 - Can remove the instances from the load balancer while they are being deployed to avoid any issues

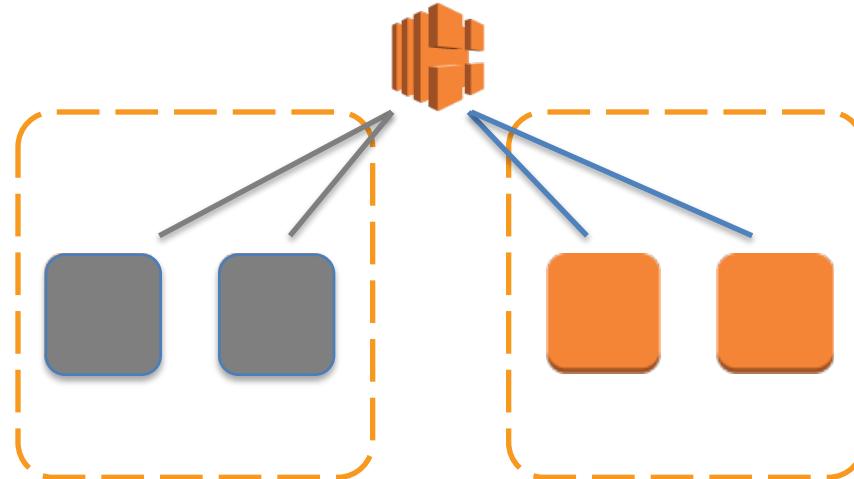




Types of code deployments In OpsWorks

Rolling Deployment

- Roll out the code to individual or subset of instances one at time
 - Can remove the instances from the load balancer while they are being deployed to avoid any issues



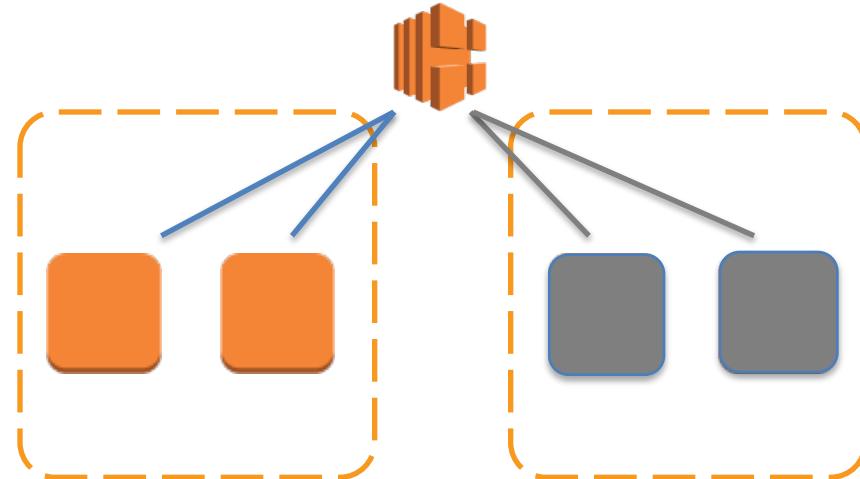
* Remove the instances from the load balancer and deploy the new app code



Types of code deployments In OpsWorks

Rolling Deployment

- Roll out the code to individual or subset of instances one at time
 - Can remove the instances from the load balancer while they are being deployed to avoid any issues



* Once testing verifies app deployment is good then add then back and do the same for the other set



Separate Stacks (blue-green deployment)

Dev.example.com



Staging.example.com

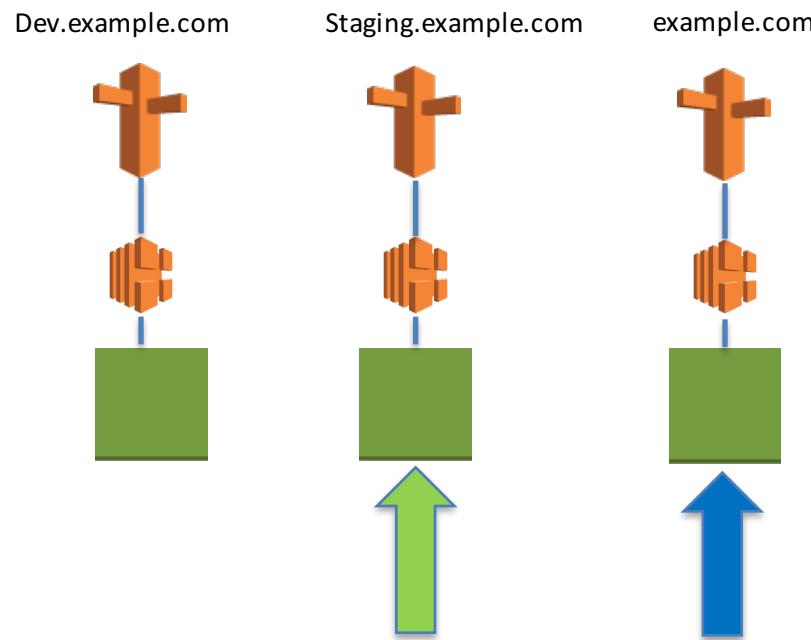


example.com





Separate Stacks (blue/green deployment)



- Staging is the green deployment
- Blue is the current production deployment
- After code is in dev deploy to staging
- After code is tested in staging instead of deploying to prod change DNS weighted to point to the current staging making it prod
- Terminate the old production stack

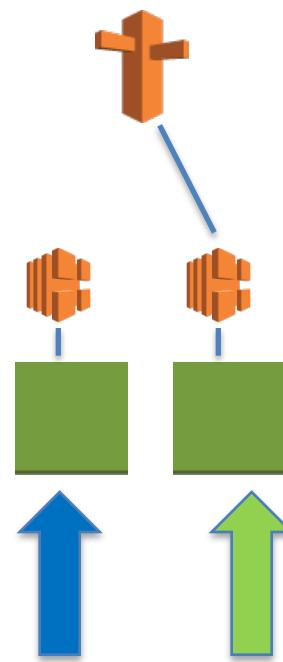


Separate Stacks (blue/green deployment)

Dev.example.com



example.com



- Weights of zero on example.com to a set of load balancers
- Assign those load balancers to the stack
- Increase the weights to the green (staging deployment)
- DNS caching is not an issue since the ELBS were set as weight 0



Linux Academy

Amazon Web Services

SQS Message Priority



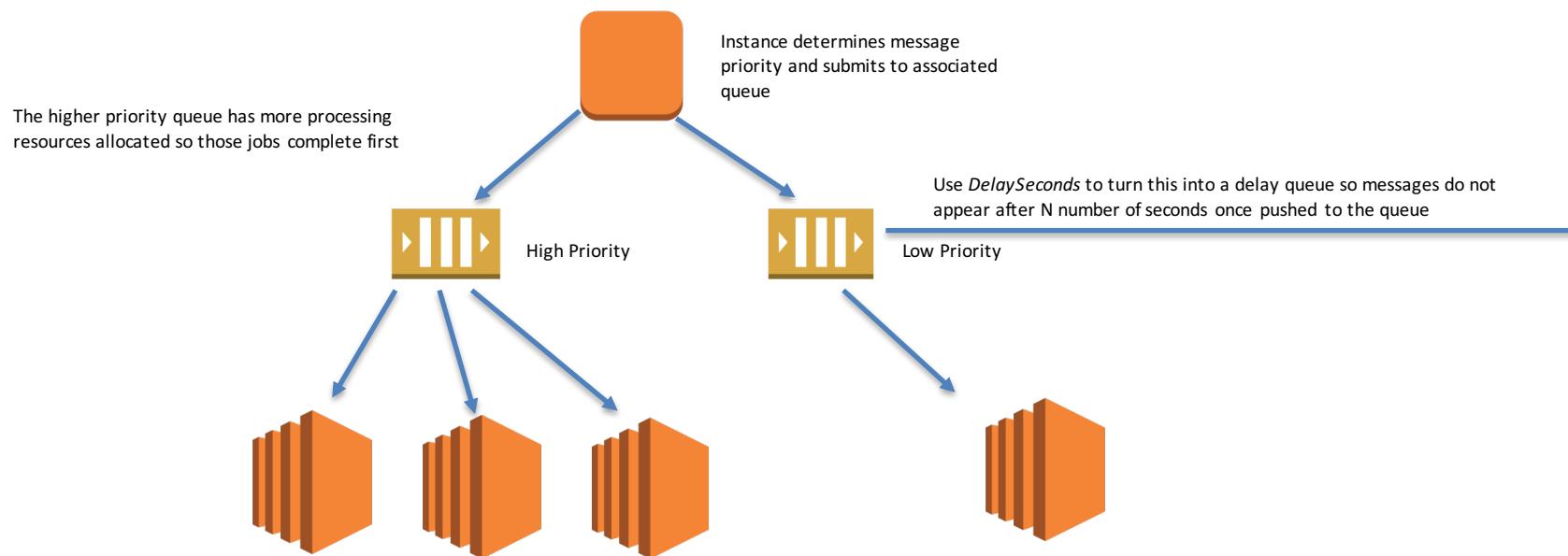
AWS SQS: SQS Message Priority

SQS is used for creating distributed architectures and is commonly used for batch processing architectures.

If your app has a “free tier” which processes “jobs” but you offer a premium service so those jobs are processed faster. How can you design a message priority queue so that premium jobs are completed first?



AWS SQS: SQS Message Priority



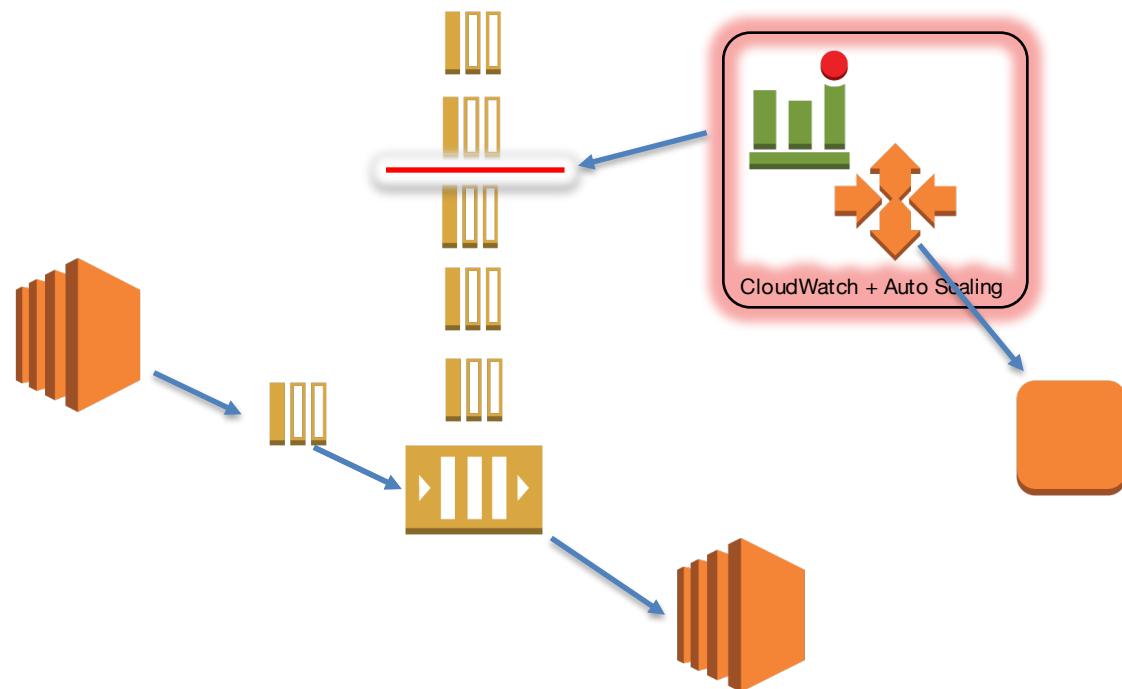


Linux Academy

Amazon Web Services

SQS Job Observer Pattern

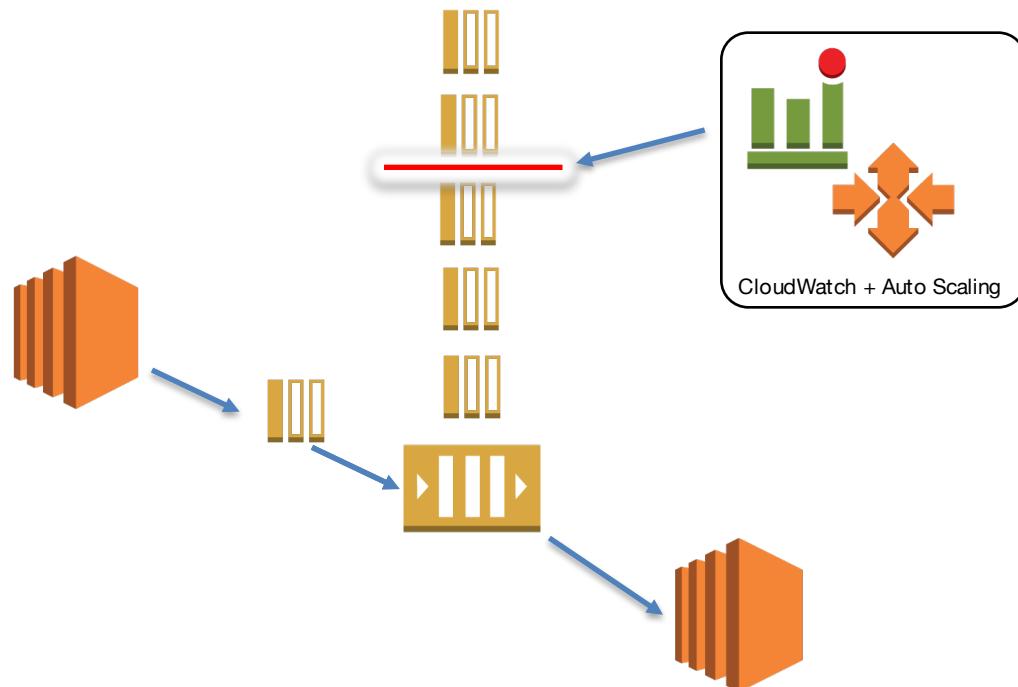
AWS SQS: Job Observer Pattern



http://en.clouddesignpattern.org/index.php/CDP:Priority_Queue_Pattern



AWS SQS: Job Observer Pattern





Linux Academy

Amazon Web Services

DynamoDB Use Cases

AWS SQS: Massive Available Voting app

BCJC contracts with NBC to handle votes for the voice! The voice final show and vote is occurring in two days and BCJC has to build an entire application and architecture that will handle hundreds of millions of votes without failing. The voters will open up a web application to vote yes or no on each candidate. A single vote cannot be missed

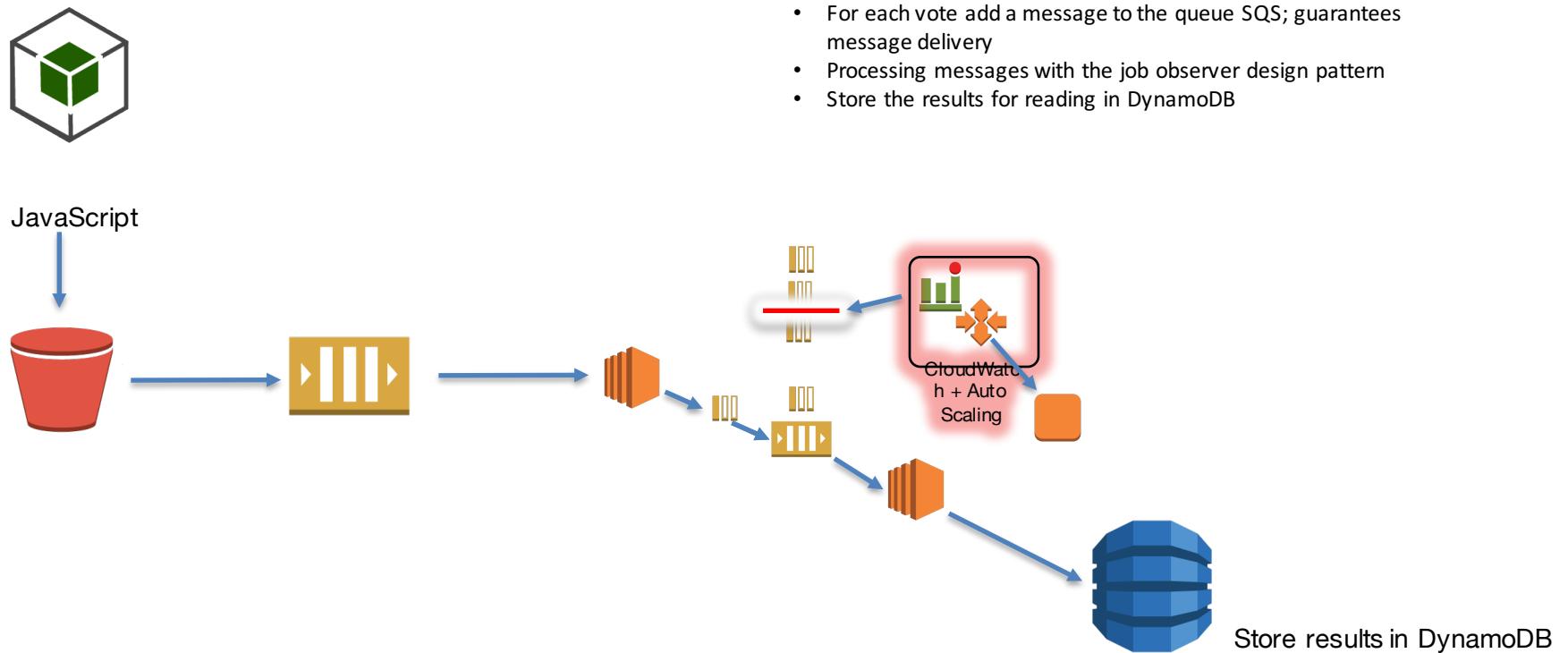
What might be a way to architect this application?

Think “zero down time”

Think “AWS Services”



AWS SQS: Massive Available Voting app





DynamoDB

Problem: Thousands of objects that need to be easily retrieved based off of attribute information.

Solution using just S3: S3 allows you to list objects based off of “key search” basically searching the prefix of the object. The list search is limited to just 1,000 objects.



DynamoDB

Problem: Thousands of objects that need to be easily retrieved based off of attribute information.

Solution using just S3: S3 allows you to list objects based off of “key search” basically searching the prefix of the object. The list search is limited to just 1,000 objects.

Correct Solution: Build an additional index that is easily searchable and stores specific attributes about the object. These attributes can be searched and are linked back to the correct object. Easily searchable solutions like DynamoDB which are require no servers, will reduce cost, and increase efficiency when searching for objects.

DynamoDB

What we already know:

- DynamodDB is made up by provisioning tables and throughput can be increased on each table
- DynamoDB is highly available and scalable
- ElastiCache can be used in front of DynamoDB in order to offload high amounts of reads for non-frequently changed data
- A table is just a collection of items and items are made up of attributes
- Each table requires a primary key and needs to be unique as possible to provision against multiple partitions
- Data is indexed by the primary key
 - Hash key: Now known as Partition Primary Key or Composite key
 - Hash + Range: Range is now known as Sort Key



DynamoDB

Table Name	Primary Key Type	Partition Key Name	Sort Key Name
Course	Simple	Name	
Lesson	Composite	CourseName	LessonName
Notes	Composite	Id	CreateDate

An index is created on the Partition Key name and for composite table types the data is stored in sorted order based off of Sort Key Name



DynamoDB

Two ways to search data within a table

1. Using “query” API call:

- Query will be performed against the primary key and a value for the sort key can be passed with a comparison operator.
- Query is the fastest lookup method as it is performed against a stored “index” in the table
- Query an “indexed” itemed is the fastest method of looking up data in the DynamoDB table

2. Using “Scan” API call

- Scan will read every item in the table and search every possible attribute rather than only indexed attributes.
- This is the most taxing available and causes performance issues



Linux Academy

Amazon Web Services

DynamoDB Secondary Indexes

DynamodDB Secondary Indexes

Scan operations occur against the index partition key. However, what if the table has multiple fields that need to be searched by? Scan API calls are very taxing on performance and are slow. How might you solve this issue?

Secondary indexes: Lets you query the data within a table using a secondary key instead of just the primary partition key.

- Global Secondary Index: An index on a new partition key and sort key that are different than that of the defined table
- Local Secondary Index: An index that has the same partition key but a different sort key



Linux Academy

Amazon Web Services

DynamoDB Multi-Region



DynamoDB Multi-Region Design

Problem: Need to migrate data over to a secondary region as part of a “daily” backup operation.

Solution: You can use Data Pipeline to schedule a pipeline that daily migrates data to a DynamoDB table in another region.



DynamoDB Multi-Region Design

Problem: Need to offload requests that are made in other regions so the data lives close to the end user requesting the data which will reduce latency for reading data from the DynamoDB table.

Solution: DynamoDB streams; Streams are essentially an exact order of modifications to a table put inside a log stream (powered by Kensis).

Many different use cases for this type of feature:

- Stream data to multiple regions in near real-time replication of data
- Secondary applications can listen for changes to data and send notifications to end users



Linux Academy

Amazon Web Services

Preparing For The Exam



Linux Academy

Amazon Web Services

Study Methods After Completing The Course

Preparing For The Exam: How To Study After Completing The Course

- DO NOT register for the exam until you have completed the course and met these study best practices. Scheduling the exam first and then studying is a sure way to not be prepared and to rush learning.
- Download the slides for memorization and study.
- Download the “Exam Study Guide” from the “Required Reading” part of the course syllabus. Be sure to follow the study guide as well.
- Use the practice exam system to help get a feel for how much time to spend on each question in the exam.
- Do NOT study the incorrect/correct answers on the results page first. First see what questions you got wrong and research “why”. This helps with understanding the concepts and becoming a qualified CSA professional, rather than just memorizing answers.
- Watching the videos and taking the labs should only be 40% of your studying. You must continue to review the slides and take the self-paced labs in order to ensure you understand and are qualified for the exam.

Preparing For The Exam: How To Study After Completing The Course

- At least read the “required” white pages in the study guide and it is suggested that you also read the “suggested” white papers in the study guide.
- You should spend at least two weeks studying, going back and reviewing videos, asking questions, and reading the slides.
- Never hurts to take an AWS practice exam by AWS or to read AWS services FAQ.
- If you spend time studying and reviewing the course for 1 to 2 weeks after completing the course on linuxacademy.com, your odds of success go up to 90%+ .
- Rushing to pass is the best way to fail the exam.
- Before you take the exam, take three days off of studying then go back and try the practice exam for your third and last time. If you pass then schedule the exam.



Linux Academy

Amazon Web Services

Test Taking Best Practices

Preparing For The Exam: Test Taking Best Practices

The test is 80 questions in 170 minutes. *Practice time management with our practice exam system!*

1. Start from the beginning.
2. Answer each question one by one. If you have no idea about the question, mark it for “review later” and DO NOT select an answer, then move on. Other questions at times help answer previous questions!
3. Read all the potential answers. Do not select one even if you think one is “obviously correct”, until you read all the potential answers.
4. Take turns reading the answers from top down and bottom up.
5. Understand WHAT the question is asking.
 - *A cost savings architecture is not always the same as a FULL highly available architecture*
6. Demo – How to answer questions even if you ARE SURE you know the answer.