

# Exam Guide

AWS publishes its own exam guide, listing all the requirements and areas to focus on. The guide for [SAA-C03](#) lists in detail all the sub-topics you need to know for each.

## Example Questions

AWS also provides a small set of [example questions](#). It doesn't help you to get prepared for the exam from a domain perspective but previews the style of questions you'll face.

Besides working a lot with AWS in your day-to-day business, you can practice easily with a subscription to [CloudAcademy.com](#). They provide thousands of example questions that are very similar to the real questions you will face in the exams. You can also take them as a timed practice exam with a detailed result afterward, which will tell you if you have passed or not.

## Keyword Collection

In contrast to the Cloud Practitioner exam, the questions and answers for the Solutions Architect are very long and can contain a lot of unnecessary (or even confusing) information. Therefore it's good to get to know the important keywords with will most of the time immediately point to a specific solution.

- **Abstracted Services / Managed Services** - mostly, this will lead you to either DynamoDB (NoSQL), Aurora (relational data), S3 (data storage), or [Lambda](#) (computing resources). Benefits are reduced administration & operations and less effort to meet compliance requirement



- **Fault Tolerant / Disaster** - use multiple availability zones and some fail-over possibilities to a different region, e.g. active-active or active-passive configuration with the help of Route53.
- **Highly Available** - use multiple availability zones, e.g. for your EC2 Instances with an LB in front of multi-AZ database deployment.
- **Long Term Storage** - archiving data into a, mostly infrequently accessed, storage solution like Glacier, Glacier Deep Storage. S3 Intelligent Tiering or S3 Infrequent Access can also be considered if certain retrieval times have to be guaranteed.
- **Real-Time Processing** - often related to some Kinesis integration.
- **Scalable** - either make use of auto-scaling policies for instances or read replicas for your databases.
- **Elastic** - your environment is able to scale based on demand.

## Wrap-ups for different AWS Services and Components

In the test exams, I've encountered almost everything mentioned in the following paragraphs regularly, so everything here is really a must-know before taking the exam.

## Virtual Private Cloud (VPC) and Networking

VPCs can span multiple availability zones in a single region and can contain several public and private subnets. A public subnet contains a route to an internet gateway (which you need to set up). A private subnet has in general no internet access. If this is needed, you need to maintain a NAT Gateway to route traffic to those. If you need SSH access from the internet to a resource in a private

subnet you need to set up a bastion host on a public subnet and configure your Security Groups and Network Access Control Lists accordingly for forwarding traffic on port 22.


## Disaster Recovery Plans

- **Backup and Restore**—self-describing; has **highest** RTO and RPO but **lowest** cost
- **Pilot Light**—Storing critical systems as a template from which resources can be scaled out in the event of a disaster
- **Warm Standby**—a duplicate version of only your business-critical systems that are always running, in case you need to divert your workloads to them in the event of a disaster.
- **Multi-Site**—self-describing; **lowest** RTO and RPO but **highest** cost

## Recovery Objectives

- **Recovery Time Objective**—time needed to bring services back online after a major incident
- **Recovery Point Objective**—the data loss measured in time

## Route Tables

- rules how traffic can flow within your VPC
- always contains a destination and a target, e.g. `0.0.0.0/0` (CIDR Destination)  contains all IPv4 addresses of the subnet and points them to the Internet Gateway.

- attached to certain subnets
- there is a default route table (main route table) which will be associated with each newly created subnet as long as you don't attach one by yourself
- the main route table can't be deleted
- you can add, modify, and remove routes in this table
- one subnet can only have one route table
- the same route table can be attached to multiple subnets
- route tables can also be attached to your Virtual Private Gateway or Internet Gateway so you can define how traffic entering your VPC will be routed
- your VPC always has an implicit router to which your route tables will be attached to

## Virtual Private Gateway (VPC Gateway)

- needed if you want to connect your AWS VPC with an on-premise Network

## Network Access Control List (NACLs)

- operating on the **subnet level** and are **stateless**
- they can define **block & allow rules**
- by default allow traffic for all ports in both directions
- return traffic



# Security Groups (SGs)

- operating on the **instance level** and are **stateful**
- they only define **only allow rules**
- the default security group allows communication of components within the security group, **allow all outgoing traffic** and **block all incoming traffic**
- return traffic is implicitly allowed (opposite to NACLs)
- SGs can be attached or removed from EC2 instances at any time (state of machine does not need to be stopped or terminated)
- rules always need to specify CIDR ranges and never a single IP
- if you want to have a dedicated IP, e.g. `10.20.30.40` you also need to define it as a CIDR range covering only a single IP by its subnet mask: `10.20.30.40/32`

## CIDR (Classless Inter-Domain Routing)

- Certain range of IP-Addresses
- Important to know how they are built, because they are used in different touch points, e.g. SGs

## VPC Endpoints

- needed to access AWS Services which are not part of your VPC
- there are different types
- Gateway Endpoints



- **Interface Endpoint**—all other Services & are powered by AWS PrivateLink

## NAT Gateway & Instance

- needed to connect to the public internet from your private subnets
- there are two different types
- **NAT Instance**—managed by the user with no default auto-scaling
- **NAT Gateway**—AWS Managed Gateway, scales based on demand, fewer administrations required, and higher availability compared to the NAT Instance

## VPC Peering

- connecting different VPCs
- also possible to connect with VPCs of other accounts
- CIDR-ranges **should not overlap**
- connections are **not transitive**, therefore `A ← Peering → B ← Peering → C` means that there is **no connection** between A and C

## Transit Gateway

- hub for VPCs to merge multiple VPCs (could also include your on-premise VPC) into one giant VPC

## Elastic IP Addresses (EIPs)

- can be moved from one instance to another within multiple VPCs in the same region

## Services

The exam heavily focuses on core services, which are the basic building blocks for almost any application.


## EC2 (Elastic Compute 2)

Your basic compute resources in AWS and one of the first services that were released.

### Amazon Machine Image (AMI)

- predefined image, e.g. for Linux or Windows
- you can create your own AMI by launching it with an instance and then extending or modifying it and then saving it as a custom AMI
- contains one or more EBS snapshots or for instance-store-backed AMIs a template for the root volume (operating system, app server, applications, ...)
- contains launch permissions for the volumes to attach to the instance after launch
- can be either **EBS-backed** or **Instance Store-backed**

### Purchase Types


- **On-Demand**—no long-term commitment and a fixed price per second. Use s that can't be interrupted

- **Reserved** —long-term commitment (1 to 3 years). Use for high-uptime requirements at core-infrastructure
- **Spot** —auction-based bidding for unused EC2 resources—very cheap, but workload can be interrupted if your offer falls below market levels
- **Dedicated Host**—entire rack/server dedicated to your account, so no shared hardware—only for very high-security requirements because it's very expensive
- **Dedicated Instance**—physically isolated at the host level, but may share some hardware with other instances from your AWS account

## Elastic File System (EFS)

- Network Drive
- good for sharing data with multiple instances
- can also be attached to Lambda functions
- paying for storage in use & for data transferred
- different modes
- **General Purpose Performance Mode**—for low latency requirements
- **Max I/O Performance Mode**—for high IOPS requirements, e.g. big data or media processing; has higher latency than General Purpose Mode

## Elastic Block Storage (EBS)

- Virtual file system drive
- can't be used  only one per time (not a network drive)



- you can take snapshots of it
- if the EBS volume is the root volume, by default it will be deleted when the instance gets terminated
- non-root volumes will not be terminated after the instance gets terminated
- created in a single region
- for high availability/disaster recovery you need snapshots saved to S3
- Pricing is only for defined storage capacity, not per transferred data
- has several types
- **Cold HDD**—lowest-cost designed for less frequently accessed workloads
- SC1—up to 250 IOPS (250 MiB/s), 99.8–99.9% Durability
- **Throughput Optimised HDD**—low-cost designed for frequently accessed workloads
- ST1—up to 500 IOPS (500 MiB/s), 99.8–99.9% Durability
- **General Purpose SSD**—the balance of price and performance
- GP2/GP3—up to 16,000 IOPS (250–1,000 MiB/s), 99.8–99.9% Durability
- **Provisioned IOPS** (Input/Output Operations Per Second)—high performance for mission-critical, low-latency, or high-throughput workloads
- IO1/IO2—up to 64,000 IOPS (1,000 MiB/s), 99.8–99.9% Durability
- IO2 Block Express—up to 250,000 IOPS (1,000 MiB/s), 99.999% Durability

- Provisioned IOPS can be attached to multiple EC2 instances—other types do not support this
- most convenient way of creating backups is using the Data Lifecycle Manager to create automated, regular backups

## Instance Store

- ephemeral storage
- gets deleted at instance termination or hardware failure
- use only for a session or cached data
- very high IOPS

## Placement Groups

- Grouping your instance together on a certain level to achieve certain goals
- **Spread**—distribute instances across availability zones for high availability
- **Cluster**—place instances on the same rack for high network bandwidth
- **Partition**—multiple cluster groups, so you can have the best from both sides: high availability through spreading and high bandwidth through clustering

## Auto-Scaling Policies

- can't span over multiple regions
- different types of policies
- **Target Tracking Policies**



- Choose a scaling metric & a target value, e.g. CPU utilization
- EC2 Auto Scaling will take care of creating the CloudWatch alarms that trigger the policy to scale
- you can define warm-up times for which the target tracking won't be activated (if your instances start and CPU spikes to 100%, you don't want to scale because of this CPU utilization)
- **Step & Simple Scaling Policies**
- scale based on defined metrics
- if thresholds are breached for the defined number of periods, AutoScaling Policies will become active
- Step scaling policies allow you to define steps based on the size of the alarm breach
- e.g. Scale-out policy `[0,10%] -> 0, (10%, 20%] -> +10%, (20%, null] -> +30%` with desired 10 instances
- if the metric value changes from 50 (desired value) to 60, the scale-out policy will add 1 Instance (10% of your desired 10)
- if the metric changes afterward (after the cooldown of the Scaling Policy) to 70 in the metric value, another 3 Instances will be launched (30% of your desired 10)
- same can be defined for Scale-in policies
- **Scaling based on SQS**
- scale based on a metric of an SQS queue, e.g. `ApproximateNumberOfMessagesVisible`
- Scheduled Scaling
- scale your application instances based on a time schedule



# Route 53

A scalable domain name service that's fully managed by AWS.

## Failovers

- **Active-Active Configuration**
  - you want all of your resources to be available the majority of the time
  - When a resource becomes unavailable, Route 53 can detect that it's unhealthy and stop including it when responding to queries
- **Active-Passive Configuration**
  - when you want a primary resource or group of resources to be available the majority of the time and you want a secondary resource or group of resources to be on standby in case all the primary resources become unavailable
  - when responding to queries, Route 53 includes only the healthy primary resources
  - If all the primary resources are unhealthy, Route 53 begins to include only the healthy secondary resources in response to DNS queries

## KMS (Key Management Service)

A service to create and manage keys that can be used for encryption or signing your data.

- Encryption c



- can be shared evenly across the AWS Account boundary by assigning the targeted accounts as users of the master encryption key
- is natively integrated with other services like SQS, S3, or DynamoDB to easily encrypt data

## S3 (Simple Storage Service) & Glacier

Durable object storage that is easy to use and almost doesn't have any limitations.

- S3 **asynchronously replicates** your data to all availability zones of your bucket region
- has **no cap** for the number of files within a bucket
- **Server Side Encryption (SSE)**
- **SSE-S3** (Server Side Encryption managed by S3)—S3 manages the data and the encryption keys
- **SSE-C** (Server Side Encryption managed by the Customer)—you're responsible for your encryption keys
- you can use different encryption keys for different versions of a file in an s3 bucket
- Amazon recommends regular rotation of keys by the customer itself
- **SSE-KMS** (Server Side Encryption managed by AWS, KMS, and the customer)—AWS manages the data key but you're responsible for the customer master key kept in KMS (Key Management Service)

**Access Restriction**



- based on Geo-Location
- with Signed URLs—needs expiration date & time of the URL (as well as AWS Security Credentials / Bucket Name containing the objects) for creation
- restrict access to certain VPCs
- S3 stores objects in **buckets**, Glacier in **vaults**
- **Object Locks: Retention Modes**
- **Governance Mode**
  - users can't override or delete objects unless they have specific required rights
  - you protect objects against being deleted by most users, but you can still grant some users permission to alter the retention settings or delete the object if necessary
- **Compliance Mode**
  - a protected object version can't be overwritten or deleted by any user, including the root user in your AWS account
  - its retention mode can't be changed, and its retention period can't be shortened

## Storage Gateway

- gives on-premise services access to unlimited cloud storage
- different Storage types
- **Stored**—use S3 to backup data, but store locally → you are limited in the amount of space you can allocate
- **Cached**—stores all data on-premise and access locally only for caching

- has different Gateway Types
- **File Gateway**—stores files as objects in S3, using NFS and SMB file protocols
- **Tape Gateway**—virtual tape library for storing backups, using iSCSI protocol
- **Volume Gateway**—using EBS volumes, using the iSCSI protocol; data written to those volumes can be backed up asynchronously to EBS snapshots

## Relation Database Service (RDS)

- There is support for Multi-AZ Deployments
- usage-based pricing
- synchronously replicates data between multiple Availability Zones
- horizontal, and vertical scaling
- offers higher availability, failover support
- only minimal downtime when scaling up
- you can select your engine of choice
- **MariaDB**
- **SQL Server** —does not support read replicas in a separate region
- **MySQL**
- **Amazon Aurora** —MySQL and PostgreSQL compatible, fully managed (serverless solution) & supports read replicas, point-in-time recovery, continuous backups to S3 & replication across AZs
- **Oracle**



# Lambda

The core building block for serverless applications.

- integrates natively with different services like SQS or SNS
- can run on x86 or ARM/Graviton2 architectures
- compute-resources (vCPU) are sized based on memory settings
- dynamic configuration with environment variables
- AWS abstracts away all your infrastructure and runs your functions in micro-containers
- can be attached to VPCs - due to the Hyperplane integration in 2019, this doesn't come with restrictions like ENI bootstrap or private IP consumption anymore
- can be triggered via notifications from other services like SQS or S3
- destinations: on successful or failed executions you can invoke other services to handle those events
- exposure of your functions to the internet either via API Gateway or Function URLs
- dependencies can be extracted into Layers that can be attached to one or several functions
- usage based pricing

## CloudFront

Edge-computing locations.  customer's



- AWS distributes your content to more than 225 edge locations & 13 regional mid-tier caches on six continents and 47 different countries
- origins define the sources where a distribution retrieves its content if it's not cached yet
- a single distribution can use multiple origins
- caching behavior is controlled by a cache policy, either AWS managed or custom (which parts of the request should be used as a cache-key)
- Lambda@Edge and CloudFront functions allow you to run general-purpose code on edge locations close to the customer
- with geo-restrictions, you can enforce approval or blocking lists for specific countries
- CloudFront supports AWS Web Application Firewall (WAF) that lets you monitor HTTP/s requests and control access to your content

## CloudFormation

The fundamental infrastructure-as-code tool at AWS.

- templates are the definition of the infrastructure resources that should be created by CloudFormation and how they are composed
- for each template, CloudFormation will create a stack which is a deployable unit and contains all your resources
- CloudFormation detects change sets for your stacks at deployment time to calculate what create/update/delete command it needs to run



- via outputs, you can reference other resources dynamically that may not exist yet

## Simple Queue Service (SQS)

A queuing service that allows you to build resilient, event-driven architectures.

- another core building block for serverless applications
- allows you to run tasks in the background
- offers different types of queues
- First-In-First-Out (FIFO): executes messages in the order SQS receives them
- Standard Queues: higher throughput but no guarantee of right ordering
- Dead-Letter-Queues (DLQs) allow you to handle failures and retries
- Retention periods define how long a messages stays in your queue until it's either dropped or redriven to a DLQ

## Simple Notification Service (SNS)

Managed messaging service to send notifications to customers or other services.

- consumers can subscribe to topics and then will receive all messages published to the topic
- comes in two different types, equal to SQS: FIFO and Standard

- messages can be archived by sending them to Kinesis Data Firehose (and afterward to either S3 or Redshift)

## Elastic Load Balancing (ELB)

Distribute traffic between computing resources.

- comes in four different flavours
- Classic (CLB): oldest types, works on both layer 4 and layer 7 - no longer featured on AWS exams
- Application (ALB): works on layer 7 and routes content based on the content of the request
- Network (NLB): works on layer 4 and routes based on IP data
- Gateway (GLB): works on layer 3 and 4 - mostly used in front of firewalls
- load balancing enhances fault-tolerance as automatically distributes traffic to healthy targets which can also reside in different availability zones
- ELB can be either internet facing (has public IP, needs a public VPC) or internal-only (private VPC, no public IP, can only route to private IP addresses)
- EC2 instances or Fargate tasks can be registered to ELB's target groups

## DynamoDB

A fully managed



- a non-relational database, based on Cassandra
- comes with two different capacity modes
- on-demand: scales based on the number of requests and you only pay per consumed read or write capacity unit
- provisioned: define how many read/write capacity units are needed and pay a fixed price per month - you can use auto-scaling policies together with CloudWatch alarms to scale based on work loads
- a (unique) primary key is either built via the hash key or the hash key and range key
- Global (can be created any time) and Local (only when your table is created) Secondary Indexes help you to allow additional access patterns
- can use streams to trigger other services like Lambda on create/update/delete events
- Time-to-Live (TTL) attribute allows for automatic expiry of items
- global tables can span multiple regions and automatically sync data between them
- encryption of tables by default: either via KMS keys managed by AWS or the customer itself
- a set of different data types (scalar, document, set)

## Identity and Access Management (IAM)

The core security building block for all applications running on AWS.

- follow best practices like never using your root user, but dedicated IAM users w.

- there are different entity types: users, groups, policies and roles
- user: end-user, accessing the console or AWS API
- group: a group of users, sharing the same privileges
- policy: a defined list of permissions, defined as JSON
- role: a set of policies, that can be assumed by a user or AWS service to gain all the policies permissions
- by default, all actions are denied and need to be explicitly allowed via IAM
- an explicit deny always overwrites an allow action

## CloudWatch

An observability platform that is integrated with almost every AWS service.

- log events: messages that are collected by CloudWatch from other services, always containing a timestamp
- log groups: a cluster of log messages, related to a service
- log streams: further drill down of messages, e.g. for a specific Lambda micro-container instance or Fargate task
- CloudWatch collects metrics by default from a lot of services, including Lambda, EC2, or ECS
- Default metrics do not come with costs
- X-Ray allows for distributed tracing to understand how a single request interacted with different services
- Alarms can be configured to send notifications via SNS on certain events or to trigger actions like auto-scaling policies

# CloudTrail

Monitor and record account activity across your AWS infrastructure (check [this article](#) for the differences with CloudWatch)

- records events in your AWS account as JSON
- you can decide which events are tracked by creating trails
- a trail will forward your events to an S3 bucket and/or CloudWatch log group
- CloudTrail records different types of audit events
- Management events: infrastructure management operations, e.g. IAM policy adjustments or VPC subnet creations
- Data Events: events that retrieve, delete or modify data within your AWS account, e.g. CRUD operations on DynamoDB or a GET for an object in S3
- Insight Events: records anomalies in your API usage of your account based on historical usage patterns
- you can additionally define filter rules to not track all events of a certain type, but only a subset. Maybe you're interested in tracking modifications and deletions in DynamoDB, but not reads.

## Taking the Online Exam with PSI

You can schedule online exams for all certifications with PSI. Exams can also be rescheduled or canceled (you'll get a full refund) until 24 hours before the exam without additional costs. For the exam itself make sure that...

- you've taken

