AWS Solutions Architect -- Associate Certification Review

•••

Brent Tuggle, Chris Kuehn, Phil Winans, Tony Rimovsky

AWS Expectations

- One year of hands-on experience designing available, cost-efficient, fault-tolerant, and scalable distributed systems on AWS
- Hands-on experience using compute, networking, storage, and database AWS services
- Hands-on experience with AWS deployment and management services
- Ability to identify and define technical requirements for an AWS-based application
- Ability to identify which AWS services meet a given technical requirement
- Knowledge of recommended best practices for building secure and reliable applications on the AWS platform

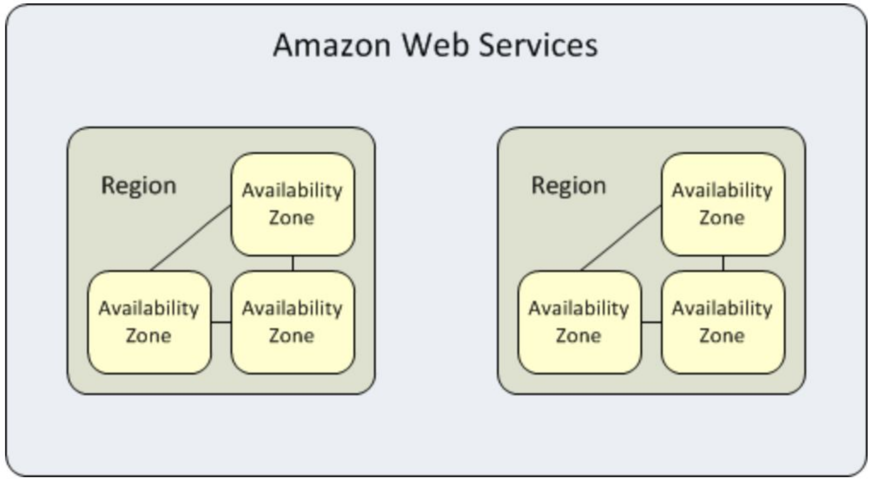
AWS Expectations

- An understanding of the basic architectural principles of building on the AWS cloud
- An understanding of the AWS global infrastructure
- An understanding of network technologies as they relate to AWS
- An understanding of security features and tools that AWS provides and how they relate to traditional services

Regions and Availability Zones

Region = Geographic location

Availability Zone (AZ) = Distinct infrastructure within a region (usually 3, at least 2), an AZ is more than just one building



Elastic Compute Cloud (EC2) Overview

EC2 is what most other AWS services are built on.

- Infrastructure-as-a-Service
- Virtual machines
  - No management provided by Amazon
- Very powerful and flexible
- Very easy to get started
- Requires more upkeep than Platform-as-a-Service options

EC2 Instance Families

T2: Lightweight, inexpensive. Good for systems that are idle most of the time.

M5: Multipurpose; balanced ratio of CPU & RAM

C5: Compute; extra CPU

R4: RAM; extra memory

G3, P3: GPU

X1, X1e: Very large, mainframe replacements (up to 128 vCPUs, 3,904 GiB RAM)

EC2 Pricing

Three ways to pay:

1. On Demand: Default, simple, guarantees resources for a fixed per second price
2. Reserved Instances (RIs): Commit to long-term usage of an instance type for a discount
    a. 1- or 3-year term, hard or impossible to cancel
    b. Full upfront (biggest discount), partial upfront, or monthly payments
    c. Variable levels of flexibility. Some RIs are convertible, some can be moved between AZ's
3. Spot Market: Real-time market price for AWS idle resources. Often save 60-80% on non-GPU instances.
    a. Relatively inflexible; spot instances terminate, rather than shutting down

Elastic Block Store (EBS)

Block storage; provisioned as volumes which can be attached to instances. Volumes appear within the OS as a SCSI or NVMe disk.

Performance tuning is largely a matter of picking the right volume type:

- General Purpose SSD (gp2): Moderate cost, good for short-duration bursts
- Provisioned IOPS SSD (io1): Best performance, highest cost
- Throughput Optimized HDD (st1): High throughput, poor seek times.
- Cold HDD (sc1): Lowest performance, lowest cost

It's possible to attach multiple volumes to a single instance. Combining multiple io1 volumes in a RAID-0 configuration can provide more than io1's max IOPs.

Instance Store

- Not EBS
- Ephemeral storage, locally installed on hypervisor
- Ideal for buffers, cache, scratch, or otherwise temporary data
- If the host fails, you LOSE data!
- You cannot take a snapshot of an instance store volume
- Limited to 1 volume per EC2 instance

NVMe Instance Stores available in C5d, I3, F1, and M5d instance families.

EBS Snapshots

Point-in-time backups of EBS volumes.

- Snapshots are stored in S3
- To create application consistent snapshots, stop instances first

Snapshots are created incrementally, for inherent de-duping.

- Initial snapshot
    - Copies every block that has been written to, not the whole volume
- Second snapshot is created after new writes to the disk
    - Copies only blocks that have changed since previous snapshot
    - Writes pointers to unchanged blocks from previous snapshot(s)

Amazon Machine Images (AMIs)

Used to launch new instances

Any snapshot can be turned into an AMI, so it’s common to install and configure software on a “gold” instance to snap into an AMI and launch repeatedly.

AMI’s are region-specific; can be copied to other regions as a new AMI

Frequently used in auto-scaling launch configurations

EC2 Roles

An EC2 instance can have an IAM role attached to it.

- More correctly called an instance profile, but “EC2 Role” is common
- Provides default credentials to any AWS API call originating from that instance

So if it’s running on EC2, your application doesn’t need credentials in its environment.

**Best practice dictates NEVER storing keys/credentials on an EC2 instance, use IAM roles instead!

EC2 Placement Groups

Placement groups allow you to run your EC2 instances on hypervisors which are near each other, or distinctly separated from each other.

- Useful for latency-sensitive HPC workloads.
- Only Certain Instances can go into a placement group

Clustered Placement Group - default reference in the exams

- Grouping of instances within a single AZ

Spread Placement Group - NEW

- Instances placed on distinct separate hardware
- Multi-AZ

Example EC2 user data script

Run as super-user as part of the system provisioning process.

```
#!/bin/bash

# Install httpd and update everything else.
yum -y install httpd
yum -y update

# Copy the website from S3
aws s3 cp s3://YOURBUCKETNAMEHERE/index.html /var/www/html/

# Start httpd at boot
chkconfig httpd on

# Odds are we got a new kernel in the update, so reboot to use it.
shutdown -r now
```

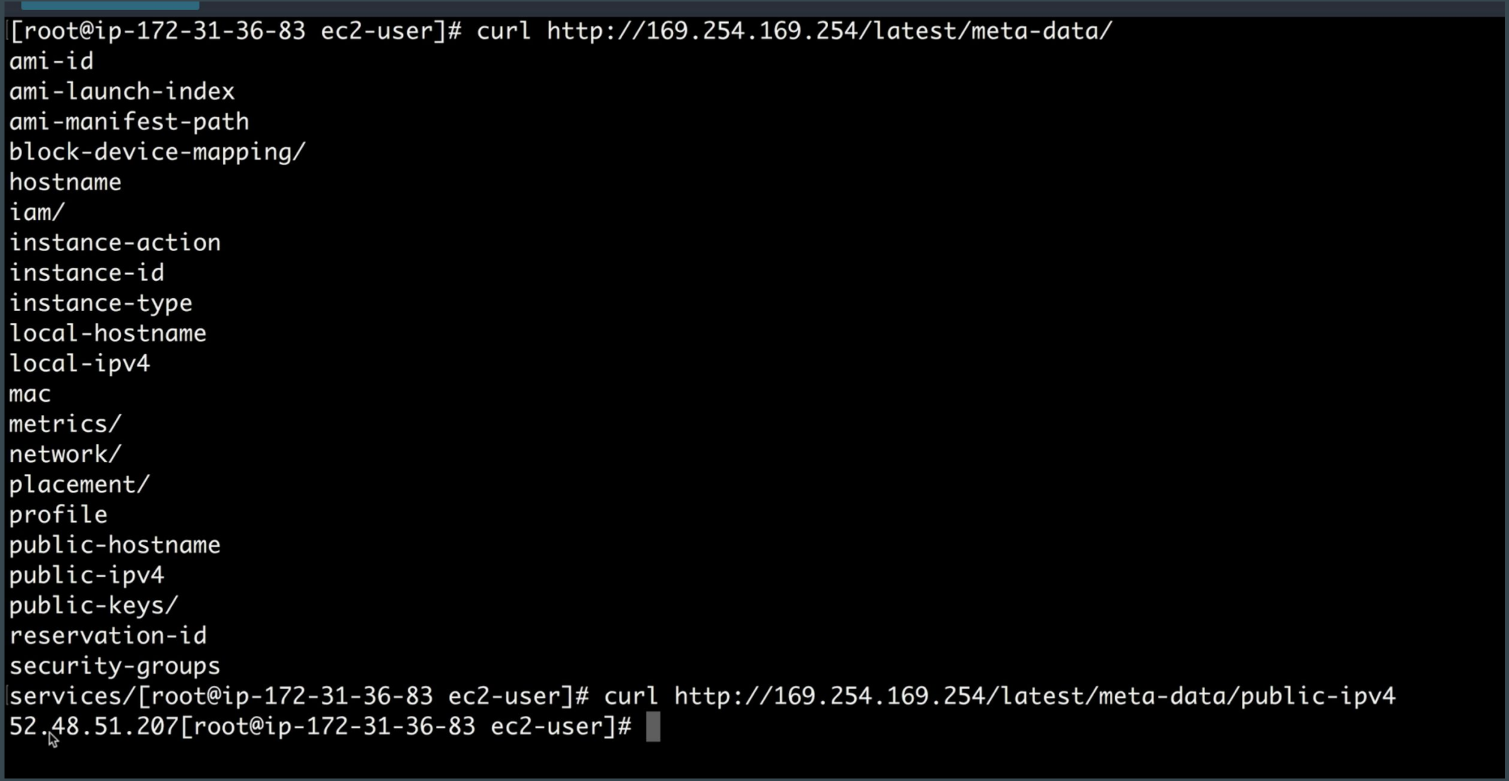
EC2 Instance Meta-data

```
curl http://169.254.169.254/latest/meta-data/
```

```
curl http://169.254.169.254/latest/user-data/
```

** Know these addresses!

EC2 Instance Meta-data



```
[root@ip-172-31-36-83 ec2-user]# curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/[root@ip-172-31-36-83 ec2-user]# curl http://169.254.169.254/latest/meta-data/public-ipv4
52.48.51.207[root@ip-172-31-36-83 ec2-user]#
```

EC2 Other

Termination Protection - turned off by default

- When off, EBS-backed instance, root volume deleted on termination

CloudWatch for performance monitoring

- Standard = 5 minutes, Detailed = 1 Minute
- Alarms trigger events - Autoscaling

EC2 Use Cases

Pick the right instance type and payment method for:

1. An always-on web server.
2. A deep learning model trainer.
3. An application development environment.

Elastic Load Balancer (ELB)

- Classic Load Balancers
  - Legacy elastic load balancers. Mix of basic networking and basic web app load balancing.
- Application Load Balancers
  - Best suited for balancing at the web/app tier. Application aware and can do things like send particular web requests to specific web servers
- Network Load Balancers
  - TCP only; best suited for simple load balancing where high performance is necessary
  - Can use a fixed set of IPs addresses

A 504 error on Classic LB means the app has timed out. Troubleshoot the app side.

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

Auto Scaling

- Auto Scaling group: a set of EC2 instances managed by an autoscaling policy.
  - Note: Reserved instances can't be part of of an autoscaling group
- Auto Scaling launch configurations are templates for EC2 instance configurations
  - Be careful about managing to account limits. Raise limits ahead of time if your scaling configuration might try.
- Auto Scaling plans: maintain, manual, scheduled, and dynamic
- Can "protect" instances from scale-in.
- Default termination
  - If multi-AZ, terminate from AZ with most instances and at least one not protected
  - Then terminate non-protected instance using oldest launch configuration
  - If multiples, then terminate node closest to the next billing hour
  - If multiples, then select at random but within the previous constraints.

Containers Overview

Elastic Container Services (ECS): Amazon-managed Docker service running on customer-specified EC2 instances. Docker containers are organized as “tasks” and managed through ECS, including auto-scaling.

- Recommend running through the “Break the Monolith” AWS tutorial.

Fargate: Docker service which removes management of EC2 capacity, higher cost than ECS.

Elastic Kubernetes Services (EKS): Managed Kubernetes environment, highly available (multi-AZ) management backplane.

S3/Glacier Overview

S3 provides extremely durable object storage, accessed via a RESTful API:

```
$ aws s3 cp s3://aws-illinois-edu/aws.css ./
download: s3://aws-illinois-edu/aws.css to ./aws.css
```

It’s easy to move data in and out of S3, but different from using a local disk.

Compared to local block storage, S3 tends to be:

- More scalable
- More widely accessible
- Less costly

S3 Storage Classes: Standard

- Immediate access to objects: low latency, high throughput
- Minimum object size is 0 bytes
- Maximum object size is 5 TB
  - Objects larger than 100 MB should use multipart uploads
- Each object is stored in an minimum of 3 distinct physical locations
- Designed for 99.999999999% data durability (eleven nines), 99.99% availability
  - SLA guarantees 99.9% availability
- Pricing is based on data stored and API request counts
- Supports versioning and encryption at rest

S3 Storage Classes: Infrequently Accessed (IA)

Same as S3 standard except:

- Designed for 99.9% availability; 99% SLA
- Minimum object size is 128kB
- Data storage price is lower
- IA introduces a per-gigabyte retrieval fee
- Minimum storage duration is 30 days

Recommended for data that may be needed at any time, but probably won’t be.

S3 Storage Classes: Glacier

- Offline storage with extremely low data storage fees
- Retrieval fee is based on first byte latency
- Minimum storage duration: 90 days

Glacier is best for long-term archival of data that you hope you’ll never need.

S3 Lifecycle Transitions

S3 allows you to configure rules that automatically migrate data between storage classes based on age. Rules are configured for all objects in a given path. For instance, you could:

1. Transition all objects to S3-IA after 7 days.
2. Migrate objects under /retiree-archives to Glacier after 60 days.
3. Delete all objects under /email-backups after 60 days.

S3 as a Web Server

S3 can be configured to make content available via HTTP and HTTPS. URL includes bucket name and region information, e.g.:

```
https://s3.us-east-2.amazonaws.com/aws-illinois-edu/index.html
```

Public sharing is off by default.

S3 presigned URLs give time-limited access to content.

```
$ aws s3 presign s3://aws-illinois-edu/index.html
https://s3.us-east-2.amazonaws.com/aws-illinois.edu/index.html?X-Amz-Algorithm=AWS4-HMAC-SH
A256&X-Amz-Credential=ASI...0611%2Fus-east-2%2Fs3%2Faws4_request&X-Amz-Date=20180611T161650
Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Security-Token=FQoDYX...&X-Amz-Signatur
e=12a2...e065
```

S3 Use Case

A campus researcher takes data sets from collaborators, processes them, and posts the results on an authenticated website.

How can we use S3 to give each collaborator secure access to their data (and only their data)?

Cloudfront Overview

Worldwide network of caching proxy servers.

Easy implementation: configure your existing web server as an origin server and update HTML to request static content from Cloudfront URLs:

```
<img src=”https://d1234567890123.cloudfront.net/big-banner.jpg” />
```

More advanced: Cache your entire site, selectively cache cookies and set custom TTLs for different website sections.

Databases Overview

Two categories of databases:

1. Relational (SQL) - RDS, Redshift, Athena
2. NoSQL - DynamoDB

Relational Database Service (RDS)

Recommended for ***OLTP*** (online ***transaction*** processing) and diverse data sets requiring JOINs across different tables.

Supported engines:

- Amazon Aurora
- MySQL
- MariaDB
- Microsoft SQL Server
- PostgreSQL
- Oracle

RDS Trivia

Backups: daily snapshot + replay logs

- You can initiate your own snapshots in addition to the automatic backups
- Restoration creates a new RDS instance

Multi-AZ: Easy high availability

- Costs a little more than double vs. a single instance
- Copies your data synchronously to another AZ
- Recovery time is suspiciously similar to EC2 instance boot time

Redshift

Redshift is intended for data warehousing

- OLAP processing (Online Analytical Processing)
- Optimized for ***column*** oriented operations
- Stored sequentially on disk
- Petabyte scale
- Massively Parallel Processing (MPP)
- Only available in single AZ

Amazon Athena

Allows you to issue SQL queries against the contents of a file in S3.

- CSV, TSV, JSON, application-specific formats
- AWS recommends Amazon Glue (extract/transform/load) to programmatically format data for Athena

Least costly relational database option.

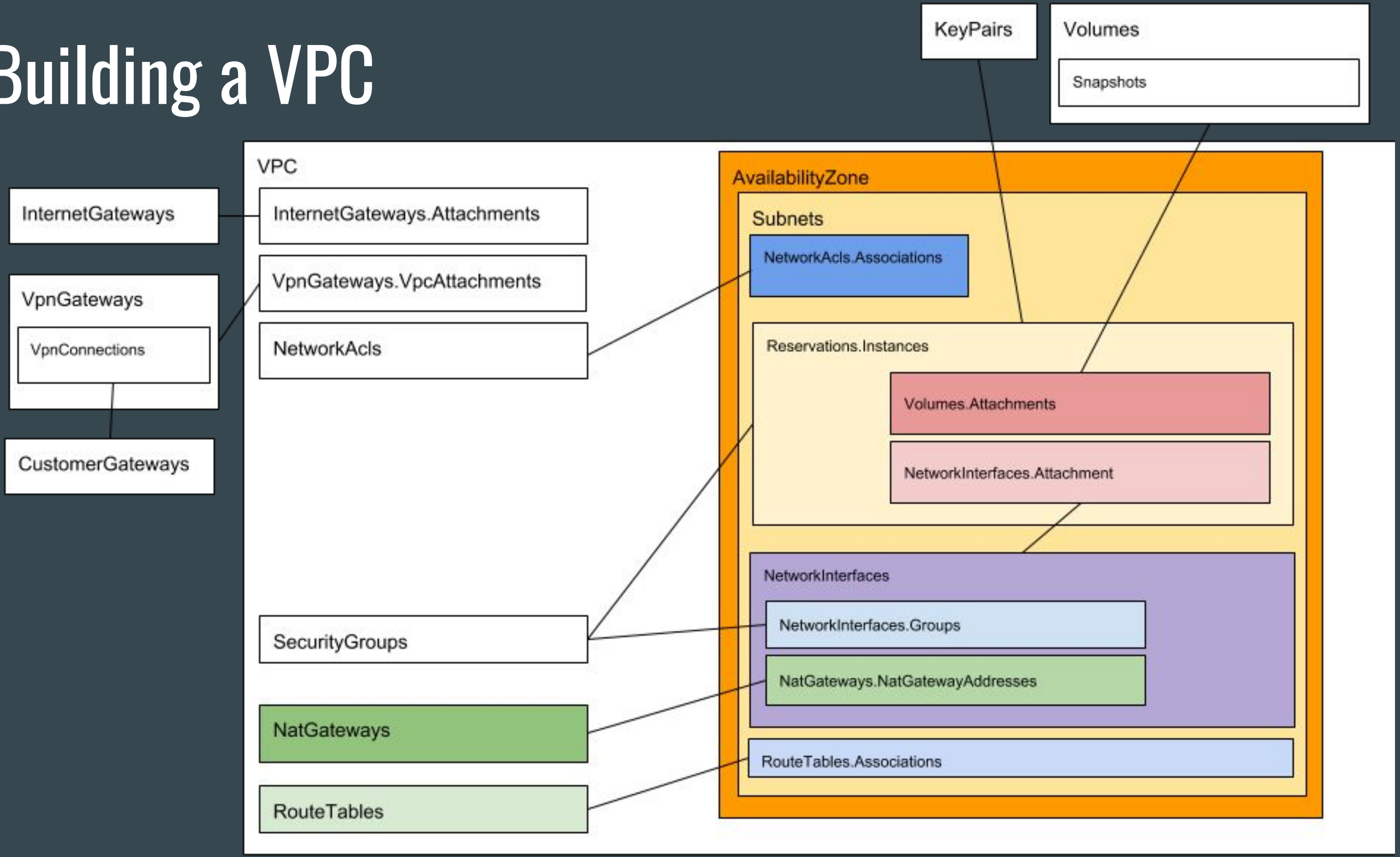
DynamoDB

- Amazon’s NoSQL database. Stores JSON objects
- Recommended for managing data structures with known identifiers
- Cannot do complex logical operations
  - DynamoDB allows you to scan the entire table or fetch a single record by its unique ID
- Serverless, so inexpensive
- Document and Key-Value data models
- All SSD storage with sub-millisecond latency, at ANY scale
- Always stored in 3 AZ’s
- Billed based on read & write capacity as well as data storage

VPC Overview

- Building a VPC
- Connectivity to a VPC
- Security
- VPC Flow Logs
- VPC Endpoints

Building a VPC



Components of VPCs: Subnets and Route Tables

- Subnets
  - Public and Private difference
  - Minimum size: /28
  - In one and only one Availability Zone
  - Must be associated with a network ACL (implicitly or explicitly)
  - Can only be associated with one network ACL at a time
  - First four IP addresses and the last IP address in each CIDR block are not available for you to use
    - e.g. in /24, .0 is network, .1 is router, .2 reserved for DNS, .3 is reserved, and .255 is broadcast
- Route Tables
  - Each VPC has Main route table by default
  - Can create custom route tables and associate subnets with them
- DHCP Options Set

Components of VPCs: Internet Gateways and NAT

- Internet Gateways
  - Only one associated with VPC
  - Egress Only Internet Gateway: used with IPV6
- NAT Gateways
  - Needs to run in a public subnet
  - Route table of private subnet points to NAT Gateway
  - Better bandwidth scaling than a NAT instance
- If creating your own NAT instance with EC2
  - Disable source/destination check on instance
  - Amount of traffic NAT instance can support related to instance size

VPC Connectivity

- VPN Gateway: VPN endpoint on AWS
- Customer Gateway: VPN endpoint on your network
- Understand what a VPC Peering does
- Understand when Direct Connect may be advantageous

VPC Security

- Understand difference between Security Groups and Network ACL
    - Security groups
        - ***Stateful***
        - Applied to instances/elastic network interfaces
        - Deny all inbound traffic by default, Allow all outbound traffic by default
    - Network ACLs
        - ***Stateless***
        - Applied to subnet
        - Default network ACL: Allow all inbound traffic by default, Allow all outbound traffic by default
        - Custom network ACL: by default denies all inbound and outbound traffic
        - Rules evaluated in order: lowest number evaluated first
        - Can be used to block IP addresses

VPC Flow Logs and VPC Endpoints

- VPC Flow Logs - used for seeing network traffic in VPC
  - Not all traffic will be logged: Amazon DNS, Windows activation traffic, DHCP, etc.
- VPC Endpoints
  - Allows VPC resources to talk to other AWS resources, such as S3, without using Internet gateway, NAT gateway, etc.
  - Two types: interface and gateway
  - Gateway endpoint:
    - S3 and DynamoDB
    - Configured as a route table entry
  - Interface endpoint:
    - ENI with private address that serves as entry point for a service
  - PrivateLink: allow your own application to be accessed as endpoint service

Route 53 - DNS

ALWAYS choose ALIAS over CNAME records

- Understand the difference between the two

Elastic Load Balancers do not have predefined public IPV4 or IPV6 address’

- The address can change, always use the alias

Know the different DNS routing policies and what they do

- Simple, Weighted, Latency, Failover, Geolocation

Identity and Access Management (IAM)

User: IAM construct which can login with a password or call AWS APIs with an access key ID and secret access key

- Often people, can also provide credentials for automated jobs

Group: Can define policy for all member users.

Role: Applies permissions to a session.

- Federated login (SAML: Shibboleth or ADFS)
- Cross-account trust
- Service roles (EC2 and Lambda are common)

IAM Policy Example

Policy: JSON document which defines API permissions:

```
{
    "Version": "2012-10-17",
    "Statement": [ {
        "Action": "*",
        "Effect": "Allow",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "aws:RequestedRegion": [ "us-east-1”, "us-east-2" ]
            }
        }
    } ]
}
```

IAM Policy Logic

Multiple policies can be applied to a user, role, or group

1. By default, everything is implicitly denied
2. Specific actions can be allowed with high granularity
3. An explicit denial trumps everything else

Simple Token Service (STS)

```
$ aws sts assume-role --role-arn arn:aws:iam::123456789012:role/ExampleRole
--role-session-name Example
{
    "Credentials": {
        "AccessKeyId": "ASIAJHQQ...",
        "SecretAccessKey": "GOyPrN...",
        "SessionToken": "FQoDYXdz...",
        "Expiration": "2018-06-15T16:32:17Z"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROA...:Example",
        "Arn": "arn:aws:sts::123456789012:assumed-role/ExampleRole/Example"
    }
}
```

Other Services

SNS, SQS, SWF, API Gateway -- Read and study the FAQs for these services

Simple Notification Services (SNS): ***push*** based, Delivers messages via HTTP, SMS, email, different costs based on protocol

Simple Queue Service (SQS): ***pull*** based (polls), guarantees the message will be processed at least once, Maximum 256kb in size

Simple Workflow Service (SWF): ***task*** oriented, tasks can only be assigned once and never be duplicated, tracks all tasks and events, workers (programs that interact) vs deciders (programs that controls coordination of tasks), can dispatch work to people.

Other Services

CloudFormation: Infrastructure-as-Code

- Define templates to create and configure “stacks” of AWS resources

OpsWorks: Hosted configuration management using Chef

- Configure your operating system

Kinesis: Ingests and processes data streams

GuardDuty: Machine-learning framework; analyzes API calls and VPC flow logs to determine what’s normal for your account and alerts on suspicious activity.

Other Services

ElastiCache: Managed memcached or Redis for in-memory key/value ***caching***

- Popular for session tracking

Lambda: Function-as-a-Service, extremely cost-effective. Triggered by other AWS services.

Elastic Beanstalk: Managed application stacks; deploy a .zip file of your code.

Tagging

Tags: Key-value pairs attached to a resource

```
{
    “Key”: “Name”,
    "ResourceId": "i-1234567890abcdef1",
    "ResourceType": "instance",
    “Value”: “BrentsInstance”
},
{
    “Key”: “Service”,
    "ResourceId": "i-1234567890abcdef1",
    "ResourceType": "instance",
    “Value”: “BrentsWebApp”
}
```

Resource groups help identify all items with same tag into a single list of resources.

Domain 1: Design Resilient Architectures

1. Choose reliable/resilient storage.
2. Determine how to design decoupling mechanisms using AWS services.
3. Determine how to design a multi-tier architecture solution.
4. Determine how to design high availability and/or fault tolerant architectures.

Domain 2: Define Performant Architectures

1. Choose performant storage and databases.
2. Apply caching to improve performance.
3. Design solutions for elasticity and scalability

Domain 3: Specify Secure Applications and Architectures

1. Determine how to secure application tiers.
2. Determine how to secure data.
3. Define the networking infrastructure for a single VPC application.

Domain 4: Design Cost-Optimized Architectures

1. Determine how to design cost-optimized storage.
2. Determine how to design cost-optimized compute.

Domain 5: Define Operationally-Excellent Architectures

1. Choose design features in solutions that enable operational excellence.

Other review material

https://aws.amazon.com/certification/certification-prep/

Whitepapers

- “Architecting for the Cloud: AWS Best Practices”
- “AWS Well-Architected webpage (various white papers)

FAQs

- EC2, S3, VPC, Route 53, RDS, SQS

Other review material

Videos:

- Scaling Up to Your First 10 Million Users -- Scaling, high availability, application architecture
- Another Day, Another Billion Flows -- VPC implementation

Tutorials:

- Break the Monolith

New exam

AWS has recently updated this exam to cover new services and architectural best practices. This preparation guide covers the updated version of the exam. Candidates have the option to take either the new exam or the previous version of the exam through August 12, 2018.

https://aws.amazon.com/certification/certified-solutions-architect-associate/