CDN and DNS

- AWS route 53 - DNS service
  - Hosted Zones
  - Domain Types
  - Record Types
  - Routing policies
- AWS CloudFront - CDN Network

DNS Service

- Hierarchical distributed naming system
- Phonebook of the internet
  - google.com === 172.217.194.113
- TLD
  - Generic Top level domain
  - Domain name could end with (.watch, .clothing)
- GD
  - Geographic Domains (.au, .uk)

AWS route 53 - DNS service

- DNS Service
- Woks inside and outside of the AWS
- secure and reliable routing of requests

AWS route 53 - Hosted Zone

- information about how to route a domain xyz.com
- Collection of resource record sets hosted by route 53
- Public hosted zone - route traffic in internet
- Private hosted zone - route traffic within VPC
  - We can use any domain we wish

AWS route 53 - record types

- A - IPV4
- AAA - IPV6
- CAA
- CNAME - Used as one name as alias for another
- MX - names of our mail server
- NPATR
- NS - Name server to hosted zones
- PTR - map IP to domain name
- SOA
- SPF - No longer recommended, idetify of the sender email messages
- SRV

- TXT

AWS route 53 - roouting policy

- Simple
- Failover
- Geo-location
  - Continent, country or state in country
  - When overlap, it would choose smallest (Continent+Country matches, chooses country)
  - Restrict access to only few country
- Geo-proximity
  - Policy based on location + users

- Latency
  - Chooses lowest latency resources
- Mulit-value answer
  - DNS request upto 8 records, and randomly picked
- Weighted
  - Weight of the individual resource record dividey by the sum of the total value in resource record
- Abbreviation - SFGGLMW

AWS CloudFront - CDN

- CloudFront is AWS's fault-tolerant and globally scalable content delivery network service.
- Speeds up distribution of your static and dynamic content through its worldwide network of edge locations.
- Data is cached, after a set period, this cached data will expire and so AWS CloudFront doesn't provide durability of your data.

AWS CloudFront Distribution

- Web Distribution
  - Static and dynamic content (html, image, javascript)
  - MediaFiles using HTTP/HTTPS
  - ADD/Upate/Delete/Submit data from webforms
  - Live streaming to stream an event real time
- RTMP Distribution
  - Streaming distribution using Adobe Flash Media Service RTMP protocol
  - RTMP doesn't require all media to be downloaded to view
  - It should be served from S3, can't be served from webserver

AWS - OAI (Origin access identiy)

- Works only with S3 based content
- Prevents direct access, can't circumvent

AWS - Edge distribution

- Which edge locations
- Different caching behaviour
- WAF (ACL) can be enabled
-

Amazon Aurora (DB cluster)

- Database engine that's compatible with MySQL and PostgreSQL.
- Aurora doesn't use local storage for the compute instances.
- It has high-performance storage subsystem.
- Involves entire clusters of database servers that are synchronized through replication, instead of individual database instances.
- Aurora cluster volume is a virtual database storage volume that spans multiple Availability Zones, with each Availability Zone having a copy of the DB cluster data.
- Not supported in the region that has lesser than 3 AZ
- Three types of DB instances
  - Primary DB Instance
  - Aurora Replica
    - Connects to the same storage volume as the primary DB instance and supports only read operations.
    - Aurora automatically fails over to an Aurora Replica in case the primary DB instance becomes unavailable.
    - Aurora multi-master clusters, all DB instances have read/write capability.

Amazon Aurora Connection Management

- Host name and port that we specify point to an intermediate handler called an endpoint.
- Types of Aurora Endpoints
  - Cluster endpoint
    - mydbcluster.cluster-123456789012.us-east-1.rds.amazonaws.com:3306
  - Reader endpoint
    - mydbcluster.cluster-ro-123456789012.us-east-1.rds.amazonaws.com:3306
  - Custom endpoint

        * Define a custom endpoint to connect to instances that use a particular AWS instance class or a particular DB parameter group
        * Might need to direct internal users to low-capacity instances for report generation or ad hoc (one-time) querying, and direct production traffic to high-capacity instances.
        * myendpoint.cluster-custom-123456789012.us-east-1.rds.amazonaws.com:3306
    - Instance endpoint
        * mydbinstance.123456789012.us-east-1.rds.amazonaws.com:3306

Viewing the Endpoints for an Aurora Cluster

- aws rds describe-db-clusters –query '*.{Endpoint:Endpoint,ReaderEndpoint:ReaderEndpoint,CustomEndpo

Amazon Aurora Storage and Reliability

- Distributed and shared storage architecture
- Cluster volume, which is a single, virtual volume that uses solid state drives (SSDs).
    - A cluster volume consists of copies of the data across multiple Availability Zones in a single AWS Region.
- Aurora doesn't make a new copy of the table data. Instead, the DB instance connects to the shared volume that already contains all your data.

Aurora Replicas

- Up to 15 Aurora Replicas can be distributed across the Availability Zones that a DB cluster spans within an AWS Region.
- Aurora Replicas work well for read scaling because they are fully dedicated to read operations on your cluster volume.

Aurora Serverless

- Aurora database product without managing the resources.
- For a cluster, you can set a min and max ACU (Aurora Capacity Units) based on the load and can even go down to 0 to be paused.
- Only billed for storage consumed when the instances are zero.
- Billing is based on resources used on a per-second basis.
- Aurora Serverless - Use Cases
    - Infrequently used applications.
    - Low volume blog site.
    - You only pay for resources as you consume them on a per second basis.

Referene

- Amazon Aurora## Why we need Cloudtrail? (Auditlog)

- Event history of AWS account activity,
  - Including actions taken through the AWS Management Console
  - AWS SDKs, command line tools, and other AWS services.
- Who created new EC2 instances (API/console/AutoScale/User)?
- Enables governance, compliance, operational auditing, and risk auditing of your AWS account
- Audit events for usage of AWS API
- CloudTrail log files to troubleshoot operational or security issues in AWS account.
- CloudWatch is monitoring, whereas CloudTrail is auditing.

AWS CloudTrail events

- An event in CloudTrail is the record of an activity in an AWS account. This activity can be an action taken by a user, role, or service that is monitorable by CloudTrail.
- CloudTrail: management events and data events. By default, trails log management events, but not data events.
- CloudTrail events that are sent to CloudWatch Logs can trigger alarms according to the metric filters you define.
- Beginning on April 12, 2019, trails will be viewable only in the AWS Regions where they log events. If you create a trail that logs events in all AWS Regions, it will appear in the console in all AWS Regions. If you create a trail that only logs events in a single AWS Region, you can view and manage it only in that AWS Region.

AWS CloudTrail integration with S3

- CloudTrail records events in each region and delivers the CloudTrail event log files to an S3 bucket
- CloudTrail must have the required permissions, and it cannot be configured as a Requester Pays bucket. CloudTrail automatically attaches the required permissions to a bucket when you create an Amazon S3 bucket as part of creating or updating a trail in the CloudTrail console.
- Logs are created every 5 mintes, but it takes sometime 15 minutes to deliver to S3 (it is not realtime).
- KMS can be used to encrypt the log files
- SNS can notify the events (with or withou CloudWatch)
- EventSelectors can be used to adjust the CloudTrail

CloudTrail meta-data

- User-agent can be used to find the source of event

  - "userAgent": "aws-cli/1.3.2 Python/2.7.5 Windows/7"
- CloudTrail can be configured to deliver log files from multiple regions to a single S3 bucket for a single account.
- eventName
- eventSource - service that generated event
- SourceIPAddress - find the IP address
- "userAgent" : {'Signin.amazonaws.com', 'Console.amazonaws.com', lambda.amazonaws.com'}

CloudTrail log-file, bucket folder name convention

- AccountID_CloudTrail_RegionName_YYYYMMDDTHHmmZZ_UniqueString.FileNameFormat (.json.gz) is the extension
- BucketName/prefix/AWSLogs/AccountID/CloudTrail/RegionName/YYYY/MM/DD

AWS CloudTrail logs (sample)


```
{"Records": [{
    "eventVersion": "1.0",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::123456789012:user/Alice",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "accountId": "123456789012",
        "userName": "Alice"
    },
    "eventTime": "2014-03-06T21:22:54Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "StartInstances",
    "awsRegion": "us-east-2",
    "sourceIPAddress": "205.251.233.176",
    "userAgent": "ec2-api-tools 1.6.12.2",
    "requestParameters": {"instancesSet": {"items": [{"instanceId": "i-ebeaf9e2"}]}},
    "responseElements": {"instancesSet": {"items": [{
        "instanceId": "i-ebeaf9e2",
        "currentState": {
            "code": 0,
            "name": "pending"
        },
        "previousState": {
            "code": 80,
            "name": "stopped"
        }
    }]}}
}]}
```

AWS CloudTrail integrity

- We can validate integrity of the log cloudtrail log files
- Used for security and integrity
- SHA-256 is used for CloudTrail
- CloudTrail creates digest every hour
- Digest files are signed by private key of key pair
- `bash aws cloudtrail validate-logs --trail-arn <trailarn> --start-time <start-time> --endtime <endtime> --s3-bucket <bucketname> --s3-prefix <s3-prefix> --verbose`

AWS CloudTrail Secondary account Account-B accessing log of primary account (Account-A)

- Create new role
- Apply policy to only allow access for Account-B's folder in S3
- Establish trus relationship between AccountA and AccountB
- Create new account in AccountB
- Create a policy and apply sts:AssumeRole to this user in AccountB
- Example steps
  - Swith to Account-A (Primary)
  - Create new role named 'Cross-Account-CloudTrail-Log' and assign Role-For cross account access
  - Select secondary account-id for that we need provide access (in Role-For cross account)
  - Attach a policy to this role (allow read-only acccess to bucket)
  - Swith to Account-B (Primary)
  - Create policy for AssumeRoleCloudTrail
  - Create user-xyz in Account-B
  - And attache AssumeRoleCloudTrail policy to user-xyz in Account-B

AWS CloudTrail S3 policy for readonly account and AssumeRlePolicy

```
{
 "Version":"2012-10-17",
 "Statement":[
    {
       "Effect":"Allow",
       "Action":[
          "s3:GetBucket",
          "s3:ListBucket",
       ],
       "Resource":"arn:aws:s3:::awsexamplebucket"
    }
 ]
```

```
}
```

```
  {
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":["sts:AssumeRole"],
         "Resource":"arn:aws:iam::primary-account-A:role/Cross-account-cloudtrail"
      }
   ]
}
```

AWS CloudTrail can be monitored by CloudWatch

- Useful CloudTrail monitoring by CloudWatch
  - IAM SecurityPolicy
  - EC2 SecurityGroup changes
  - API Calls requestion significant resource changes
  - EC2 events (start/stop/reboot/scaling)
  - Failed login attempts to management console
  - UnSuccessful attempts
- Steps to configure
  - Create new CloudTrail
  - CloudTrail has to use CloudWatch LogGroups (roles/permissions)
  - CloudTrails should be able to create new log-group to log API/mangement events
  - Configure CloudTrail_CloudWatchLogs_Role for new Trail
  - Configure MetricFilter for monitoring/dashboard/alarm
- CloudWatch has 256KB size limits, CloudTrail won't send events larger than 256KB
- aws_lab_cloudtrail_monitor_alarm.md

AWS CloudFront Access Logs

- CloudFront === CDN
- Logs are distributed across regions
- CloudFront can forward the logs to S3 (only storage is cost)
- The name of each log file that CloudFront saves to S3 bucket uses the file name format: //.YYYY-MM-DD-HH.unique-ID.gz
  - butcket-name.s3.amazonaws.com/optional-prefix/EMLARXS9EXAMPLE.2019-11-14-20.RT4KCN4SGK9.gz
  - Distribution ID: EMLARXS9EXAMPLE
- Steps to enable
  - CloudFront > Select Distribution > Distribution Settings > General Tab > Edit > "Logging:On, Bucket for Logs: CloudFrontAccessLogs,

LogPrefix: access, Cookie Logging: On"
  - CookieLogging is only required when orgin is not S3 (such as EC2)
- To Enable logging, user should have following role
  - FULL_Control
  - s3:GetBucketAcl/s3:PutBucketAcl
- Why ACL Role for CloudFront?
  - The user account activating log to S3 must have full control on the ACL for the S3 bucket, along with the S3 GetBucketAcl and S3 PutBucketAcl.
  - The reason is during the configuration process, CloudFront will use your credentials to add the AWS data-feeds account to the ACL with full control access.
  - data-feeds account used by AWS which will write the data to the log file and deliver it to your designated S3 login bucket.
  - Therefore, if you're trying to enable the loging feature for your distribution and it's failing, then you should check your access to ensure you have the required permissions.

AWS VPC Flow logs

- VPC has 100s or 1000s of resources connected and communicating each other
- Capture IP Information that flow within your VPC
- VPC Flow logs are not sent to S3, they are sent to CloudWatch
- EC2-classic is not in scope
- VPC Flow logs has been created, it can't be changed (delete/recreate)
- Many limitations (DNS can't be captured)
- What is Flow logs are allowed?
  - A network interace on one of your EC2 instance
  - A subnet within VPC
  - VPC itself
- Few permissions are required
  - logs:CreateLogGroup, logs:CreateLogStream, logs:PutLogEvents, logs:DescribeLogGroups, logs:DescribeLogStreams
  - ec2:CreateFlowLogs, ec2:DescribeFlowLogs, ec2:DeleteFlowLogs, logs:GetLogData, iam:passrole
- Steps for VPC flow-logs (ENI/Subnet/VPC)
  - Mgmt Console > Network and Security > Network Interface > select ENI > "Flow Logs: Tab" > "Create Flow Log" > "Filter: Accept/All/Reject" > Select Role to push to CloudWathLog > "Destination Log Group"
  - Mgmt Console > VPC > Subnet > PublicSubnet > FlowLogs >... (just like above)
  - Similar steps for VPC
- Flow log record - Available fields
  - version

  - account-id
  - interface-id
  - srcaddr
  - dstaddr
  - srcport
  - dstport
  - protocol
  - packets
  - bytes
  - start
  - end
  - action (if security group rejected, we have clue here)
  - log-status

AWS Config

- Why AWS config?
  - What resources we have? can we save some money?
  - Are there any security vulnerabilities?
  - How are resources linked within the environment
  - Is infrastructure are compliant to compliance
- Resource Management (using AWS Config)
  - AWS Config can capture resource changes
  - Act as resource inventory
  - Stores configuration history
  - Provide snapshot of all resources (and configurations)
  - Notification changes
  - Use rules to check compliance
  - Perform security analysis
  - Identify relationship
- AWS Config is region specific

AWS Config - Components

- AWS Resource
- AWS Configuration item (json file)
  - Configuration information
  - Relationship information
  - Other metadata
  - Resource changes are automatically recorded as CI and sent to Configuration Stream
  - Configuration history/Configuration streams/Configuration snapshots
- Configuration stream
  - When configuration history files are arrived
  - When configuration snapshot started
  - State of compliance changes for a resource

- Configuration history
  - aws configservice get-resource-config-history –resource-type AWS:EC2::Subnet –resource-id subnet-ef22b32b7
  - Can be delivered every 6 hours to S3 bucket
- Configuration snapshot (for a given time to s3 bucket)
- Configuration recorder
  - Core component that records CI details
  - Can be started/stopped
  - Resource filter can be applied
- Config Rules (help to enforce compliance rules?)
  - Can trigger lambda function
  - AWS Predefined rules (below are few samples)
    - RDS-Storage-Encrypted
    - Encrypted volume
    - Root-Account-MFA-Enabled
    - IAM-User-NOPolicy-Check
- Resource Relationships (dependency matrix?)
- SNS Topic can be used for notification
- S3 bucket can store configuration information
- AWS Config permission
  - Requires read only permission to record all information
  - Requires access to S3 and SNS

CloudTrail lab notes

- CloudTrail records the last 90 days of events for AWS accounts.
- The default events do not support triggering alerts, event metrics, and long term storage. We require Trail for that.
-

Configuration Item


```
[
  {
    "version": "1.3",
    "accountId": "777719877733",
    "configurationItemCaptureTime": "2020-07-11T23:57:46.383Z",
    "configurationItemStatus": "ResourceDiscovered",
    "configurationStateId": "1594511866383",
    "configurationItemMD5Hash": "",
    "arn": "arn:aws:ec2:us-west-2:777719877733:security-group/sg-0bf9c8e5ea6375194",
    "resourceType": "AWS::EC2::SecurityGroup",
    "resourceId": "sg-0bf9c8e5ea6375194",
    "resourceName": "default",
    "awsRegion": "us-west-2",
```

```
"availabilityZone": "Not Applicable",
"tags": {},
"relatedEvents": [],
"relationships": [
  {
    "resourceType": "AWS::EC2::VPC",
    "resourceId": "vpc-009882a2845337a77",
    "relationshipName": "Is contained in Vpc"
  }
],
"configuration": {
  "description": "default VPC security group",
  "groupName": "default",
  "ipPermissions": [
    {
      "fromPort": 80,
      "ipProtocol": "tcp",
      "ipv6Ranges": [],
      "prefixListIds": [],
      "toPort": 80,
      "userIdGroupPairs": [],
      "ipv4Ranges": [
        {
          "cidrIp": "0.0.0.0/0"
        }
      ],
      "ipRanges": [
        "0.0.0.0/0"
      ]
    },
    {
      "ipProtocol": "-1",
      "ipv6Ranges": [],
      "prefixListIds": [],
      "userIdGroupPairs": [
        {
          "groupId": "sg-0bf9c8e5ea6375194",
          "userId": "777719877733"
        }
      ],
      "ipv4Ranges": [],
      "ipRanges": []
    },
    {
      "fromPort": 3389,
      "ipProtocol": "tcp",
```

```
      "ipv6Ranges": [],
      "prefixListIds": [],
      "toPort": 3389,
      "userIdGroupPairs": [],
      "ipv4Ranges": [
        {
          "cidrIp": "0.0.0.0/0"
        }
      ],
      "ipRanges": [
        "0.0.0.0/0"
      ]
    },
    {
      "fromPort": 443,
      "ipProtocol": "tcp",
      "ipv6Ranges": [],
      "prefixListIds": [],
      "toPort": 443,
      "userIdGroupPairs": [],
      "ipv4Ranges": [
        {
          "cidrIp": "0.0.0.0/0"
        }
      ],
      "ipRanges": [
        "0.0.0.0/0"
      ]
    }
  ],
  "ownerId": "777719877733",
  "groupId": "sg-0bf9c8e5ea6375194",
  "ipPermissionsEgress": [
    {
      "ipProtocol": "-1",
      "ipv6Ranges": [],
      "prefixListIds": [],
      "userIdGroupPairs": [],
      "ipv4Ranges": [
        {
          "cidrIp": "0.0.0.0/0"
        }
      ],
      "ipRanges": [
        "0.0.0.0/0"
      ]
```

```
        }
      ],
      "tags": [],
      "vpcId": "vpc-009882a2845337a77"
    },
    "supplementaryConfiguration": {}
  }
]
```


- Getting Started with AWS CloudTrail Tutorial
- AWS Sample logs
- Cloud Trail documentation
- AWS CloudTrail User guid
- AWS CloudTrail Update – SSE-KMS Encryption & Log File Integrity Verification## AWS CloudWatch
- Used for
  - Tracking metrics
  - Trending
  - Setting Alarms
  - Consolidated Resource Logging
- Comprehensive monitoring tool, could react to events
- Create Trend reports
- Example: Monitor EC2, and setup alarm, Concern about recurrence Billing

AWS CloudWatch - Overview

- Geneos + ELK
- Region specific billing
- Free 5 minutes refreshed monitoring with 3 dashboard is free
- Enhanced monitoring refresh rate is 1 minutes
- AWS CloudWatch - Logging
  - Similar to ELK, but not applicable for complex applications
  - There is no logstash equivalent in CloudWatch, you need lambda function
  - Search features are not as efficient/feature-rich like kibana
  - Kibana dashboard has more options and well suited for logs from multiple-sources
  - We can export logs from CloudWatch Log and analyze using Splunk/ELK/3rd party

AWS CloudWatch - monitoring theory

- Why bother monitoring?
- Customer experience: Are my customers getting good experience?
- Performance and costs: How are my charges impacting overall performance
- Trends: Do I need to scale my environment?
- Troubleshooting and radiation: Where did the problem occur?
- Learning and improvement: Can we detect and prevent this problem in the future?

AWS CloudWatch - How to?

- Launch EC2
- Select EC2 > Monitoring

AWS CloudWatch - What could be monitored?

- CPU (Utilization/Credit-Usage/Credit-Balance)
- Network (Byte-In/Byte-Out/Packets-In/Packets-Out/Status-Check)
- Disk (Read/Write/inBytes/Count of operations)
- What is missing in basic?
  - How much memory is left?
  - Disk space level monitoring?

AWS CloudWatch - Build Dashboard

- AWS CloudWatch > Dashboard > CreateDashboard > "DashboardName"
- Add Configuration (Line/Stacked-Area/Number/Text) to Dashboard
- Choose EBS or EC2 Metrics > Per-Instance Metrics > "Cpu Utilization" > Create Widget > Save Dashboard

AWS CloudWatch - Monitor EC2?

- EC2 requires additional role with CloudWatchFullAccess (policy) for EC2, so that it could be monitored by CloudWatch with more the details.
  - We can create upfront - EC2WithCloudWatchFullAccess (role)
- Configure monitoring script available in AWS documentation using 'crontab -e'
- AWS CloudWatch > Metrics > Select Metrics > "ADD to Dashboard" > "Select Dashboard that you already created"
- Chaging GrphOptions (from text to trend-line)
  - Select Widget (any widget) > Graph Options > "Line (from Number or vice Versa)" > Update Widget
- Save Dashboard (often)

AWS CloudWatch - Sending files to CloudWatch

- CloudWatch acts as simple repository for logs
- Its analytics is not as sophisticated like elk/splunk
- What Is Amazon CloudWatch Logs?
- CloudWatchLogFull access policy can be added to roles that requires logging
- EC2 instances can have user-id assigned by storing keys (but less convenies compared to assigning roles), Roles can be assigned dynamically to running instance
- View CloudWatchLog
  - AWS CloudWatch > Logs > /var/log/messages > LogStreams (select one) > view log-messages
- Rotate logs
  - Select log-groups > logs > Select "Expire Events After" > "2 weeks" (instead of never)
  - They are stored in S3 by default, without changing log will be kept forever

Monitor Apache ec2 logs

```
# Ensure EC2 has role that has policy CloudWatchFullAccess CloudWatchLogFullAccess
sudo yum update -y
sudo yum install -y awslogs
sudo chkconfig awlogs on
cd /etc/awslogs/
vi /etc/awslogs/awscli.conf ## Edit the region=us-west1 to us-west1
sudo service awslogs start
```

Monitor Apache httpd access logs

```
# Ensure EC2 has role that has policy CloudWatchFullAccess CloudWatchLogFullAccess
sudo yum install httpd
sudo service httpd start
sudo su
cd /var/log/httpd ## access_log error_log are listed
vi /etc/awslogs/awslogs.conf
#copy whatever available for /var/log/messages to /var/log/httpd/access_log (clone/paste)
service awslogs restart
```

Sample awslogs.conf

```
[general]
state_file = /var/awslogs/state/agent-state

[syslog]
```

```
datetime_format = %Y-%m-%d %H:%M:%S
file = /var/log/syslog
buffer_duration = 5000
log_stream_name = {instance_id}-{hostname}-{ip_address}-syslog
initial_position = start_of_file
log_group_name = ecs
```

AWS CloudWatch Alarms

- Thresholds are upper/lower tollerances.
- Simple healthcheck setup (via) Route53
  - EC2> Select Security Group > Open inbound port 80
  - cd /var/www/html/
  - echo 'success' > healthcheck.html
  - curl http://aws-ec2-dns/healthcheck.html
  - Goto Route53 > HealthCheck > Create Health Check > "Configure above url"
  - Goto Route53 > HealthCheck > Select Health Check > Selct Alarms Tab
  - Create Alarm > Name: "Server Down" > "Send Notification": Yes > "Topic Name"; Server Down > "Receipient Email Address": "mm@mm.com"
  - In Sufficient Data : "Wait for 1 minute" (just created, wait for enough data)

AWS CloudWatch Alarms

- CloudWatch > Alarms > CreateAlarm > Linux System Metrics > FileSystemInstanceIdMountPath
- DiskspaceUsed > Select "Root Volume" > Next > Name: "name" && >= "80" (Period minutes/1 hr/6hr/days/) > "Average" (Treat missing data as bad or good using dropdown)
- Send notification to : "Email" > Create Alarm
- Email should be confirmed within 72 hours

Reference

- Monitoring memory and disk metrics for Amazon EC2 Linux instances
- What Is Amazon CloudWatch Logs?
- Quick Start: Install and Configure the CloudWatch Logs Agent on a Running EC2 Linux Instance
- Sample loggroups awslogs.conf## Planning for AWS
- Choosing instance type based on general recommendations can result in signficant overspend and performance penalty

  - We should choose based on specific requirement of the application
  - Use on-premise footprint metrics on your existing environment
- Optimize for performance and cost (high performance might also cost more)
- Use automation to get it right from the begining
- Model entire application instead of instance-type
- Cloud requires automation (it shouldn't be considered as extension to existing legacy replacement)

AWS Cost optimization

- Continiously measure
- Pay for what you need than "Pay for what you use"
- Use consolidated billing
- Use AWS Orgnization to centrally manage governance policies
- Prefer convertible RI if not standard RI

AWS Governance

- AWS native tools vs CMP (cloud management platform)
- Procure RI centrally
- Use master account for centrally procure, manage the lifecycle of RI
- Use resource tagging
  - Useful in many ways
- AWS Budgets for quotas and entitlements

Think beyond EC2

- Poorly architected lambda would cost more than EC2 equivalent
  - Lambda cost is based number of invocation and memory allocation

Reference

- Top 10 Strategies to Manage Cost and Continuously Optimize AWS # Badri Venkatachari ## AWS Database
- AWS Non Relational database
  - Amazon DynamoDB
  - Amazon ElasticCache
  - Amazon Neptune
- AWS Relational database
  - Amazon RDS (Relational Database service)
    * Commercial

        - Microsoft SQL Server for RDS
        - Oracle for RDS
    - Amazon Aurora for RDS
        - PostgreSQL
        - MySQL for RDS
    - Amazon Community DB
        - PostgreSQL
        - MySQL
        - MariaDB
- Amazon RedShift

AWS Non-Relational database

- Amazon DynamoDB
    - Key-Value
    - Document
- Amazon ElasticCache
    - Redis
    - Memcache
- Amazon Neptune
    - BlazeGraph
- Features
    - No table schema
    - Fast and secure data store
    - Lack of processing engine

AWS Cloud Database

- Same as what we use, but renting instead of owning
- Operated by cloud, no upfront license cost
- Infrastructure and everything is provisioned on demand
- Majority of the patch is owned by AWS
- Autoscaling
- Backup managed
- AWS KMS

AWS DynamoDB

- It is web-service based database and can massively scale for OLTP applications
- Fully managed service
- Document and key-store object
- Local version for testing is accessible

- Supports encryption at rest
- Speed and Performance

AWS ElastiCache

- Memcached vs Redis
- Redis
  - Lots of features
  - Supports multiple data-types (string, hashes, lists, sets and sorted-sets, bitmaps)
- Memcached
  - Simpler
  - Need to run on multi-core and multi-threads
  - Scale-in/scale-out

AWS Neptune

- Graph Database based on blazegraph
- Knowledge graphs
  - W3C RDF
  - Property Graph
  - Sparql, Gremlin

AWS RDS (instead of own EC2)

- scale compute metrics in or out independently
  - processor size, the amount of storage, or the IOPS speed, independently of each other.
- The automatic backups and patching.
- RDS run a synchronous, or asynchronous version of your database, in a different availability zone. (multi AZ support for all RDS)
- Automatic failure detection and recovery

Amazon RDS for MySQL

- Supports 5.5, 5.6, 5.7 and 8.0
- Four instance types - micro-instances, general purpose, memory optimized, and the burstable
- Can create read replicas of database and deploy those across more than one availability zone, using the multi-AZ feature of MySQL and RDS.
- Increase the size of your database storage on the fly, with zero down time.
- Point-in-time restore and snapshot restore features for InnoDB storage engine.
- MariaDB also supported

Amazon RDS for Microsoft SQL Server

- Default database size is four terabytes
- Create a database up to 16 terabytes, and as a managed service you can select a multi-AZ deployment
- provision from 1000 IOPS to 32,000 IOPS for new SQL Server DB
- The transaction log is backed up at five-minute intervals enabling point-in-time recovery to any one given second.
- Automatic backups can be stored for up to 35 days.

Amazon RDS for Oracle

- BYOL
- Supports multiple edition - SE, SE1, SE2 and EE
- Amazon Redshift is a supportive data-source for Oracle Business Intelligence versions 12.2.1.0 and 12.2.1.1

Amazon RDS for Postgres

- Supports from 9.3 to 12.3 on Amazon RDS

Amazon Aurora, the cloud native database from Amazon

- Amazon's own fork of MySQL and PostgreSQL
- Designed and built from the ground up to be cloud native
- Amazon Aurora replicates data across three availability zones by default
- Aurora service also deploys a cloud native database cluster and this database cluster as the underlying data store
- Each cluster has one primary instance which performs all of the data modifications to the cluster volume and supports read and write operations.
- Each cluster also has at least one Aurora replica which supports only read operations
- Aurora DB cluster can have up to 15 Aurora replicas of the primary instance
- Multiple Aurora replicas distribute the read workload

Creating DB service

- Choose DB type and version
- Choose instance type
- Choose storage and provisioned IOPS
- Choose DB instance name and password
- Create VPC
- Select AZ
- Encrypt database at rest and AWS KMS details
- Backup retention and backup window
- Maintenance window (for patching)

Choosing between Database types

- Table join and structured query at database layer or code?
- Do you need cache?
- Is it OLTP vs OLAP
- Do we need internet scale database?

Reference

- AWS Database Options## AWS Deepracer - for whom?
- Reinforcement learning tech - platform
- Do you want to create a machine learning model that makes inferences for a fictitious scenario just doesn't cut it in terms of interest factor.
- Maybe you're just keen to experiment with reinforcement learning tech in a global community context where you can share ideas.

AWS Deepracer - Overview?

- AWS DeepRacer is a new global racing league for autonomous handheld sized racing cars.
- Build and train a reinforcement learning model that can be uploaded into an autonomous handled sized car.
- The RL model is then used in conjunction with the onboard camera, gyroscope, and accelerometer sensors to guide it around a racing track as quickly as possible.
- fastest willll be at the top of the leader board.
- Simulated vs real evnts
- Anyone can purchase their own AWS DeepRacer to autonomous car, which is being sold through amazon.com The car is a 1:18th scale vehicle which has a camera at the front as its main guiding sensor.
- Sensors of the cars
  - CPU Intel Atom - 4GB RAM
  - 32GB wifi, 4MP Camera
  - Ubuntu OS 18 or 16
  - Accelerometer and gyroscope

How to win deepracer?

- Building and training your Reinforcement Models, that uses a supported RL framework, algorithm, reward function, and other hyperparameters.
- The reward function is central to the entire outcome and is something you must develop and invest time in correcting and optimizing.
- The reward function will ultimately determine how fast your autonomous car can navigate the course correctly,

- Reward function is implemented as a Python based script, and the logic needs to consider many input variables,
- Parameters
  - The X and Y car location coordinates
  - On or off the track,
  - Displacement from the center line
  - The car orientation
  - The percentage of track completed,
  - The number of steps completed
  - The speed of the car
  - and The steering position.## Find tutorial
- "Getting Started Tutorial"
- "Getting Started Tutorial"
- Getting started with Amazon Simple Storage Service
- Getting started with AWS Lambda## AWS DynamoDB - OVerview
- Components of dynamodb
  - Tables, items, and attributes are the core components
- Fully managed proprietary NoSQL database service that supports key-value and document data structures
- Supported local secondary index, recently global seconday index
- Automatically replicated across multiple Availability Zones within an AWS Region.
- Automatically partition data and incoming traffic across multiple partitions which are themselves stored on numerous backend servers distributed across three availability zones within a single region.

DynamoDB Units

- **Read unit** - One read request unit represents one strongly consistent read request, or two eventually consistent read requests, for an item up to 4 KB in size.
- **write unit** - One write request unit represents one write for an item up to 1 KB in size.
- Use-cases
  - Transactional read requests require 2 read request units to perform one read for items up to 4 KB.
  - if item size is 8 KB
    - 2 read request units to sustain one strongly consistent read
    - 1 read request unit if you choose eventually consistent reads
    - 4 read request units for a transactional read request

  - Transactional write requests require 2 write request units to perform one write for items up to 1 KB.
  - if your item size is 2 KB
    * 2 write request units to sustain one write request
    * 4 write request units for a transactional write request

DynamoDB Keys

- Partition key – A simple primary key, composed of one attribute known as the partition key.
- Partition key and sort key – Referred to as a composite primary key, this type of key is composed of two attributes. The first attribute is the partition key, and the second attribute is the sort key.

Global table - Cross region replication

- In DynamoDB, we can create tables that are automatically replicated across two or more AWS Regions, with full support for multimaster writes.
  - To build fast, massively scaled applications for a global user base without having to manage the replication process
-

DynamoDB supports two kinds of indexes

- Global secondary index – An index with a partition key and sort key that can be different from those on the table.
  - GSI can span all of the data in the base table, across all partitions.
  - Has own provisioned throughput settings for read and write activity that are separate from those of the table.
  - No size limit
- Local secondary index – An index that has the same partition key as the table, but a different sort key.
  - Total size of indexed items for any one partition key value can't exceed 10 GB.
- Each table in DynamoDB can have up to 20 global secondary indexes (default quota) and 5 local secondary indexes.

Point-in-Time Recovery for DynamoDB

- Continuous backups of your DynamoDB table data.
- Optional, has to explicitly enable PITR (Point-in-time-recovery)
- PITR backup can be restored into new table

What is the consistency model of DynamoDB?

- Read can choose the consistency model depending on demand

  - Eventually consistent reads (the default, high throughput)
  - Strongly consistent reads (read from master, low throughput)

AWS DynamoDB DAX (Accelerator)

- in-memory cache for DynamoDB that delivers up to a 10x performance improvement
  - from milliseconds to microseconds – even at millions of requests per second.
- Dynamo DB - milliseconds, DAX - MicroSeconds
- DAX is compatible with DynamoDB (no application code change)
- DAX supporst AES encryption
- DAX doesn't handle table related operations, they are handled by DynamoDB
- All write operations ("write-through") are written to DynamoDB first and later to DAX (DAX is eventual consistency)
- DAX is not ideal
  - Applications that require strongly consistent reads.
  - Applications that do not require microsecond response times for reads.
  - Applications that are write-intensive, or that do not perform much read activity.
  - Applications that are already using a different caching solution with DynamoDB, and are using their own client-side logic for working with that caching solution.

AWS DynamoDB DAX Cluster

- Minimum of 3 nodes, maximum of 10 nodes (1 Primary, 9 Replicas)
- DAX EC2 cluster requires additional role and would use your existing VPC
- DAX EC2 requires additional inbound rule for port 8111
- We should install "DAX Client" software on those EC2 instances
- Read Capacity Units can be reduced since DAX would take care most of the reads

AWS DynamoDB Commands

```
aws dynamodb create-table --region us-west-2 --table-name cloudacademy-courses \
    --key-schema AttributeName=courseid,KeyType=HASH --attribute-definitions AttributeName=c
    --billing-mode PAY_PER_REQUEST

aws dynamodb batch-write-item --request-items file://ProductCatalog.json

aws dynamodb update-continuous-backups --table-name Music --point-in-time-recovery-specifica

aws dynamodb describe-continuous-backups --table-name Music
```

```
aws dynamodb restore-table-to-point-in-time --source-table-name Music --target-table-name Mu
--no-use-latest-restorable-time --restore-date-time 1519257118.0
```

Reference

- https://www.dynamodbguide.com/what-is-dynamo-db
- Amazon Dynamo Paper
- Global Tables## Core AWS Services
- CDSN - Compute, Storage, Database and Network

EC2

- AMI
- Instance Types
- Instance Purchasing Options (On-demand, reserved or spot)
- Tenancy
- User Data
- Storage Options
- Security

AMI

- Preconfigured EC2 instance templates
- Image baseline with operating system and applications
- AWS AMI vs Custom AMI vs Marketplace AMI vs Community AMI

Instance Types

- Instance types
- Attributes
  - Family - Micro-instances/General-purpose/Compute/GPU/FPGA/Memory Optimized-large-scale-in-memory/Storage Optimized
  - Type - t2/t3/(micro/nano/small/medium)
  - ECU
  - vCPU
  - Physical Processor
  - Clock speed
  - Memory GiB
  - Instance Storage (EBS Only)
  - EBS Optimized availability
  - Network performance
  - IPv6 support
  - Processor Architecture
  - AES-NI

  - AVX
  - Turbo

Instance Purchasing Options

- OnDemand
- Scheduled (lesser than on-demand but recurring charge)
- Reserved
- Spot Instances
  - Suddently interrupted
  - Bid based, can be procured cheaper
- OnDemand capacity reservations

Tenancy

- Shared tenancy
- Dedicated tenancy
- Dedicated Hosts

User data

- yum update -y
- Download latest os updates

Storage options

- Persistent storage
- Ephemeral Storage
  - EC2 instance local storage
  - Like laptop hdd
  - Unable to detach instance

Security

- During EC2 creation security group needs to be selected
- Security group decides the protocol, egress and ingress
- Create or use key-pair
- Keypairs are used to encrypt/decrypt login information
- Possible to use the same key-pair on multiple instances
- Shared responsibility model
  - Customer responsibility for os and other security patch updates

Create new EC2 instance

1. AWS Management console

2. EC2 console (under compute)

3. Launch Instance
4. Select AMI (Amazon linux 2 AMI)
5. Choose instance type
6. Configure instance details
    1. Number of instance, purchasing option, network, subnet, auto-assign public IP, placement group, capcity reservation, IAM role, shutdown behaviour (stop/terminate), Enable terminate protection, monitoring, tenancy
7. Next add storage
    1. Volume-Type (root/ebs), device (/dev/xvda), Snapshot, Size, Volume-Type (SSD/HDD), IOPS, Throughput, DoT (Delete on Termination), Encrypted
8. Add Tags (50 tags maximum)
9. Configure Security Group Rules
    1. Type (SSH,HTTP,LDAP)
    2. Protocol (TCP/UDP)
    3. Port Range
    4. Source
    5. Description
10. Launch (Create keypair)

Get the EC2 instance meta-data

```
curl  -w "\n" http://ec2-54-185-39-21.us-west-2.compute.amazonaws.com/meta-data
curl http://169.254.169.254/latest/meta-data/ami-id
curl http://ec2-54-185-39-21.us-west-2.compute.amazonaws.com/meta-data
```

Key files

- PEM - Privacy Enhanced Mail (.pem)
    - Not natively supported by putty
- PEK - PuTTY Private Key (.ppk)
    - PuTTYgen - required to register this file

Connecting to EC2 instance

```
chmod 400 mohan_ec2.pem
ec2-54-185-39-21.us-west-2.compute.amazonaws.com
ssh -i "/path/to/your/mohan_ec2.pem" ec2-user@ec2-54-185-39-21.us-west-2.compute.amazonaws.c
ssh -i mohan_ec2.pem ec2-user@ec2-54-185-39-21.us-west-2.compute.amazonaws.com
ssh -i mohan_ec2.pem ec2-user@54.185.39.21
```

Convert PEM into PPK

https://aws.amazon.com/premiumsupport/knowledge-center/convert-pem-file-into-ppk/

Popular EC2 users

```
Amazon Linux AMIs typically use:ec2-user
Debian: admin
RedHat: ec2-user
Ubuntu: ubuntu
```

EC2 instance normalization factor

1. nano 0.25
2. micro 0.5
3. small 1
4. medium 2
5. large 4
6. xlarge 8
7. 2xlarge 16
8. 3xlarge 24
9. 4xlarge 32
10. 6xlarge 48
11. 8xlarge 64
12. 9xlarge 72
13. 10xlarge 80
14. 12xlarge 96
15. 16xlarge 128
16. 18xlarge 144
17. 24xlarge 192
18. 32xlarge 256

Unofficial factor

- nano = 1
- micro = 2*nano
- small = 2*micro
- medium = 2*small
- large = 2*medium
- xlarge = 2*large

Common system status check failure

- AWS owned problem due to underlying host/power/network issue/corrup file-syste
- Better to relaunch ## AWS Storage offerings
- S3 - Object storage
- EBS - Block level storage (low latency service)
  - Persistent

AWS EFS - File storage (Kind of SAN or NAS file storage)

- Low latency - file level
- Multiple instance can access the storage
- Acts as network resource
- Mount point
- Can be scaled to peta-bytes in size
- Suppots High level throughput
- NFS 4.1 and 4.0
- Replicated across single region but multiple AZ (Regional boundry)

AWS EFS - Storage class

- Standard
- IA (Infrequent access) - cheaper - higher latency

AWS EFS LifeCycle Management

- Move to IA after 30/60/days
- If it accessed, it would automatically move from IA to standard
- Metadata would not be moved to IA

AWS EFS - Peformance modes

AWS EFS - General Purpose

- Max I/O
- Test before choosing between MaxIO or General Purpose

AWS EFS - Througput mode

- Bursting throughput mode
- Provisioned throughput mode

AWS EFS - EC2 instance - EFS - Mounting methods

- Linux NFS
- EFS Mount helper (new and recommended)
  - Log to /var/log/amazon/efs
  - Automatically connect to EFS at startup by editing /etc/fstab
  - ```
    sudo yum install -y amazon-efs-utils
    sudo mkdir efs
    sudo mount -t efs fs-0bc8309:/ efs
    sudo mount -t efs -o tls fs-0bc8309:/ efs
    ```
- EC2 instances connect to EFS using mount-target
  - Each mount target has ip-address
- EFS - has dns name
    ```

AWS EFS Overview

- Allow access to
  - elasticfilesystem:CreateFileSystem
  - elasticfilesystem:CreateMountTarget
- Allow access to mount
  - ec2:DescribeSubnet
  - ec2:CreateNetworkInterface
  - ec2:DescribeNetworkInterfaces

AWS EFS Security

- Data at rest - AWS KMS
- Data at transit - tls
- Uses stunnel (is an open source multi-platform application)

AWS EFS - Import existing data

- Aws DataSynch - securely move data from prop network to AWS
- Download dataysnc agent and should be installed as VMWare ESXi host to your site
- The AWS DataSync In-cloud Transfer Quick Start and Scheduler can be used for a wide variety of use cases to transfer NFS file data from one file system to another. Below are a few examples.
  - Migrate an NFS file system from Amazon EC2 to Amazon EFS within the same AWS region
  - Replicate an NFS file system from Amazon EC2 in one AWS region to an Amazon EFS file system in a different AWS region for disaster recovery.
  - Migrate an Amazon EFS file system from EFS standard (no lifecycle management) to an EFS file system with lifecycle management enabled. File systems with lifecycle management enabled will automatically move files to a lower-cost Infrequent Access storage class (EFS IA) based on a predefined lifecycle policy.
  - Migrate an Amazon EFS file system from one performance mode to another performance mode within the same AWS region.
  - Replicate an Amazon EFS file system from one AWS region to another Amazon EFS file system in a different AWS region for disaster recovery.

AWS EFS Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid" : "AllowFileSystemPermissions",
```

```
      "Effect": "Allow",
      "Action": [
        "elasticfilesystem:CreateFileSystem",
        "elasticfilesystem:CreateMountTarget"
      ],
      "Resource": "arn:aws:elasticfilesystem:us-west-2:account-id:file-system/*"
    },
    {
     "Sid" : "AllowEC2Permissions",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeSubnets",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS EFS Demo

- Create EC2 security group (EC2 -> SecurityGroup -> Create)
- Add inbound rule within VPC security-group
- Create EFS ## AWS Elasticache Overview
- What is cache
- What is ElastiCache Service? Why we need them?
- What are ElastiCache features?

AWS Elasticache

- WebService create,operate and scale in-memory datastore or cache in the cloud
- To make it more faster (works similar to L2 cache, for applications)
- Most of the applications reads more than write data
- We can horizontally scale web layer, but we can't do the same for DB layer
  - Cache can reduce the load on the DB layer, cache can horizontally scale
- Managed Service (no patching, update)

AWS Elasticache - When to use?

- Requires scalaing

- To improve performance
- Compatible with MemCached and Redis
- It is located part of Database (under management console)

AWS Elasicache usecase

```
We have a website that provides support information about the range of motorcycles that we s

One day, a fault is reported in a hose pipe, commonly used in motorcycle engines. Anyone who

However, we fear the worst, since last year a similar fault was announced and our website cr

Now when the web server requests the press release page, the content of that page is deliver
```

AWS recommends that you use an ElastiCache Redis (cluster mode disabled) cluster in which situation?

To handle highly read intensive application

- Runs within VPC
- Memcached vs Redis ### AWS ELB
- Elastic Load Balancing automatically distributes incoming application traffic across multiple targets,
    1. Amazon EC2 instances, containers, IP addresses, and Lambda functions.
- It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.
- Elastic Load Balancing offers three types of load balancers with feature of high availability, automatic scaling, and robust security.
    - Application LB
        * load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures.
        * Operating at the individual request level (Layer 7), Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request.
    - Network LB
        * Network Load Balancer is best suited for load balancing of Transmission Control Protocol (TCP), User Datagram Protocol (UDP) and Transport Layer Security (TLS) traffic
        * Low latency - millions of request per second
    - Classic LB
        * Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances

        * Operates at both the request level and connection level.
        * Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

ELB Components

- Listeners
  - Process that checks for connection request+protocol+port
  - Ports and Protocols set as conditions
  - One or more listener
  - Each listner has one or more rules
- Target groups
  - Target resources
- Rules
  - Network routing rule
    * If source is abc forward to target xyz
  - Configured to each listener
  - Has one or more condition
  - ```
    IF
      Source IP is 10.0.0.0/24
      HTTP GET request
    THEN
      Forward to TargetGroup1
    ```
- Health check
  - Specific protocol to check health of resource
- Facing - Internal vs Internet-facing
- ELB Nodes
  - Part of AZ
- Cross Zone LB
  - To distribute request acorss AZ between target

Server certificate (For encrypted request)

- To allow ALB to handle traffic over https it requires two things
  - Server certificate
  - Associated security policy
- You can choose/upload certificate from ACM/IAM (4 options)
  - ACM is recommended
- ACM - Certificate provided by AWS

- IAM - We can use to handle certificate provided by 3rd party

ALB

- Layer 7 (OSI model)
- Types of application - http, ftp, smtp, nfs
    ```

Steps for ELB

1. AWS Management console

2. EC2 console (under compute)
3. Load balancing section
4. Step-1) Setup target groups
    1. Target-group = name + type (instance/ip/lambda) + protocol + port + vpc + HealthCheck (protocol+port)
    2. Health-check advanced settings
    3. [healthy-thrshold + un-healthy-thrshold + timeout + interval + sucess codes]
5. Step-2) select instances for above target groups
6. Step-3) Create LB > ALB
    1. Name
    2. Scheme (internet vs internal)
    3. IP type (V4 vs v6)
    4. Configure Listeners (Port + protocol)
    5. Select AZ and VPC (subnet
7. Step-4) Create LB > ALB > Configure security settings
8. Step-5) Create LB > ALB > Configure security groups
9. Step-6) Create LB > ALB > Configure routing (same as target-group)
10. Step-7) Review and Create

Network load balancer (APSTNDP)

- Works at layer 4 (Transport layer)
- TCP, TLS vs UDP
- Low latency choice
- Cross -zone could be enabled/disabled
- Provisioned in AZ
- Algorithm chooses target based on TCP Sequence, the protocol, source port, source-ip, target-port, target ip
- Creation steps are similar to ELB but need to choose TCP/UDP instepad of protocol

Classic load balancer

- TCP, SSL, HTTP, HTTPS
- ALB should be prefered over CLB
- Classic network not supported for accounts created after 12-April-2013
- Works when network shared with other customers (without VPC)
- CLB (where ALB won't support)
    - Supports EC2 classic
    - Sticky session using application-generated cookies
    - TCP and ssl listener

AWS EC2 autoscaling

- Autoscale using metrics
  - CPU utilization above 80%
  - CPU utilization below 20%
- Scale-out vs Scale-in
- Avoids manual intervention to right scale cloud resources
- Cost reduction and Great customer satisfaction
- Scalalble and Flexible architecture
- Groups
  - Your EC2 instances are organized into groups so that they can be treated as a logical unit for the purposes of scaling and management.
  - When you create a group, you can specify its minimum, maximum, and desired number of EC2 instances.

AWS EC2 autoscaling - components

1. Launch configuration - Creaion of launch configuration (or Lauch template)
   1. Lauch template is advanced version of launch configuration
2. Autoscaling Group - Creation of Autoscaling Group
3. Your group uses a launch configuration as a template for its EC2 instances.
4. Launch configuration
   1. AMI
   2. Instance type
   3. If we need Spot instances
   4. if and when public ip-address
   5. if any user-data on first boot
   6. storage volume configuration
   7. security groups
   8. When you create a launch configuration, you can specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for your instances.

AWS EC2 autoscaling - launch template steps

1. AWS Management console

2. EC2 console (under compute)
3. Launch template > Create Launch Template
4. A launch template is similar to a launch configuration, in that it specifies instance configuration information.
5. Defining a launch template instead of a launch configuration allows you to have multiple versions of a template.
6. With versioning, you can create a subset of the full set of parameters and then reuse it to create other templates or template versions.
7. Step-1) Launch Template
   1. Create new tempalte

    2. name + description + source-template
    3. Launch template content
        1. AMI, Instance-Type, Key-pair + network type + security groups
        2. Network interace
            1. Device, Network-Interface+subnet+AutoAssignIP+Primary+Secondary+SecGroup
        3. Stoage volumes
        4. Tags
        5. Advanced details
        6. Spot
        7. IAM instance profile-role of ec2 instance
        8. Shutdown behavior
        9. User data (any user commands on boot)
8. Launch configuration
9. Similar to launch configuration
10. We can select IAM role (in LT - instance profile)

11. Ramdisk is possible
12. Key-pair

AWS EC2 autoscaling group

1. Launch configuration/template is mandatory for auto-scaling group
2. Desired capacity and other limitations
3. In which AZ to scale?
4. Steps
5. EC2 management console > AutoScaling Groups > Create ASG
6. Choose LC/LT and version
7. Fleet composition (LT vs combine purchase options)
8. Choose VPC subnet
9. Advance details 1. LB 1. Health check grace period 1. Instance protection 1. Service linked role

10. Scaling policy 1. Number of instances between 2 and 8 1. We can choose step of simple scaling policy 1. Increasing group size vs decreasing group size using alarm 1. Create alarm
    1. Avg CPU utilization is over 75% for consecutive period of 5 minutes (1 count) 1. Add notification type
    2. launch, terminate, fail to launch or fail to terminate
11. Create

CloudWatch

- By default, CloudWatch monitors EC2 instances approximately every 5 minutes. Detailed monitoring enables monitoring more often (each minute). Note: Detailed monitoring does have an associated cost.
- Cooldown period - The cooldown period helps you prevent your Auto

Scaling group from launching or terminating additional instances before the effects of previous activities are visible. You can configure the length of time based on your instance startup time or other application needs.

Combine ELB vs Autoscaling group

- Select the target-group in AutoScaling group to tie the ELB

Lab

- 257937829427/student/Ca1_3X1Kh4ha
-

```
#!/bin/bash
#Enable the epel-release
amazon-linux-extras install epel
#Install and start Apache web server
yum install -y httpd php
service httpd start
#Install CPU stress test tool
sudo yum install -y stress
```

chmod 700 ec2_challenge_1.pem ssh -i "ec2_challenge_1.pem" ec2_user@ec2-54-68-106-177.us-west-2.compute.amazonaws.com

To stress CPU

stress -c 2 -i 1 -m 1 –vm-bytes 128M -t 5m

###ELB Product comparisons
###What Is Amazon EC2 Auto Scaling? ### (https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-suspend-resume-processes.html)## Components of decoupled architecture in AWS (utilizing aws services for decoupled and/or event-driven environment.)

- Amazon Simple Queue Service
- Amazon Simple Notification Service
- Amazon Kinesis
- AWS Lambda

Decoupled Architecture

- Each component in a decoupled solution is effectively unaware of changes to other components due to the segregation of boundaries applied.
- Each service within a decoupled environment communicates with others using specific interfaces which remain constant

•

Event-Driven Architectures.

- Event-driven architectures closely relate and interact with decoupled architectures, however, services in an event-driven architecture are triggered by *events* that occur within the infrastructure.
- What is an event?
  - EC2 instance changing from 'running' to 'stopped' (Change of state)
  - An order has been placed on a website
  - An item has been moved from for sale to sold
  - A message that conveys something tangible
- Three components of event-driven archiecture
  - CFS (Consumer, Function, Supplier) - Spring Integration terms
  - PRC (Producer, router and consumer)
    - A producer is the element within the infrastructure that will push an event to the event router.
    - The event router then processes the event and takes the necessary action in pushing the outcome to the consumers.

AWS SQS

- Sending, storing and receiving messages at scale
- Serverless system, microservices and distributed architecture
- SQS types
  - Standard Queue
    - Unlimited throughput
      - Unlimited TPS
    - At-least once delivery
    - Best-effor ordering
  - FIFO Queue
    - High throughput
      - Default 300 TPS
      - Batching could work with 10 messages per operation (3000TPS with batching)
    - FIFO-deliver delivery
    - Exactly-once processing
  - Dead letter queue
    - Depends on above two types
    - If consumer couldn't process the message after maxiumum number of tries specified, queue will send the message to DLQ
- SQS Components
  - PQC - Producer, Queue and Consumer
  - Visibility timeout
  - Queue attributes
    - Message retention period

        * Delay deliverry
        * Maximum Receives (1-1000)
        * Maximum message size
- Visibility timeout
  - Visibility-timeout starts when a message is retrieved by a consumer
  - Consumers can process the message before visibility timeout expires
  - 30-seconds to 12-hours
  - If visibility timeout expires, the message will become available for other consumer to process the message

AWS SNS

- Publisher, Topic and Subscriber
- Policy of publisher and sender
  - Allow specific/anyone to subscribe/publish
  -
- AWS Deliver protocols
  - HTTP/HTTP(S)
  - Email
  - Email-JSON
  - SQS
  - Application
  - Lambda
  - SMS

AWS SNS and AWS SQS integration

- AWS SNS could be producer for SQS
- SQS Queue actions > Subscribe SNS Topic
  - Topic region and Topic ARN (ER Ireland/arn:aws:sns:eu-west-1:737739171055:EmailQueue)
- SQS Queue actions > Configure trigger for Lambda function
- SQS Topic actions > Subscribe to topic
  - Protocol > AWS Lambda > arn:aws:lambda:eu-west-1:737739171055:function:OthersTest
  - Version (or alias)

AWS SNS Policy

- The following example of a full policy grants the AWS account ID 1111-2222-3333 the ability to subscribe to notifications from a topic.

```
{
  "Statement": [{
    "Sid": "Statement1",
    "Effect": "Allow",
    "Principal": {
      "AWS": "111122223333"
```

```
    },
    "Action": ["sns:Subscribe"],
    "Resource": "arn:aws:sns:us-east-2:444455556666:MyTopic",
    "Condition": {
      "StringEquals": {
        "sns:Protocol": "https"
      }
    }
  }]
}
```

- AWS account 1111-2222-3333 can publish messages to the topic.

```
{
  "Statement": [{
    "Sid": "grant-1234-publish",
    "Effect": "Allow",
    "Principal": {
      "AWS": "111122223333"
    },
    "Action": ["sns:Publish"],
    "Resource": "arn:aws:sns:us-east-2:444455556666:MyTopic"
  }]
}
```

Amazon Kinesis (event-driven)

- Collect, Process and analyze realtime-streaming data
- Collect streams of data from sources and pipe to sinq
  - Application-logs/IOT data/Imagery data/Website clickstream
  - Database/Data lakes/Data warehouses
- Kinesis doesn't store data itself
- Versions of Kinesis
  - Amazon Kinesis Streams
    - Data streams (Amazon Kinesis Streams)
    - Video streams (Amazon Kinesis Streams)
  - Date Firehose
    - Doesn't require consumer application
    - We can configure transformer to transform data
    - Transform the data using readymade blueprint lambda function vs custom lambda function
    - Target delviery stream should be configured
      - Streaming data into S3/Redshift/Elastic Search/Splunk
  - Kinesis analytics
    - Standard SQL queries on streaming data

Amazon Kinesis Streams

- Amazon Kinesis Streams is ordered sequence of data records
- A record is the unit of data (in Kinesis stream)
- A record (Sequence number + Partition Key + Data Blob)
- Web service sending data can be considered as producers
- Consumer can be RedShift/S3
- Consumer application should use kinesis client library

Amazon Kinesis Analytics

- 3 steps requires
  - Create input stream
  - Create SQL processing logic
  - Create output stream
- Data processing real-time
- Output of stream of one-query can feed to second in-application stream

AWS Lambda

- It is like legacy CGI
- Serverless compute service
- Allow to run application code without having to manage EC2 instances
- Pay only for compute power
  - Subsecond metering
- Lambda functions should be configured to be triggered by events
  - Events from S3 buckets
- Event sources can be triggered to lambda functions
- Trigger - operation from an even source that causes the function to invoke
- Log streams - help to identify issues and troubleshoot issues with your Lambda functions
- Blueprint template provided by AWS Lambda functions

Reference

- SNS Access policy## AWS IAM - Features
- Shared access to AWS account
- Secure access to AWS resources for applications that run on Amazon EC2
- Groups, Users and Roles
- IAM policies
- Multi-factor authentication (MFA)
- Identity federation - (You can allow users who already have passwords elsewhere)
- PCI DSS Compliance - Payment Card Industry (PCI) Data Security Standard (DSS).

- AWS KMS

AWS IAM - Overview

- Identity - validate {user-id/password/key}
  - User-name/password
  - Multi-factor authentication (MFA)
  - Federated Access
- Authroization
  - Access - Authorization management {role, permission}
  - S3:CreateBucket
  - RDS:ReadOnly
  - EC2:FUllAccess
  - CloudTrail:FullAccess

AWS IAM - Components

- Root user
- IAM User - person or service who uses the IAM user to interact with AWS
- Roles - identity with permission policies that determine what the identity can and cannot do in AWS. However, a role does not have any credentials
- Groups - is a collection of IAM users (Groups can't be nested)
- Policies - A policy is an object in AWS that, when associated with an identity or resource, defines their permissions.
  - According policy certain resources can or can't be accessed
  - How complex a password policy

- Permissions = Policy + Identity (or resource like EC2)

IAM User

- Person or service account
- User can be created either for aws management console or programmatic access (access key id/secret access key)
- When user is created
  - Add user to group
  - Copy permissions from existing user
  - Attach existing policies directly (not good practice)
- Download security credentials via csv (you can't download second time)
- Access key id - 20 random upper case alpha-numeric
- Securet access key - 40 random upper case alpha-numeric
- Sample user ARN: arn:aws:iam::0123456789:user/Admin
- AWS CodeCommit is github equivalent, we can provide access while user is created!

AWS IAM Identity vs entity

IAM entity (user or role) IAM identity (user or role or group)

IAM Groups

- Used in authorization process
- Group = Policies + collections of users
- Groups are based on job role or specific requirement
- Group = GroupName + Set of policies that has access
- Default 100 groups, and user could be associated to 10 groups

IAM Role

- Users and resources can acquire temporary access to IAM permissions
- By attaching roles to aws-services, they act like users
- Roles can acquire policy
- In AWS, we can't attach policy to aws services
  - Attach role to services, and attach policy to roles
- Nowadays, EC2 roles can be dynamically changed (unlike early days we have to destroy to change roles).
- EC2 instance requiring access to Amazon S3 to Put and Get objects
  - Credentials could be stored on the EC2 instance itself (store access key-id and secret access key)
  - We can dynamically attach roles (instant)
- They can be considered like groups for services (not groups for users)
- Predefined IAM Service Role
  - Amazon EC2
  - Amazon Directory Service
  - Amazon Lambda
  - Amazon Redshift
  - Amazon Batch service role
- Predefined IAM Service Linked Role (can't modify permission)
  - Amazon Lex - Bots
  - Amazon Lex - Channels
- Cross-Account-Access (Truted-Trusting-Account)
  - When Truted account need to access resources of "Trusting Account"
  - Role is created in the trusting account
  - Trusted account will assume role of trusting account
- Role for Identify Provider Access
  - Grant access to web identiy providers
  - Grant access to web SSO SAML providers
  - Grant access to SAML Providers

AWS Policy (Like traditional granular permission)

- Example Policy: AmazonS3FullAccess, AmazonS3ReadOnly, AmazonEC2FullAccess
- arn:aws:iam::aws:policy/AmazonEC2ReadOnlyAccess

AWS IAM - User policies based on job functions

1. Administrator - User has full access and can delegate permissions to every service and resource in AWS.
2. Billing - user needs to view billing information, set up payments, and authorize payments. The user can monitor the costs accumulated for the entire AWS service.
3. Database Administrator
4. Data Scientist - user runs Hadoop jobs and queries. The user also accesses and analyzes information for data analytics and business intelligence.
5. Developer Power User - user performs application development tasks and can create and configure resources and services that support AWS aware application development.
6. Network Administrator
7. Security Auditor
8. Support User - user contacts AWS Support, creates support cases, and views the status of existing cases.
9. System Administrator - user sets up and maintains resources for development operations.
10. View-Only User

IAM Policies

- Policies are JSON format document
- IAM JSON Policy Elements Reference
- PolicyTypes
  - AWS Managed policies
  - Customer Managed policies
    - Copy aws manged policy/policy generator/Create your own policy
  - In-line policies
    - They are custom and not-reusable for others to use
    - They are added when resources created (user/ec2)
    - When we need to reduce the risk of permission being reused by any other entity
- Policies resolution conflict
  - By default, all access is denied
  - Explicit allow is required
  - Single Deny would overrule any number of Allow

```
{
  "Version": "2012-10-17",
```

```
  "Statement": {
    "Effect": "Allow",
    "Action": "s3:ListBucket",
    "Resource": "arn:aws:s3:::example_bucket"
  }
}

{
  "Version": "2012-10-17",
  "Id": "S3-Account-Permissions",
  "Statement": [{
    "Sid": "1",
    "Effect": "Allow",
    "Principal": {"AWS": ["arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:root"]},
    "Action": "s3:*",
    "Resource": [
      "arn:aws:s3:::mybucket",
      "arn:aws:s3:::mybucket/*"
    ]
  }]
}
```

IAM MFA

- Governance controle might force MFA
- Additional factor for authentication
- MFA Devices are allowed
  - Virtual MFA device
    - Any mobile application that supports open TOTP standard can be used
  - Hardware Key Fob/ Hardware Display Card
  - SMS MFA device
- Virtual MFA device
  - Example: Google Authenticator
    - Scan QR code and pass two authentication code for activating virutal MFA

IAM identify federation

- External identify providers could provide service equivalent of what IAM does
- Id providers allow user to access AWS resource access securely
- OpenId Connection Provider
- Microsoft Active Directory can grant access to your AWS resource
- Trust relationship between AWS account and IdP
  - Two types SAML-V2.0 and OpenID

        * OpenId providers includes Google, Facebook, Amazon
        * SAML (Security assertion markkup language)
            · MS-AD uses SAML
- Active directory authentication
    - Authenticated by MS-AD
    - SAML will be issued to user
    - User would send SAML to AWS STS
    - User would access S3 using STS
    - Security Token Service
- Steps to create create OpenId provider *Steps to create - IAM SAML Identity Providers

AWS IAM account settings

- Password policy (State 8 points)
    - Require atleast one uppercase letter
    - Require atleast one lowercase letter
    - Require atleast one number
    - Require atleast one non-alphanumeric
    - Allow user to change own password
    - Enable password expiry in 99 days
    - Prevent password reuse (number of password to remember)
    - Prevent password reuse (number of password to remember)
- Security Token service can be selectively activated for certain regions
- Credential report can be generated only once every 4 hours
- Credential report format
- AWS KMS
    - Managed service
    - Manages encryption keys
    - CMK - Customer master keys can be managed
    - Keys can be used to encrypt data

New user credential

```
csv User name,Password,Access key ID,Secret access key,Console
login link john,tCN5X&VugLQ7,AKIA5WBGSXIIEW4A4RE4,167paOvNFTHYNGGK6Uz4F/3FuLgw73moTzNEpUje,h
```

AWS KMS - overview

- KMS generates and provides a secure central repository of encryption keys
- Used in data encryption (changing data from readable to unreadable)
- Longer the key; the more robus is the encryption
- Easy to secure keys than data, and use stricter and different policy for keys
- S3 server side encryption may use AWS-KMS (SSE-KMS)
- Should be used only for data-at-rest
- AWS Administrators doesn't have access to KMS keys, all os updates

involved KMS requires two administrator approval.
- CloudTrails logs realted to KMS is key for investigation
- Region specific service
- KMS can be found under IAM (in management console)

AWS KMS - encryption to protect data

- Symmetric cryptography
  - Intercept is avoided since KMS is central repository for keys
  - Only decrypt keys are transmitted to decrypt the data
- Key management service
- How key encryption process works
- Different key types
- Manage Key policies
- Key management such as rotate, delete and reinstate keys

AWS KMS - terms

- Plaintext
- Ciphertext - encrypted, not human readable
- Key - string of characters
- Symetric key algorithm - AES/DES/Triple-DES/Blowfish
- Assymetric key algorithm - RSA, Diffie-Hellman, Digital Signature Alogorithm
- Symmetric crypto is fater than assymetric cryptography
- Do we need both private and public key to decrypt data?
- CMK - Customer master keys
- DEK - Data encryption key
- Envelope encryption
  - Envelope encryption allows store, transfer, and use encrypted data by encapsulating its data keys (DKs) in an envelope
- Key policies - who can use, access keys from KMS (json policy document).
- Grants - Resource based policy, another method to control use & access to AWS KMS (principle + kms:PutKeyPolicy)

AWS KMS - Components

- CMK - Main key type
  - Works with upto 4KB
  - CMK used in relation to DEK (Data-encryption key)
  - CMK used to generate DEK
  - AWS Managed CMK
    - For service, by service, created and used by AWS Service
    - Region specific
  - Customer Managed CMK
    - Rotation, governing access, key policy configuration

        * Enable/disable based on requirement
        * AWS service can be configured to make use of customer-managed-CMK
        * Protected by FIPS 140-2
- Three ways of gaining access (to CMK)
    - Key policies, IAM policy and Grants
- By default root-account gains access to CMK using "CMK-A" policy

AWS KMS - DEK

- Generated by CMK, DEK is immediately encrypted and plain-text DEK is deleted upon data encryption is completed
- Even if hacker has access to encrypted DEK, he can't use it on encrypted data
- Access to CMK is required to decrypt the DEK (even before decryption data)
- AWS KMS discourages encrypting/decrypting data directly with Customer Master Keys (CMKs).

How Amazon s3 uses KMS for SSE

- In diagram

Key policies

- Define key administrators and key users
- PAREC (Principal, Action, Resource, Effect (allow/deny) and Condition)
- Gain access to CMK
    - Key policy (is required for all CMK)
    - Key policies + IAM policy could grant acecss
    - Grants
- IAM Policy alone not sufficient to access CMK (Key policy should allow them to access)
- We must have following entry within the key policy allowing the root full KMS access to the CMK
- Below policy example shows the policy statement that allows access to the AWS account and thereby enables IAM policies.

```
{
  "Sid": "Enable IAM User Permissions",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
  "Action": "kms:*",
```

```
    "Resource": "*"
  }
```

- Root user of the account by default has full access to CMK
- Key administrators do not have access to use the CMK, but have access to update the associated key policy.
  - They can delete the key, they can update association and could give access to key access
- Key administrators actions
  - kms:Create, kms:Describe, kms:Enable, kms:List, *kms:Put*, kms:Update, kms:Revoke, kms:Disable, kms:Get*, kms:TagResource, kms:UntagResource
- Key user actions
  - kms:Encrypt, kms:Decrypt, kms:ReEncrypt, kms:GenerateDataKey*, kms:DescribeKey

Key Grant

- ALlow to delegate identity permission to another AWS principal (within AWS Account)
- Resource based method of access control to the CMK
- GrantId/GrantToken provided by user to another user
- Grant can be provided via AWS KMS API
  - No management console ui for Grants
- Action related to Grants (kms:CreateGrant, kms:ListGrant and kms:RevokeGrant)
- Grantee (receiver of Grant)
  - Some operations for Grantee
  - Decrypt, Encrypt, ReEncryptTo, ReEncryptFrom, GenerateDataKey, GenerateDataKeyWithoutPlaintext, CreateGrant, RetireGrant, DescribeKey
- GrantId and GrantToken are issued to Grantee
- Immediate use of GrantToken with some API would force AWS to replicate faster (workaround for eventual consistency)

AWS Cli

```
aws configure --profile new-user-id-bob
#should provide access-key-id and secret-access-key once prompted
aws kms encrypt --plaintext "This is sample plain text" --key-id alias/DemoKey --profile use
## ALice allowing bob to encryp using grant
aws kms create-grant --key-id arn:aws:kms:us-east-1:72384324391066:key/39439-3439493-3434343
aws kms encrypt --plaintext "This is sample plain text" --key-id alias/DemoKey --grant-token
```

```
## Update manual key-id
aws kms update-alias --alias-name BobKey --target-key-id 34939-3434-343-3343UUID
```

Steps for CMK policy (and for new user)

- IAM > Encryption keys > "Create Key" (or Select Key) > "Alias: value, Description: Value and Key-Material-Origin: KMS"
- Add tags
- Define key administrative permissions (choose users/roles) (key adminstrators are not users, they can't use)
- Define key Usage permissions (choose users/roles) (key users can't manage policies)

AWS Key management

- Rotate key
  - Why to rotate?
    - Reduce security breach, longer the usage probability of breaching is higher
    - Wider blast area when we use same keys
- Automatic rotate key - every 360
  - backing key is changed
  - Older backing key is retained for decrypt older data
  - Won't work for imported keys
  - Aws managed CMKs are rotated every 3 years (1095 days)
- If Old keys are compromised
  - Attacker still can breach the data that was encrypted using it
  - Rotation would help only newer data that used newer keys
- Manual update-alias for manual rotation
- What key material?
  - Backing key
  - Backing key/Key material is not generate when origin is "External"
- Key material can be imported
  - You need public-key (wrapping key)
  - RSAES_OAEP_SHA_256 or RSAES_OAEP_SHA_1 or RSAES_PKCS1_V1_5
  - You can't import plain-text key-material
  - Import-token is generate in this step
  - Use of the import token
    - The import token contains metadata to ensure that your key material is imported correctly.
    - When we upload encrypted key material to AWS KMS, we must upload the same import token that we downloaded.
  - Import the key-material (it should be in binary format to make use of wrapping key)
  - We can set expiration date

- Deleting CMK
  - Signficance impact to the entire applicaion
  - KMS enforces scheduled deletion process (7-30 days), not imeedite delete. We can cancel the "Cancel key deletion" within this period
  - Would be in pending-deletion (useless state)
  - We can find CloudTrail log for last time usage of this CMK
  - We can setup an alarm to find if anyone using deleted CMK
  - We have option to disable (instead of delete)

Reference

  - Importing key material step 4: Import the key material
  -

AWS Logging

- Amazon CludWatch Logging Agent
- Amazon CloudTrail logging
- Amazon CloudFront Access logs
- Amazon VPC Flow logs
- Monitoring AWS Amazon CloudTrail with CloudWatch Logging

Logging benefits

- Without logging finding resolution, safeguarding environment would be difficult or impossible
- Audit Control would require logging (date-time, ip-address, user-name, txn-id) for compliance certifications
- Thresholds monitoring would help to notify and corrective actions
- Performance and different service interactions can be deduced from logging

AWS CloudWatch Logs

- Install AmazonCloudWatchAgent using AWS SSM on your instances before collections logs (if it is not AWS AMI Linux)
- Unified CloudWatch agent allows to collect logs from EC2 and on-premise servers
- EC2 instance requires two roles to upload logs to CloudWatch
  - Role to install AWS SSM
  - Role to upload log to CLoudWath

AWS Roles

- AWS Service (identify - EC2 allows EC2 instances to call AWS services on your behalf)
- CloudWatchAgentAdminPolicy & AmazonEC2RoleforSSM

AWS Systems Manager

- Distribute software safely
- Common configuration files for any software can be distributed using SSM manager. SSM parameter store.
- Group of resources can be operated together (onpremise and cloud)
- Update patch/install application/generate cli command
- Groups 100s of resource types into business-unit/environment/application
- What is installed on particular instance?
- Starte the systems agent for all EC2 instance

AWS Systems Manager Agent (SSM Agent)

- AWS Systems Manager Agent (SSM Agent) is Amazon software.
- Can be installed and configured on an EC2 instance, an on-premises server, or a virtual machine (VM).
- SSM Agent makes it possible for Systems Manager to update, manage, and configure these resources.
- The agent processes requests from the Systems Manager service in the AWS Cloud.

SSM Agent is preinstalled, by default, on the following Amazon Machine Images (AMIs):

- Amazon Linux
- Amazon Linux 2
- Ubuntu Server 16.04
- Ubuntu Server 18.04
- Amazon ECS-Optimized
- sudo yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/linux_amd64/amazon-ssm-agent.rpm (for other instances)

CloudWatch configuration wizard amazon-cloudwatch-agent-config-wizard

- sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-config-wizard
- Are you installing the agent on an Amazon EC2 instance or an on-premises server?
- Is the server running Linux or Windows Server?
- Do you want the agent to also send log files to CloudWatch Logs? If so, do you have an existing CloudWatch Logs agent configuration file? If yes, * the CloudWatch agent can use this file to determine the logs to collect from the server.

- If you're going to collect metrics from the server, do you want to monitor one of the default sets of metrics or customize the list of metrics that you collect?
- Do you want to collect custom metrics from your applications or services, using StatsD or collectd?
- Are you migrating from an existing SSM Agent?

Sample CloudWatch agent configuration file

```
json     {       "agent": {         "metrics_collection_interval":
10,          "logfile": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log"
},       "metrics": {         "namespace": "MyCustomNamespace",
"metrics_collected": {           "cpu": {             "resources":
[               "*"             ],             "measurement": [
{"name": "cpu_usage_idle", "rename": "CPU_USAGE_IDLE", "unit":
"Percent"},               {"name": "cpu_usage_nice", "unit":
"Percent"},               "cpu_usage_guest"             ],             "totalcpu":
false,             "metrics_collection_interval": 10,             "append_dimensions":
{               "customized_dimension_key_1": "customized_dimension_value_1",
"customized_dimension_key_2": "customized_dimension_value_2"
}           },           "disk": {             "resources": [
"/",               "/tmp"             ],             "measurement":
[               {"name": "free", "rename": "DISK_FREE", "unit":
"Gigabytes"},               "total",               "used"             ],
"ignore_file_system_types": [               "sysfs", "devtmpfs"
],             "metrics_collection_interval": 60,             "append_dimensions":
{               "customized_dimension_key_3": "customized_dimension_value_3",
"customized_dimension_key_4": "customized_dimension_value_4"
}           },           "diskio": {             "resources":
[               "*"             ],             "measurement": [
"reads",               "writes",               "read_time",
"write_time",               "io_time"             ],             "metrics_collection_interva
60           },           "swap": {             "measurement": [
"swap_used",               "swap_free",               "swap_used_percent"
]           },           "mem": {             "measurement": [
"mem_used",               "mem_cached",               "mem_total"
],             "metrics_collection_interval": 1           },
"net": {             "resources": [               "eth0"             ],
"measurement": [               "bytes_sent",               "bytes_recv",
"drop_in",               "drop_out"             ]           },
"netstat": {             "measurement": [               "tcp_established",
"tcp_syn_sent",               "tcp_close"             ],             "metrics_collection_int
60           },           "processes": {             "measurement":
[               "running",               "sleeping",               "dead"
]           }         },         "append_dimensions": {           "ImageId":
"${aws:ImageId}",           "InstanceId": "${aws:InstanceId}",
```

```
"InstanceType": "${aws:InstanceType}",           "AutoScalingGroupName":
"${aws:AutoScalingGroupName}"         },         "aggregation_dimensions"
: [["ImageId"], ["InstanceId", "InstanceType"], ["d1"],[]],
"force_flush_interval" : 30       },       "logs": {         "logs_collected":
{           "files": {             "collect_list": [               {
"file_path": "/opt/aws/amazon-cloudwatch-agent/logs/amazon-cloudwatch-agent.log",
"log_group_name": "amazon-cloudwatch-agent.log",                 "log_stream_name":
"amazon-cloudwatch-agent.log",                  "timezone": "UTC"
},               {                 "file_path": "/opt/aws/amazon-cloudwatch-agent/logs/test.
"log_group_name": "test.log",                 "log_stream_name":
"test.log",                 "timezone": "Local"               }
]           }         },         "log_stream_name": "my_log_stream_name",
"force_flush_interval" : 15       }     } ## AWS RDS Mulit-AZ
```
deployments

- Amazon RDS Multi-AZ deployments provide enhanced availability and durability for RDS database (DB) instances
- Amazon RDS automatically creates a primary DB Instance and *synchronously replicates* the data to a *standby instance* in a different Availability Zone (AZ).
- Each AZ runs on its own physically distinct, independent infrastructure, and is engineered to be highly reliable.
- In case of an infrastructure failure, Amazon RDS performs an automatic failover to the standby (or to a read replica in the case of Amazon Aurora)
- Amazon RDS read replicas can be set up with their own standby instances in a different AZ.

Multi-AZ deployments, multi-region deployments, and read replicas

- Comparision

Multi-AZ deployments

- Main purpose is high availability
- Non-Aurora: synchronous replication; Aurora: asynchronous replication
- Non-Aurora: only the primary instance is active; Aurora: all instances are active
- Non-Aurora: database engine version upgrades happen on primary; Aurora: all instances are updated together
- Automatic failover to standby (non-Aurora) or read replica (Aurora) when a problem is detected

Multi-AZ deployments Failover mechnism

- Secondary instance would take over from primary instance

- It changes based on databse engine type
- For MS-SQL-server, it uses mirroring supported by MS-SQL-Server
- 60-120 seconds for DNS fail-over to secondary instance
- RDS-EVENT-0025 (Failure completion event)

Multi-Region deployments

- Main purpose is DR and local performance
- Asynchronous replication
- All regions are accessible and can be used for reads
- Automated backups can be taken in each region
- Non-Aurora: database engine version upgrade is independent in each region; Aurora: all instances are updated together
- Aurora allows promotion of a secondary region to be the master

Read-Replica

- Retention value of backups needs to be set a value >= 1
- It is neither primary/secondary instance, initally cloned from snapshot of one of this instance.
- It is read-only database, not for resilency
- It could be from secondary-db-instance
- They are available for MySQL, MariaDB, PostgreSQL, Oracle, and SQL Server
- There can be multiple read-replica
- Read replica can be promoted as primary if there is an event

Read-Replica - MySQL (MariaDB)

- Should be MySQL5.6 or more
- InnoDB can be used for RR
- Nested read-replica upto 4 layers are possible
- Read ReplicaLag can be used to find the lag (ideally zero is prefered)

Read-Replica - PostGresSQL

- Should be 9.3.5 or higher
- native PgSQL streaming is used for replication and read-replica
- write-ahead log is asynchrnously replicated between master and read-replicas
- Dedicated DB role is their to manage replication
- Multi-AZ read replica can be easily created for PgSQL## Multi-tier (tier = layer)
- Presentation + Logic + Data -> 3 tier architecture

- Every tier can be highly available in multiple AZ
- Each tier should be in different subnet to make it secure and scalable
- In AWS, we can increase the resources in anyone tier without scaling other tiers

Private Subnet vs Public Subnet

- A subnet is public if it has an internet gateway, and a route to that internet gateway
- A subnet is private if it doesn't have route to internet gateway
- Public subnet has a default route table enable us to allow inbound and outbound traffic
- Private subnet need a route table to direct traffic flow within VPC
- Always make sure your subnet has spare so it could be scaled for future use
- Always makes sure we leave spare capacity for additional subnets.

How to scale multi-tier

- Use network load balancer to front the presentation tier
- Use application load balancer between web-tier and application-tier

Scaling and securing web tier

- AWS WAF - Aws web application firewall solution can be integrated with cloudFront
- AWS Shield - AWS Shield could be used to protect from DDoS attacks
- AWS VPC Flow logs - can be used to audit/monitor VPC traffic

Autoscaling default termination policy

- Auto scaling selects the availability zone with two instances, and terminates the instance launched from the oldest launch configuration.
- If the instances were launched from the same launch configuration, then auto scaling selects the instance that is closest to the next billing hour and terminates that.
- This helps you maximize the use of your EC2 instances, while minimizing the number of hours you are billed for Amazon EC2 usage.

Serverless architecture patterns

- Aws Lambda + AWS API Gateway
- What is avoided?
  - No need to provision EC2 instances
  - No need for autoscaling groups, no autoscaling rules
  - No need to install code interpretters

  - No patching OS'es
  - No need to define AZ
- What may be requried?
  - Some definition of VPC might require
  - Some subnet configuratio may be required (based on design)

- We can configure ENI for database kinds of resources
  - So they are accessible only lambda functions (not internet accessible)
  - Lambda function can access them only via private subnet or private ENI
- AWS Serverless multi-tier architecture

- Logic layer can use AWS API Gateway
- We can improve API performance via caching and content delivery which immediately means that we don't need to create, manage and pay for Elastic load balancers between our tiers.
- One lambda for one api (/tickets, /weddings, /info)
- Each lambda could have IAM role
- Amazon ElasticCache can be configured with dynamo-db table, if cache miss, it could retrieve from DynamoDb table

MicroService architecture patterns

- MicroService can realize significant benefits when used with serverless resources.
- Each of the application components are completely decoupled and independently deployed and operated.
- API Gateway manages their API, and the functions subsequently are executed by AWS Lambda (those two are enough to build microservice architecture)

MicroService architecture patterns benefits

- The serverless microservice pattern lowers that bar to creation of each subsequent microservice.
- Optimizing server utilization is not so much of an issue with this pattern.
- An API gateway provides programmatic-generated client SDKs, which can make it even easier to integrate with other services and other runtimes.

Reference

- AWS Serverless multi-tier architecture # Managing network infrastructure and optimizing networking connectivity within an AWS environment

Networking components

- Elastic IP Addresses (EIPs)

- Elastic Network Interfaces (ENIs)
- EC2 Enhanced Networking with the Elastic Network Adapter (ENA)
- VPC Endpoints
- AWS Global Accelerator

Elastic IP addresses

- Private IP addresses
- Public IP addresses (EIP)
  - Pooled public IP address
  - IP address will remain with instance until it is stopped or terminated
  - can't convert an existing pooled public IP address to an EIP
- Elastic IP Addresses
  - Persistent IPv4 address that attached to aws account
  - Can attach an EIP address to an instance or an Elastic Network Interface
  - Detached EIP will still incur cost (unless returned to AWS)

Elastic Network Interface (ENI/Eth0)

- Why we need?
  - We might need secondary interface for management network
  - We might need more than one interface for single instance
- Should be attached to a subnet

- Logical virtual network interface card inside cloud
  - Create, configure, and attach to your EC2 instances
  - We can attach private IP address or an elastic IP address or it's MAC address.
- AWS VPC Flowlogs can be used to capture the trafic of ENI/PNI (logs)
- PNI - Primary network interface (default one attached to EC2)

How to craete enhanced networking features to reach speeds of up to 100 Gbps?

- Enhanced Elastic Network Adapter (ENA)
  - No additional cost involved
- Enhanced networking offers higher bandwidth with increased packet per second (PPS) performance
- Prerequisite
  - Not all aws instance supports ENA
  - Kernel version >= 2.6.2 or >= 3.2
- Instances with latest version of the Amazon Linux AMI will have enhanced networking enabled by default
- How to check if ENA is supported
  - `aws ec2 describe-instances --instance-ids instance_id --query "Reservations[]`

VPC Endpoints

- Allow to privately access AWS services using the AWS internal network (without public DNS network)
- Doesn't require following gateway/instances
  - Internet Gateway, NAT Gateway, a Virtual Private Network or a Direct Connect connection.
- Interface Endpoints vs Gateway Endpoints
- Interface Endpoints
  - Interface Endpoints are essentially ENIs
  - PrivateLink
    - PrivatLink tartget would be Interface enpoints (that could in turn allow other AWS services accessble)
    - Allows a private and secure connection between VPCs, AWS services, and on-premises applications, via the AWS internal network.
    - Private DNS name that resolved to the private IP address of the interface endpoint and will route through the internal AWS network instead of the internet
- Gateway Endpoint
  - RouteTable would be target enpoint for Gateway
  - Amazon S3 and DynamoDB - are supported
  - Route that was selected will have automatic entry for Gateway endpoint
  - Entry will have prefix related to the vpc-endpoint-id

Aws Global Accelarator

- Enable client TCP/UDP connection quickly communicate with AWS Global infrastructure
- Intelligently routes customer traffic (uses Edge locations)
- Configure global accelarators
  - Requires 2 ip-address (that sits behind load-balancers)
  - Configure listeners for TCP/UDP - port + protocol-TCP/UDP
    - Can select client affinity
  - Configure end-point groups
    - Region + Traffic-Dial
    - Configure health check
  - Add endpoints (with weight info)
    - EC2 instance
    - ELB
    - EIP
  - Global Accelarator - will have output DNS names to be used globally

AWS Organization - Overview

- Multiple account needs to be managed
- Multipl account advantage
  - Cost optimzations
  - Security Groupings
  - Management of controls and workloads
  - Resource groupings
  - Helping to define business units
- Multipl account const
  - Security risk
  - Operation issues
- AWS Orgnization can allow to centrallay manage

AWS Organization - Foundations

- Root
  - Organizational Unit (Marketting)
    - Accounts (Marketting - 12 digit number)
      - Service control policies
  - Organizational Unit (sales)
    - Accounts (sales - 12 digit number)
      - Service control policies
  - Organizational Unit (HR)
    - Organizational Unit (HR-APAC)
      - Organizational Unit (HR-India)
  - OU - can be nested at 5 levels.

AWS SCP - Service Control policies

- Permission that could be applied to root/ou/account
- AWS has master account (that manages other accounts)

AWS Organization

- provides Single Payer and centralized cost tracking
- Lets you create and invite accounts
- Allows you to apply policy-based controls
- Helps you simplify organization wide management of AWS Services

AWS Organization - Management

- Every organization has master account
- Don't use master account to create service such as EC2
- Use it only to manage root,ou and other accounts
- Consider merger & acquisition

  - Master account can create/invite/merge/remove an account from organization

AWS Organization - How

- AWS Management console > Services
- Management & Governance > AWS Organizations
- Create Organization
  - Master account would be created (by default) when orgnization is created
- Under Organization > Create Account (or) Invite Account
- Account-Id could be 12 digit number (or Email-ID)
- Invitation would send mail to account-id owner
- Sample organization id - o-6ixk4mv3io
- Under Orgnization Accounts
  - Can create new OU
  - Move accounts to OU

AWS SCP (Service Control Policies)

- using SCPs, we can specify Conditions, Resources, and NotAction to deny access across accounts in your organization or organizational unit.
- Feature of AWS Organizations rather than IAM
- Specify maximum permissions for affected entities
- SCPs can be specified to roots, organizational units (OUs), or accounts in your organization
- SCPs doesn't affect resource based policies
- SCPs affects principles managed by accounts in your organization

AWS SCP - Sample

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyChangesToAdminRole",
      "Effect": "Deny",
      "NotAction": [
        "iam:GetContextKeysForPrincipalPolicy",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:ListAttachedRolePolicies",
        "iam:ListInstanceProfilesForRole",
        "iam:ListRolePolicies",
        "iam:ListRoleTags"
      ],
```

```
      "Resource": [
        "arn:aws:iam::*:role/AdminRole"
      ],
      "Condition": {
        "StringNotLike": {
          "aws:PrincipalARN":"arn:aws:iam::*:role/AdminRole"
        }
      }
    }
  ]
}
```

AWS SCP - Policy inheritance

- policies can be attached to organization entities (organization root, organizational unit (OU), or account):
- Attaching a policy to the organization root, all OUs and accounts in the organization inherit that policy.
- Attaching a policy to a specific OU, accounts that are directly under that OU or any child OU inherit the policy.
- Attaching a policy to a specific account, it affects only that account.
- How policies affect the OUs and accounts that inherit them depends on the type of policy
- Eample: If root SCP is governed by 1,2,3 and 4. Child Org is SCP is applied 2,3,5 and6. Effectively Org will have 2 and 3 alone (Intersection of root and Org)

Reference

- How to use service control policies to set permission guardrails across accounts in your AWS Organization## What is the cardinality relationship between EC2 and EBS Volume

1. An EBS volume can be attached to only one EC2 instance in the same AZ
2. Multiple EBS volume can be attached to one EC2 instance in the same AZ

What are storage types

Block Storage, File storage, Object Storage

Where does EBS volumes are stored

S3, Snapshots are incremental

EBS Volumes

- They are only in one AZ

- EBS volume types are - SSD backed and HDD backed
- SSD Types - 3000 IOPS, 3 IO OPS upto 10000, 128MB upto 170GB
- Provisioned 4-16TB
- Can be encrypted (at rest and in-transit)
- Can be elastic
- Can restore volume on larger volume to increase the file-system size
- GP2, IO1, ST1, SC1
- Not suitable for temporary or multi-instance storage (S3 is suitable)
- Amazon EBS encryption uses AWS Key Management Service (AWS KMS) customer master keys when creating encrypted volumes and any snapshots created from your encrypted volumes.

S3

- There are 5 storage classes, Standard, OneZone-IA, Standard IA, Intelligent Tiering, Glacier, Glacier-Deep-Archive
- Amazon Glacier is an extremely low-cost storage service that provides secure and durable storage for data archiving and backup. To keep costs low, Amazon Glacier is optimized for data that is infrequently accessed and for which retrieval times of several hours are suitable. The standard retrieval option, which is the default option, takes 3-5 hours to complete. The other options are expedited, which downloads a small amount of data (250 MB maximum) in 5 minutes, and bulk, which downloads large amounts of data (petabytes) in 5-12 hours.
- Stanard - Properties
  - Versioning,
  - Server acess logging
  - Static website hosting
  - Object-level logging
  - Default Encryption
- Advanced - Properties
  - Object lock
  - Tags
  - Transfer acceleration
  - Events
  - Requester Pays
- Versioning - Not enabled by default
  - Once we delete, object would have delete marker on it, with GET operation ends up with 404. But older version exist if we have enabled versioning.
- Versioning - suspended
- Server access logging
  - Default disabled (Enable by specifying Target bucket and Target prefix)
  - Requires LOG DELIVERY GROUP write access for the target bucket ACL to log

  - Mangement console adds LOG DELIVERY GROUP by default when loggin is enabled.
  - Access log will only be delivered - SSE-S3 - should be enabled and KMS is not supported
- Static website hosting
  - Region specific end-point - HTTP only and requestor can't pay
  - Should add index document and error document
  - Redirect access to bucket
  - By default blocked to public
  - Should we need to add policy to grant access for public
  - Policy to access S3 bucket `json { "Version": "2012-10-17", "Statement": [ { "Sid": "PublicReadGetObject", "Effect": "Allow", "Principal": "*", "Action": [ "s3:GetObject" ], "Resource": [ "arn:aws:s3:::example.com/*" ] } ] }`
- Object logging - Related to CloudTrail
  - GetObject/DeleteObject/PutObject
  - Can select events that needs to be logged with timestamp
  - We can configure existing trail to log S3 access logging
- Encryption
  - None, AES-256, AWS-KMS
  - SSE-S3 - Server side encryption with S3 managed keys
  - Other encryption
    - SSE-C
    - CSE-KMS
    - CSE-C (customer manged keys)
- Object lock
  - WORM - Write once read many
  - Object lock withou version is not possible
  - We can't disable once we enabled
  - RetentionMode and Governance Mode
  - Retention Period can be enabled
  - Compliance mode would force retention period
  - Governance mode can be disabled that have specific IAM permission
  - Compliance mode cannot be disabled by any user
  - Legal hold - we can't delete even after retention period
- Tags
  - Environment - Production/UAT/DEV
  - S3 Cost Allocation tags
- Transfer Acceleration
  - Amazon CloudFront - is used to transfer acceleration
  - Additional Cost
  - Should be domain accessisible without DOT in bucket name
  - Does not support Get/Put/Delete,
  - Cross region copies using Put Object Copy

- Events
  - Selected events with Prefix or Suffix
  - SendTo components
    - Lambda, SNS or SQA
- Requester Pays
  - x-amz-requester-pays header is used

Instance storage

- Temporary and ephemeral
- Reboot retain data, but stopped and terminated would loose data
- It is in the price of instance

S3 encryption mechanism to secure data

- AWS S3 Encryption Infographic
- S3 Encrption mechanism
- Server side encrypted
  - SSE-S3 (S3 manged keys)
  - SSE-KMS (KMS manged keys)
  - SSE-C (Customer provided keys)
- Client side encrypted
  - CSE-KMS (KMS managed)
  - CSE-C (customer provided, and client side encrypte)

SSE S3 headers

- "s3:x-amz-server-side-encryption": "aws:kms"
- "s3:x-amz-server-side-encryption-aws-kms-key-id" : "arn:aws:kms:us-west-2:568157667383:key/86b02606-7d41-4a20-a694-d2b4d93cb522"
- New – Amazon S3 Server Side Encryption for Data at Rest
- AWS Key Management Service Cryptographic Details ## References
- https://cloudacademy.com/blog/how-to-encrypt-an-ebs-volume-the-new-amazon-ebs-encryption/
- Offers very high speed IO# Virtual Private Clouds

Components of VPC

1. Subnets
2. Route Tables
3. NAC - Network Access List
4. Nat instances and Nat Gateways
5. Bastion Hosts
6. VPN and Direct Connect

7. VPC Peering
8. VPC Transit Gateway
9. IGW - Internet gateways and Load Balancers

? What is netmask ? How many bits are supported for netmask

VPC

1. Each account is allowed to create 5 VPC per region
2. VPC = Name + CIDR block (IP-Address range)
3. VPC Quotas

NACL

- NACL works at network level (not instance level)
- NACL (pronounced like nuckle) are stateless
  - Any response traffic generated from a request will have to be explicitly allowed and configured in either the inbound or the outbound ruleset, depending on where the response is coming from.
  - In SecurityGroup, if we allow request traffic, response traffic is enabled (can't accept one-way)
- by default, NACL will allow all traffic, both inbound and outbound, so it's not very secure
- Inbound Rules + Outbound Rules
  - Rule-#, Type, Protocol, Port range, Source, Allow/Deny
  - PPTRASI (Self made abbreviation - Port,Protocol+Type+Allow-Deny,Source and ID)
  - Rule-ID with "*" is considered fall back rule
- A NACL can be applied to a number of subnets (like route tables)
- A single NACL can be associated to one subnet
- NACL evaluate the rules starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL.
  - The highest number that you can use for a rule is 32766.
  - It is recommended to creating rules in increments (for example, increments of 10 or 100) so that you can insert new rules where you need to later on.
- NACL document
- When you add or remove rules from a network ACL, the changes are automatically applied to the subnets it is associated with.

Security Group

- All the rules are evaluated before decisions are made (unlike NACL)
- Works at instance level (not at network level)
- We can specify allow rules, but not deny rules.

- Security-Group = Type/Protocol/Port-Range/Source (MySQL|HTTP/TCP/3306/10.0.1.0/24)

- By default, a security group includes an outbound rule that allows all outbound traffic. You can remove the rule and add outbound rules that allow specific outbound traffic only. If your security group has no outbound rules, no outbound traffic originating from your instance is allowed.
- Note that NACLs may take a bit longer to propagate, as opposed to security groups, which take effect almost immediately (1-2 seconds).

Subnets and CIDR block

- We can create public-subnet and private subnet
- Each subnet is ip-address block using CIDR
- Every subnet has a route-table
- For VPC, CIDR can rnage from */16 to* /28
  - Example: 10.0.0.0/16
  - Example: 10.0.1.0/24 - 10.0.2.0/24
- Isolated VPC can access internet by connecting with IGW
- VPN requires associated route table to access internet
- We can't attache more than one route table to a subnet
  - Same route table can be attached to multiple subnets
- Local Route - Undeletable default route helps to subnets talk ti other subnets within VPC
  - Eample Local Route - Destination: 10.0.0.16/16 -> Target: Local (Example default route)
- Route table entry in IGW
  - 0.0.0.0/0 -> igw-212823ab5g (Any host within VPC can contact IGW)
- Private subnet
  - If a subnet doesn't have a route to the Internet gateway, the subnet is known as a private subnet.
  - The private subnet routes internet traffic through the NAT instance.

Subnets, Region and Resillency

- For every subnet in AZ, it is better to have replica one in another AZ
- Web/App/DB Layer - Each can be in different sub-net

CIDR-Limitation on VPC

- First four IP address and last ip-address within subnet can't be used
- 10.0.1.0/24 - is a subnet
  - 10.0.1.0 -> Network IP address any subnet (Application can't use)
  - 10.0.1.1 -> AWS Routing reserved (Application can't use)
  - 10.0.1.2 -> AWS DNS reserved (Application can't use)
  - 10.0.1.2 -> AWS DNS reserved (Application can't use)
  - 10.0.1.3 -to- 10.0.1.254 -> No restrictions

  - 10.0.1.255 -> Broadcast IP address on any subnet (Application can't use)
  - 251 IP addresses out of 256 is accessible for our VPC

0.0.0.0/0 -> Every IP that is not explicitly configured in route table.

Largest possible VPC network

- Deploying the largest VPC possible results in over 65,000 IP addresses. ### 10.x.x.x address space
- We can use over 16,000,000 IP addresses

VPC - Nat Gateway

- Why NAT Gateway
  - If private subnet requires OS/security update, how to download patch from internet
  - Nat gateway accepts private-subnet initiated connection
  - Nat gateway rejects internet initiated connection into private subnet
- Nat gateway should sit in public gateway (not in private)
- Private subnet should have an entry 0.0.0.0/0 —> Nat Gateway to gain access to internet
  - 0.0.0.0/0 -> nat.0fabcdef678ghk
- Nat-gateway should be configured for resillent fallback VPC as well
- NAT instance is in the public subnet
  - It cannot be reached from the internet.
  - It has an inbound rule that only grants instances from the private security group (private instances) access.

Bastion Hosts

- Why we need Bastion hosts
  - EC2 instance in private subnet might need to be accessed from internet (internet initiated)
  - Remote engineer need to work
- Bastion Host
  - It acts as Jump server
  - Should be in public subnet
  - Should be very secure/hardened/Robust
  - Hackers attack them to gain access into private subnet
- Security group configuration
  - We need two security group
  - Security group to accepts connection to bastion hosts (Example: allow SSH incoming)

  - Security group in private subnet to accepts connection from bastion hosts (Example: allow SSH incoming connection from bastion host)
  - Bastion hosts receives connection via IGW
- Checklist
  - Don't store EC2 or other instances private key in bastion hosts
  - SSH Agent forwarding should setup on trust client machine who wants to gain access

VPN & Direct connect to VPC

- How to access VPC private subnet from Data Center using VPN
- VPN - 192.168.0.0/16 - Address space
- VPC - 10.0.0.0/16 - Address space
- VGW - Virtual Gateway should be attached to VPC
  - VGW - should configure about CGW details such as ip-address of customer gw
  - What type of routing table to be used @ CGW dynmic/static should be configured in VGW
- CGW - Customer gateway should be attached to DataCenter (Hw/SW)
- VPN Tunnel should be created between CGW & VGW
  - CGW alone can initiated connection to VGW (not the other way)
- By defaul VPN tunnel will drop connection if it is idle for 10 seconds or more, so monitoring should be in place using customer ping
- Routing table associated with private subnet should have entry
  - 192.168.0.0/16 -> vpw-02abcndGis3ds72k2 (to route data to Data-Center from VPC)
- Route Propagation supported for customer gateway that also support BGP
  - Static route table within VPC is not required if VPN has CGW that supports BGP (border gateway protocol)
  - There could be multiple CGW with VPC
- Security group should allow only certain protocol from CGW to Private subnet

Direct Connect

- Direct Connect is private connection and also get speeds from 1 through to 10 gigabits per second.
- AWS gas regional telecom Vendor support to add router for partner (client) and AWS router.
- All client data traffic would be routed via private connection within Vendor infrastructure
- Both AWS Public and Private VPN network is acceissble using Direct Connect
- Direct Connect = Customer data center + Direct Connect location (Customer Router + AWS router) + AWS infrastructure (specific region)

VPC peering

- A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately
- Instances in any aws VPC can communicate with each other as if they are within the same network.
- The owner of the requester VPC sends a request to the owner of the accepter VPC to create the VPC peering connection. The accepter VPC can be owned by you, or another AWS account
- VPC peering doesn't work transitively (it is 1-1)
  - (If VPC-A talks to VPC-B, and VPC-B talks to VPC-C, VPC-A can't talk to VPC-C)
- If two VPC has CIDR overlap, they can't estabilsh VPC-Peering
- Routing table betwen two VPC (10.0.0.0/16 and 172.31.0.0/16)
  - Routing table at VPC1
    - 10.0.0.0/16 -> local
    - 172.31.0.0/16 -> pcx.0fabcdef678ghk
  - Routing table at VPC2
    - 172.31.0.0/16 -> local
    - 10.0.0.0/16 -> pcx.9823fig122xA83
- Security group should allow for other VPC

AWS Transit Gateway

- A transit gateway is a network transit hub that can be used to interconnect VPC and on-premises networks.
- It is alternative to VPN Peering
- It reduces number of connection (using hub and spoke model)
- All routings are centrally managed
- Transit Gateway goes through this central hub, it allows you to centralize all your monitoring as well for your network traffic and connectivity all through the one dashboard
- Core concepts
  - Attachment - Attach new VPC to VPN

AWS VPC Lab (604274498499/student/Ca1_IhPhfoCK/172.31.0.0/16)

1. Creating a VPC subnet
2. Creating a VPC Internet Gateway
3. Connecting the Internet Gateway to the VPC Route Table
4. Creating an EC2 instance
5. Allocating and Associating an Elastic IP

AWS Lab-1

1. Create VPC with 10.0.0.0/16
2. Block size always should be between /16 and /28

3. VPC = name + cidr-block (10.0.0.0/16) + tenancy (default/dedicated)
    1. DHCP Options set + Main route table + Main Network ACL (default it creates)
    2. DHCP Options set enables DNS for instances that need to communicate over the VPC's Internet gateway
4. VPC ID - vpc-09598320b1f03592a
5. To create subnet (Name-tag+VPC+AZ+VPC CIDRs)
6. Choose subnet with lesser netmask for the VPC subnet - Subnet ID subnet-06b1516d2cab54272
7. We can create maxium of 5 subnet within a VPC
8. Internet Gateway - Internet gateway ID - igw-0d060a75a8b2a2a9e
9. To configure ping to instance, configure security group
10. All ICMP - IPv4 ICMP All 0.0.0.0/0 igw_to_instance
11. There is a route table for every VPC, that should be used to add routes

AWS Lab-2

1. Create public subnet, a private subnet, and a network address translation (NAT) instance in the public subnet.
2. Create public subnet with Bation host NAT to connect to private instance
3. Actual steps
    1. Create VPC - cloudacademy-labs with CIDR block - 10.0.0.0/16 (vpc-032ec65cd5ad76f2c)
    2. Create igw with name labs-gw
    3. Attach this IGW with the VPC - vpc-032ec65cd5ad76f2c
    4. Create Public subnet's IPv4 CIDR with name Public-A: - - 10.0.20.0/24
    5. Create new route table named: PublicRouteTable and attach it to Public-A subnet (PublicRouteTable)
    6. Route all address not belong to 10.0.0.0/16 to IGW
    7. Create Bastion host (in public VPC) (name: Bastion) 1. A bastion host is typically a host that sits inside your public subnet for the purposes of SSH (and/or RDP) access (Jump server) 1. Upon creation - assign public-ip and add the instance part of public subnet 1. In security group allow SSH connection from required IP address (find.MyIP)
    8. Create Private subnet's IPv4 CIDR with name Public-A: - 10.0.10.0/24 (subnet-0af2d53860d38dd8e)
        1. PrivateRouteTable should be created with IGW (IGW only stop gap, we should remove it)
        2. PrivateRouteTable should allow route to 0.0.0.0/0 via IGW and associate this route table to Private Subnet
    9. Create Private-NACL and attach to Private-Subnet
        1. Add Inbound Rule
            1. Allow SSH from 10.0.20.0/24
            2. Allow TCP port-range from 1024-65535 from 0.0.0.0/0

    2. Add Outbound Rule
        1. Rule-100 : Allow HTTP to 0.0.0.0/0
        2. Rule-200 : Allow HTTPS to 0.0.0.0/0
        3. Rule-300 : Allow TCP Port range 32768-61000 to 10.0.20.0/24
10. Launch Private EC2 instance in private subnet
    1. Select private subnet and disable public-ip
    2. Configure SecurityGroup (named SG-Provate)
        1. Allow TCP SSH Input from SG-Bastion (SSH only from public subnet of our own VPC)
11. Open SG-bastion security group (SG created during Bastion)
12. Allow Destination SSH to SG-Private
13. Use agent forwarding connect to Public instance
    1. From public instance connect to private instance
    2. sudo yum udpate
14. Create NAT instance to provide internet for private instances
    1. Choose public subnet
    2. Public-Ip should be enabled
    3. Configure SG-NAT security group
        1. Type of All Traffic and Source of SG-Private (allow all traffic from any instance using your private security group)
        2. Allow - SSH TCP 22 My IP
    4. Select the NAT instance, then click Actions > Networking > Change Source/Dest. Check. Verify that the attribute is disabled:
15. PrivateRoute Table should be updated
    1. Alllow 0.0.0.0/0 using above NAT instances created.

How to confirm is SSH forwarding was used

```
[ec2-user@ip-10-0-20-129 ~]$ env | grep SSH_AUTH_SOCK
SSH_AUTH_SOCK=/tmp/ssh-rfEo0aushe/agent.3598
```

ssh ec2-user@10.0.10.177

Errors

- No supported authentication methods available (server sent: public-key, gssapi-keyex, gssapi-with-mic)
    1. You may be using wrong user-id, ec2_user instead ec2-user caused me.

sg-02a0d700a289d3bc9 SG-Private sg-073ccdd96554911fc SG-bastion ec2_user@W34.216.151.145

1) Public-VPC - VPC ID-vpc-01691915afc11f067 - 10.0.0.0/19 1.1) Public-Subnet - Subnet ID/subnet-069b5b1d926b1a3f1 - 10.0.128.0/24
2) Private-VPC - VPC vpc-0d78ffac9dd928baa - 10.0.0.0/19 2.1) PrivateSubnet - Subnet ID/subnet-0e2a7d97838e62659 - 10.0.0.0/24

References

Public/Private Subnet security