AWS Associate Developer Study Guide

Read https://www.selikoff.net/2019/03/02/how-i-recommend-studying-for-the-aws-associate-developer-exam/ before this study guide

Security

API use	• Better to use AWS roles within AWS than access/secret keys
Users	• People
Groups	• Users with shared permissions • Assign policies to groups
Roles	• Assigned to AWS resources • Users and applications assume roles • Grant permissions to entities: ex: user, app, service (ex: Ec2, Lambda) • Prevents having to use access/secret key • Controlled by policies
Policies	• JSON document defining permissions • Attach to user/group/role • Includes allow/deny, action and resources
Access key/secret access key	• Can only view secret access key once • If lose, must regenerate • Use if connecting from outside AWS
Policy Generator	• Creates JSON • Type: SQS, SNS, S3, VPC Endpoint, IAM Policy • Statements: Allow/Deny, Principal's ARN, AWS Service, Actions, ARN of resource • Principal can be * • Resource can be arn or arn/* for all within
ARN	• Amazon Resource Name • Ex: arn:aws:iam::accountId:user/name
Encryption types	• In transit (SSL/TLS) • At rest (keys) • Client side encryption
CloudHSM	• Hardware security module (for keys)

KMS

General	• Key Management Service • Create/control data encryption keys • Multi-tenant. • Best practice; user who manages keys can't encrypt/decrypt
Keys	• Encryption keys are regional. Must decrypt in region encrypted in • Customer master key – generated by Amazon or AWS provided. Cannot be exported.
APIs	• encrypt • decrypt

	• re-encrypt – re-encrypt with new master key and delete original encrypted file • enable-key-rotation – rotates the key yearly
Envelope encryption	• Encrypt envelope (data) key • Master key encrypts/decrypts envelope key • Envelope key encrypts/decrypts data
Systems Manager Parameter Store	• Store confidential information • Under EC2 in console • Create key/value parameter as string/list • Encrypt with KMS • Available from EC2, CloudFormation, Lambda

IAM

Overview	• Identity and Access Management • Manage users and their access to AWS Console • Pass role to EC2 via instance profile
Web Identity Federation	• Authenticate with Amazon/Facebook/Google • Trade web token/auth code for temporary AWS credentials
Cognito	• Identity Broker • Provides Web Identity Federation • Can also use SAML • Synchronizes user data across devices using SNS • Good for mobile apps • No custom code • User pools - user directories to manage sign up/sign in. Generators JSON Web Tokens • Identity pools – unique identity/temp creds • To configure; needs user pool, app client and domain
Policy types	• Managed Policies – created by AWS. • Customer Managed Policies – only within your account. More granular than built in. • Inline Policies – embedded in user/group/role. Only use if need to ensure will not be reused.
AssumeRoleWithWebIdentity	• From STS (Security Token Service) • API to return temporary security creds • Key for creds: Includes AssumedRoleUser ARN and AssumedRoleID (not IAM role) • Creds include session token, access key, secret access key

	• Temporary creds default to an hour
Cross account access	• Same creds to use multiple accounts in console. Don't need to re-login • Can grant specify policies from one account

EBS

Overview	• Virtual disk • Block storage • Attach to EC2 instances • Stored in specific availability zone • Automatically replicated within zone
Encryption	• Creating volume from encrypted snapshot is encrypted • Creating volume from unencrypted snapshot is unencrypted • If copy an unencrypted snapshot to create a new snapshot, can encrypt it when creating the copy. Then can make AMI of it to have encrypted root device.

S3

Overview	• Object storage, key/value pairs • Not for database, OS • Unlimited storage • High availability/disaster recovery built in • Zero bytes - 5TB. Can upload up to 5GB with PUT. Use multi-part upload API for 100MB+ • Files stored in buckets or folders within buckets, no nested buckets • Bucket names must be globally unique • Basic charges for storage, data transfer and requests • Buckets partitioned by key name • URL: bucket.s3.location.amazon.aws.com
Data consistency	• Read after write – available right away. For new objects(PUTS) • Eventual consistency – can take time to propagate. For updating objects (PUTS)/deleting objects (DELETES)
Storage Tiers	• S3 – 99.99% availability, 11 9's durability. Stored across multiple facilities. Designed to sustain loss of two data centers • S3 - IA (Infrequently accessed) – Lower fee, but charged every time access. Min 30 days

	• S3 – One Zone IA – 99.5% availability. Only in one availability zone. Min 30 days • Reduced Redundancy Storage – 99.99% durability. For data that can be recreated if lost. Not recommended for use. • Glacier – for archiving. Very cheap. For data infrequently accessed. Several hours to retrieve data. Min 90 days • S3 – Intelligent Tiering – 2 tiers. Automatically moves data to most cost effective tier based on how frequently access. New option. • Seet using x-amz-storage-class header
Security	• Buckets private by default • Bucket policies at bucket level • Access control lists – at object level • Can log all access to bucket in another bucket
Encryption at rest	• AES-256/SSE-S3 – S3 Managed keys - each object gets own key • AES-KMS/SSE-KMS – Key Management Service – additional key to encrypt data's encryption key. Get audit trail of when key used. • SSDE-C – Customer provided keys • Enable when creating a bucket. Alternatively, create Bucket policy to deny all PUTS without x-amz-server-side-encryption header
CORS	• Cross Origin Resource Sharing • Avoids same origin policy problem (which prevents XSS) • Allow a resource in one bucket to access one in another bucket • Configure as XML on bucket being referenced from elsewhere and specify bucket that can access • Enforced by client

CloudFront

CDN	• Content Delivery Network • Serve static content from closer location around world • Has Viewer protocol policy
Edge location	• Where content is cached and can be written • More edge locations than availability zones • Not read only. Can PUT an object to S3
Origin	• Where content starts – S3, EC2, ELB, Route 53

Distribution	• Web distributions – websites • RTMP (real time messaging protocol) – media streaming
S3 Transfer Acceleration	• Uses edge locations to route to S3
Caching	• Stored for TTL (time to live) • Get charged to clear cache object before TTL
Using CloudFront	• URL cloudfront.net • Takes about 15 minutes to propagate initially • Slow first time because caching at edge location

Lambdas

Overview	• Serverless • Scales continuously with more functions (can't automatically add memory) • Very cheap • Compute service • Upload code to create Lambda function • Use cases: event driven compute service, compute service in response to HTTP requests • Max timeout changed 5 minutes to 15 minutes • Max 50MB compressed/250MB uncompressed
Languages	• Node.JS, Java, Python, C#, Go, Ruby
Debugging	• Lambdas can call other lambdas • Debug with XRay
Triggers	• Data Stores – S3, DynamoDB, Kinesis • Endpoints – API Gateway, IOT, Step Functions, Alexa • Dev/Management Tools – CloudFormation, CloudTrail, CodeCommit, CloudWatch • Event/Message Services – SES, SNS, SQS, cron • Other - Cognito
Version control	• Versions have different ARN • Versions are immutable • Qualified ARN has version suffix. • $LATEST is a built in qualifier. Create more versions by publishing new version • Unqualified ARN lacks version suffix • Alias – name pointing to a version. Ex: QA • Can only edit $LATEST • Can have 0+ aliases for a version • Can do blue/green deployments by setting up an alias pointing to two versions with % split. (can't use $LATEST)
Step Functions	• Type of application integration

	• Graphical console to arrange/visualize components • Automatically triggers next step • Types of steps – sequential, branching, parallel • Coded in JSON using Amazon State Language • Generates Lambda Functions • Logs each step • Differs from SWF (simple workflow service) in that can only have one state definition vs multiple deciders. Also deciders can't be implemented as lambdas. Use step functions as first choice and SWF if doesn't meet needs
To rollback	• Change PROD alias

ELB

Overview	• Elastic load balancer • Equally balance load
Application Load Balancer	• Layer 7 (app layer) • Can make decisions based on content • HTTP/HTTPS traffic
Network Load Balancer	• Layer 4 • TCP traffic • Extreme performance/low latency • Assumes static IP addresses • Most expensive
Classic Load Balancer	• Legacy; no longer recommend [but on exam] • Can use layer 4 or 7
X-Forwarded-For	• Original (public IP) • Load balancer converts public IPv4 address to private IP

Route 53

Overview	• DNS • Map domain names to EC2, load balancer, S3 buckets
Create record set	• Maps domain name abc.com to AWS resource

API Gateway

Overview	• Managed service • Publish/maintain/monitor/secure APIs • "Front door" for APIs in EC2/Lambda/web app • Exposes HTTPS REST endpoints • Each endpoint has a different target • Can log to CloudWatch

	• Can configure multiple versions of API
To use	• Define API (container) • Define resources (URL Paths) • Chose HTTP verbs, set security, choose targets, set transformations • Free SSL/TLS certs if using Route 53
API caching	• Reduce load/improve latency • Set TTL in seconds
Security	• CORS if using multiple domains • Use API key to track/control usage • Can throttle usage
Create API	• Has visual editor • Can create from New/existing/example or swagger API • Supports Open API • Set HTTP verb • Set integration type (lambda, HTTP, mock, AWS service, VPC link) • Set proxy integration so lambda can see request • Set lambda function name and region • Can upload via copy/paste of zip file
Deploy API	• Makes API available • Can test from here • Shows stages (like tags)
Import API	• Can import from Swagger • Can create new or merge existing definition
Throttling	• 10K requests/second • max 5000 concurrent requests/millisecond across AWS account
SOAP	• Doesn't handle, but can pass through
Access control	• IAM roles, lambda authorizers, Cognito pools

X-ray

Overview	• Collects data about requests • X-ray SDK in side your app. API sends to X-ray • SDK adds interceptors to code to trace HTTP requests, calls to other AWS services and web services • View using X-ray console – shows error rate, traces, timings
Integrations	• Works with Lambda, EC2, API Gateway, Elastic Beanstalk and ECS

RDS

Overview	• Relational Database Service
Database Types	• SQL Server, Oracle, MySQL, PostGres, Aurora, MariaDB
Aurora	• Compatible with MySQL
Security	• If EC2 and RDS in different security groups, need to open port 3306
Backups	• Automated backup – 1-35 days. Daily snapshot and transaction logs throughout day. Enabled by default. Get free storage space matching RDS disk space. • Database snapshots – manual. Kept even after delete RDS instance • When restore, get new RDS instance with new DNS endpoint. Can restore to any point in time.
Multi AZ	• For disaster recovery only • Synchronously replicated to standby in another availability zone • Automatic failover. Name stays same even though IP changes
Read replica	• For performance/scaling • Up to 5 read replicas • Requires automatic backups to be enabled • Can have read replicas of read replicas. Latency. • Can be in different availability zone or region • Not available for SQL Server or Oracle • Read replica can have Multi-AZ • Can “clone” to be own db and turn off replication • Can encrypt even if source is not encrypted

DynamoDB

Overview	• NoSQL database • Fully managed, autoscales • Single digit millisecond latency • Supports key-value and document data models • Stored on SSD • Spread across 3 data centers • Supports conditional writes and optimistic locking with version numbers
Consistency Model	• Eventually Consistent Reads – default. Might see stale data but usually less than a second to propogate data • Strongly Consistent Reads. All writes will be available for read

Terms	• Tables • Item – single record • Attributes – key/value pairs • Key - name of data • Value - data • Documents – JSON, HTML and XML • Partition – physical storage location
Primary key	• Partition key – unique attribute that hashes to partition • Composite key – partition key + sort key. Partition key doesn't need to be unique but combined key does
Security	• IAM Condition – restricts access by record • Partition key must match user id • Must create new table to encrypt
Scan	• Looks at everything in table • Can add filter to limit results returned
Query	• ProjectionExpression – limit attributes returned • KeyCondition – like where clause • Better performance than scan • Must include primary key • Results sorted by sort key (or reverse with ScanIndexForward=false) • Defaults to Eventually Consistent
Local Secondary Index	• Must be created when create table, cannot add/remove later • Same partition key as table • Different sort key
Global Secondary Index	• Can create when create table or later • Different partition key than main table
Performance	• Can reduce impact by setting smaller page size to avoid throttling. • Can configure parallel scans. Bad if table already under heavy load
Capacity Units	• Measure of provisioned throughput • 1 write capacity unit is one 1KB write/second • 1 read capacity unit is 1 strongly consistent read of 4KB/second • 1 read capacity unit is 2 eventually consistent reads of 4KB/second • No fractional capacity units. Round up.
On Demand Capacity	• (won't be on exam before May 2019) • Autoscales based on activity • Don't need to specify capacity in advance

	• Pay per request • Provisioned Capacity costs less if predicatable • Can switch one per day.
DynamoDB Accelerator (DAX)	• Fully managed, clustered in-memory cache • Up to 10x performance improvement • Microsecond response times • Ideal for read heavy bursty workloads • Writes to cache at same time as db • If not in cache, does eventually consistent get
Transactions	• (won't be on exam before May 2019) • ACID, span tables
TTL	• Time to live, measured since 1970 • Expiration time for data • Marked for deletion and deleted within 48 hours • Reduces cost by automatically removing data
Streams	• Time ordered sequence of modifications • Guaranteed delivery exactly once • Logs stored 24 hours • Encrypted at rest • Separate endpoint from stream than db • Primary key always stored. Before/after can be stored too • Can trigger events – ex: lambda
If too many requests	• ProvisionedThoroughputExceededError • SDK will automatically retry until successful. • Use exponential backoff (applies to most AWS services). SDK does automatically. Jitter adds randomness so don't all try at same second • Check if request size too big
Common APIs	• BatchGetItem, GetItem • BatchWriteItem, PutItem • DeleteItem, UpdateItem • Query, Scan
Global Tables	• Specify regions want table available • AWS replicates

Elasticache

Overview	• In memory cache in cloud
Supports	• Sand DynamoDB
Types	• Supports Memcached (memory object caching with no persistence) and Redis (in memory key-value store, works with Multi-AZ and maser/slave replication. Manages like RDS) • Use Memcached for: object caching, simple, large

	cache nodes with threads, scale horizonatally • Use Redis for: advanced data types, sorted/ranking (ex: leaderboard), persistence, failover, pub/sub [unless data warehousing, then RedShift]
Caching Strategy	• Lazy Loading – loads only when needed. Returns null if not found. Data can be stale if changed after placed in cache and before TTL expires • Write through – updates cache when data changes. Write penalty because updates even if not needed and updates even if not read.

SQS

Overview	• Simple queue service • Pull based • Up to 256 KB per message • Defaults to 4 days max in queue. Can increased to two weeks. • Can build in autoscaling • First AWS service
Types	• Standard – default queue, message delivered 1+, order not guaranteed • FIFO – message delivered exactly one in order. Ends with .fifo. Can use message group id to guarantee order within groups, when don't want overall FIFO.
Visibility timeout	• Number seconds message invisible after reader picks it up • Message deleted if job processed. Else available for processing again • Default 30 seconds • Maximum 12 hours
Polling types	• Short polling – returns immediately • Long polling – waits for response or timeout. Maximum/default 20 seconds. Saves money when queue typically empty
Delay	• How long before readers see message when new

SNS

Overview	• Simple Notification Service • Push based, pub-sub • Send to topics. Can have multiple subscribers
Types	• Devices, SMS, Email, Email JSON, SQA, HTTP, lambda

SES

Overview	• Simple Email Service • Can deliver to S3 or trigger lambda/SNS • Can use for incoming mail • Doesn't require subscribing from the user

Kinesis

Overview	• Receive streaming data
Kinesis Streams	• Stores received data or video • Stored for a day; can increase to a week • Stores in shards. Use more to increase read/write in parallel • Send to consumers
Kinesis Firehose	• Data analyzed immediately using lambda or forwarded. Not stored locally • Forward data to S3 or ElasticSearch • Can forward from S3 to RedShift
Kinesis Analytics	• Run SQL queries from firehose/streams and send results to S3/ElasticSearch/RedShift

Developer Tools

CI/CD	• Continuous Integration • Continuous Delivery – prepared for release • Continuous Deployment – actually deployed
CodeCommit	• Private git repository • Has notifications tab to send SNS message
CodeBuild	• Build management system • Tests/create package: ex: Docker • Data encrypted in transit and at rest (HTTPS/SSL connection only) • SNS notification or trigger on repo events • buildspec.yml file lists phases and commands • Commands are UNIX Commands • Full CodeBuild log in CloudWatch; partial log in CodeBuild console
CodeDeploy	• Deploy to EC2, on-prem or lambdas • Automatically scales • Integrates with external tools (ex: Jenkins, etc) • Deployment Group – set of EC2/Lambda. Use tag to reference • Deployment – process/components to deploy • Deployment Config – rules and success/failure conditions • AppSpec File – deployment actions

	• Revision – all artifacts to deploy • Application – unique key • For Lambda, YAML or JSON. Set version (must be 0.0), resources (lambda and properites: name/alias/current version/target version) and hooks (BeforeAllowTraffic/AfterAllowTraffic lambdas) • For EC2, YAML. version (0.0), os (linux/windows), files (source/destination pairs), hooks (BeforeInstall/AfterInstall scripts location and timeout). Place appspec.yml in root dir.
CodePipeline	• Continuous deployment service to visualize/automate • Workflow of stages/tasks • Automatically configured so commits trigger CloudWatch which triggers CodePipeline • Enable versioning in S3 bucket • Get code from S3, CodeCommit or GitHub • Manual approvals steps fails if not approved in a week
OpsWorks	• Manage infrastructure/layers • Supports Chef and Puppet
Docker	• Runs on EC2 • docker build –t name • docker tag name:latest awsUrl/name:version • docker push awsUrl/name:version

CloudFormation

Overview	• Infrastructure as code • Template supports YAML and JSON • Can use to create/rollback/delete entire stack • Store template in S3
Stack	• Resources created
Template	• Resources section mandatory • Optional sections: AWSTemplateFormatVersion (must be 2010-09-09), Description, Metadata, Parameters (input when run. For Prod or Test), Conditions (based on env), Mappings (by region), Transform (include external code from s3), Outputs (to browser of another template)
SAM	• Serverless application model • CloudFormation extension • Simplified syntax for APIs/Lambdas/Dynamo

	• SAM CLI • Package - converts similified yaml to cloud formation yaml and uploads to s3 • Deploy – deploys app using sam yaml
Nested stacks	• Code reuse • Standard template for component • Include under resources as AWS::CloudFormation::Stack and include template url on s3

Elastic Beanstalk

Overview	• Upload code and automatically provision infrastructure • Control AWS resources created • Pay for EC2 and S3 created/used • Automatically scales • Integrated with CloudWatch and Xray
Configuring in zip/war	• YAML or JSON format • In folder .ebextensions file *.config
Integrating with RDS	• Launch from Elastic Beanstalk console. Within Elastic Beanstalk environment so deleted when delete app • For Prod, create standalone RDS. Create extra security group in autoscaling group. Add connection information to RDS.
Security	• Can set roles on service on instance
Supports	• Tomcat, Passenger, Docker

Deployment strategies

All at once/in place	• Outage while update all If update fails, redeploy old version • Don't use in Prod. • Elastic Beanstalk: All at once • CodeDeploy: In Place deployment - All at Once
Rolling	• Deploys in batches. • Less live instances while batches down for update. Can reduce performance • Repeat to rollback • Elastic Beanstalk: Rolling, Rolling with Additional Batch (the later launches a new batch so live instance count unchanged) • Code Deploy: In Place Deployment – One at a Time, Half at a Time
Immutable/blue green	• Starts new servers with new code

	• Maintains full capacity • Rollback is terminating new instances • Elastic Beanstalk: Immutable • Elastic Beanstalk & Code Deploy: Blue/Green • Blue/Green – (blue = active; green = new) Deploys to new environment Keeps old up for rollback. • Immutable – new autoscaling group – New instances are provisioned with new server in new environment and swap DNS.

CloudWatch

Overview	• Monitors performance and other stats • Can install agent for on-prem data
Host level metrics	• CPU, Network, Disk, Status Check of EC2 instance
Custom metrics	• RAM Utilization • Get data at minimum once a minute even if script runs more often
Frequency	• Default – 5 minute intervals • Detailed – 1 minute intervals • High resolution – 1 second intervals
Data storage	• Forever unless configure otherwise. • Logs not deleted when EC2/ELB terminated
Alarm	• Monitor any metric for sustained state changes • States: OK, ALARM, INSUFFICIENT_DATA • Criteria: period of time, evaluation period, data points to alarm
Not done by CloudWatch	• CloudTrail does API calls • Config does state change

Other

EC2	• Elastic cloud • Like virtual server
Serverless services	• Lambda, API Gateway, S3, DynamoDB, SNS, SQS, Step Functions, Kinesis, Athena (queries), Tooling
ECS	• Elastic Container Service – manages containers
ECR	• Elastic Container Registry – image repo

HTTP Error codes

2xx	• Success

3xx	• Redirection

4xx	• Client error
	  ○ 400 – Bad request
	  ○ 401 – Unauthorized
	  ○ 403 – Forbidden
	  ○ 404 – Not found
	  ○ 409 – Conflict
	  ○ 429 – Too many requests/throttling error

5xx	• Server error
	  ○ 500 – Server error
	  ○ 502 – Bad gateway
	  ○ 503 – Service unavailable
	  ○ 504 – Gateway timeout/not responding