

Amazon S3 Section

Section introduction



- Amazon S3 is one of the main building blocks of AWS
- It's advertised as "infinitely scaling" storage
- Many websites use Amazon S3 as a backbone
- Many AWS services use Amazon S3 as an integration as well
- We'll have a step-by-step approach to S3
- The CCP exam requires "deeper" knowledge about S3

S3 Use cases

- Backup and storage
- Disaster Recovery
- Archive
- Hybrid Cloud storage
- Application hosting
- Media hosting
- Data lakes & big data analytics
- Software delivery
- Static website



Nasdaq stores 7 years of data into S3 Glacier



Sysco runs analytics on its data and gain business insights

Amazon S3 Overview - Buckets

- Amazon S3 allows people to store objects (files) in “buckets” (directories)
- Buckets must have a globally unique name (across all regions all accounts)
- Buckets are defined at the region level
- S3 looks like a global service but buckets are created in a region
- Naming convention
 - No uppercase
 - No underscore
 - 3-63 characters long
 - Not an IP
 - Must start with lowercase letter or number



Amazon S3 Overview - Objects

- Objects (files) have a Key
- The **key** is the FULL path:
 - `s3://my-bucket/my_file.txt`
 - `s3://my-bucket/my_folder/another_folder/my_file.txt`
- The key is composed of **prefix** + **object name**
 - `s3://my-bucket/my_folder/another_folder/my_file.txt`
- There's no concept of “directories” within buckets (although the UI will trick you to think otherwise)
- Just keys with very long names that contain slashes (“/”)



Amazon S3 Overview – Objects (continued)

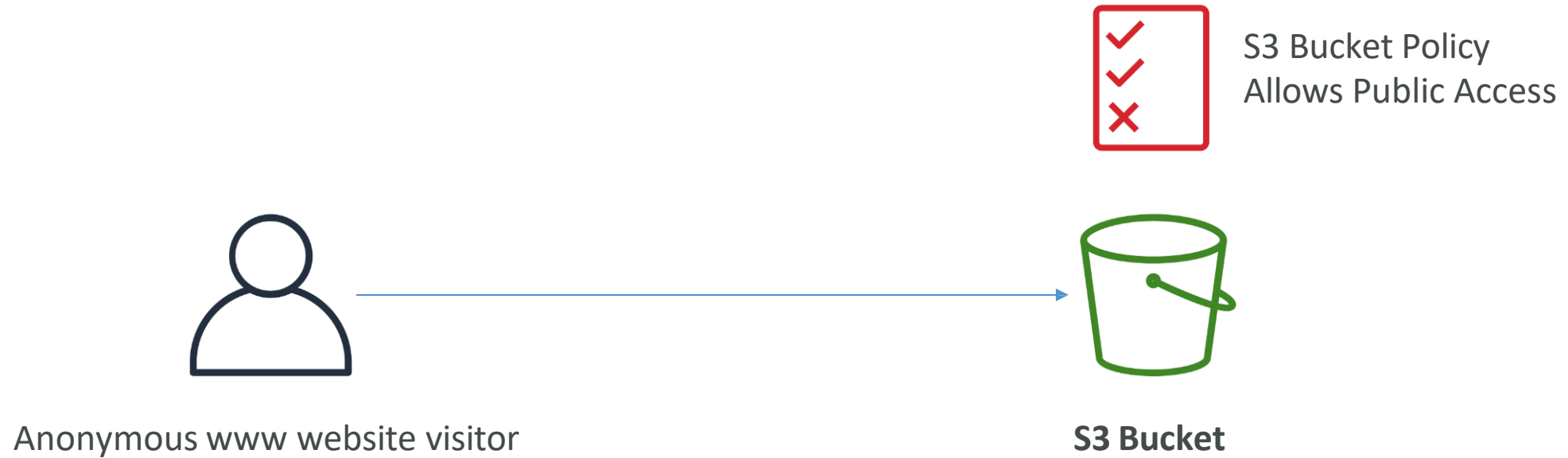
- Object values are the content of the body:
 - Max Object Size is 5TB (5000GB)
 - If uploading more than 5GB, must use “multi-part upload”
- Metadata (list of text key / value pairs – system or user metadata)
- Tags (Unicode key / value pair – up to 10) – useful for security / lifecycle
- Version ID (if versioning is enabled)



S3 Security

- User based
 - IAM policies - which API calls should be allowed for a specific user from IAM console
- Resource Based
 - Bucket Policies - bucket wide rules from the S3 console - allows cross account
 - Object Access Control List (ACL) – finer grain
 - Bucket Access Control List (ACL) – less common
- Note: an IAM principal can access an S3 object if
 - the user IAM permissions allow it OR the resource policy **ALLOWS** it
 - AND there's no explicit DENY
- Encryption: encrypt objects in Amazon S3 using encryption keys

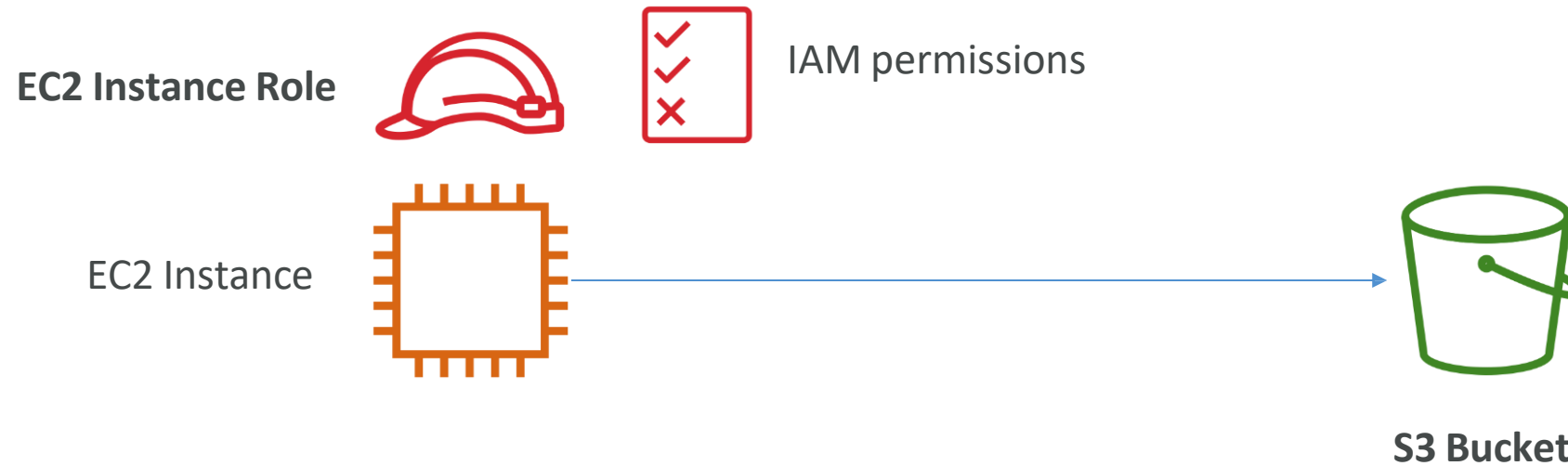
Example: Public Access - Use Bucket Policy



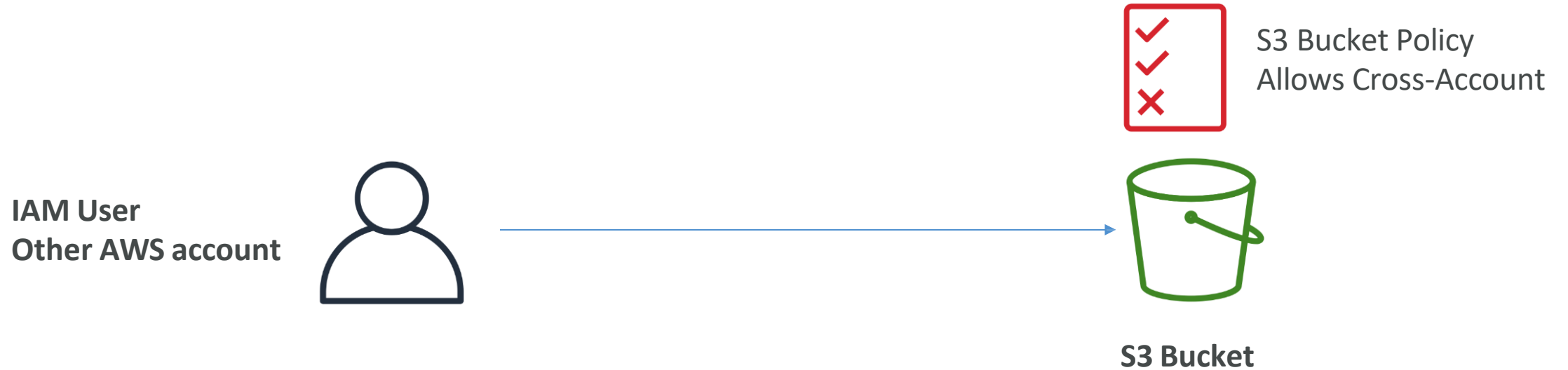
Example: User Access to S3 – IAM permissions



Example: EC2 instance access - Use IAM Roles



Advanced: Cross-Account Access – Use Bucket Policy

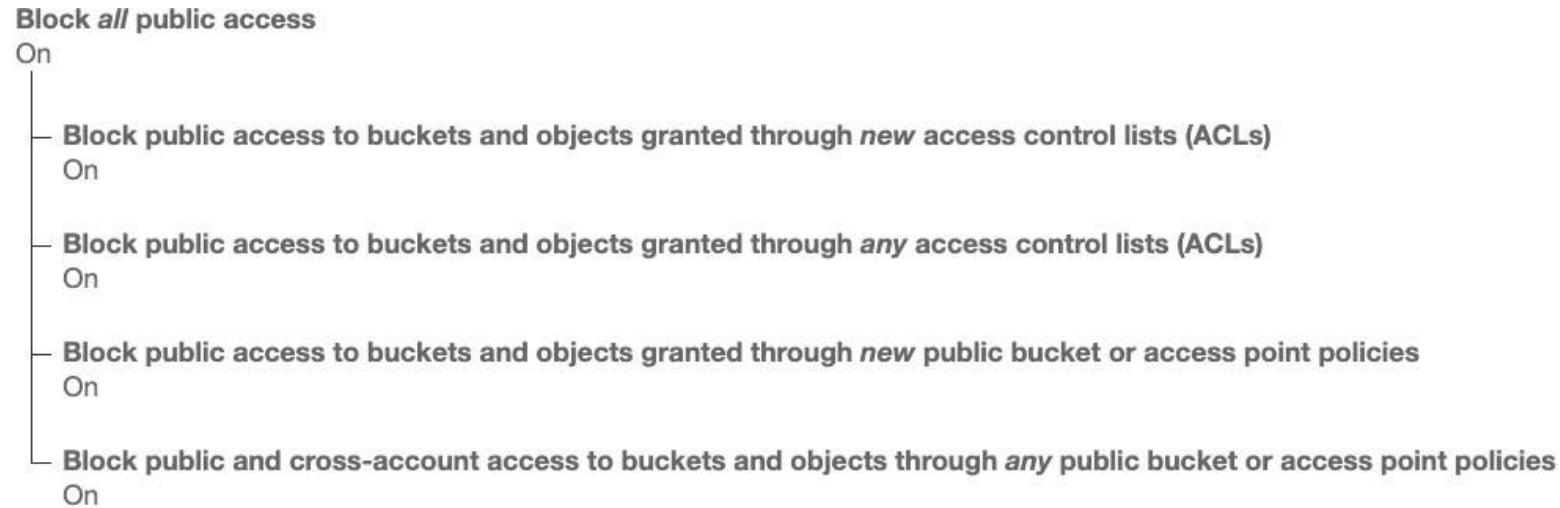


S3 Bucket Policies

- JSON based policies
 - Resources: buckets and objects
 - Actions: Set of API to Allow or Deny
 - Effect: Allow / Deny
 - Principal: The account or user to apply the policy to
- Use S3 bucket for policy to:
 - Grant public access to the bucket
 - Force objects to be encrypted at upload
 - Grant access to another account (Cross Account)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicRead",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket/*"
      ]
    }
  ]
}
```

Bucket settings for Block Public Access



- These settings were created to prevent company data leaks
- If you know your bucket should never be public, leave these on
- Can be set at the account level

S3 Websites

- S3 can host static websites and have them accessible on the www
- The website URL will be:
 - <bucket-name>.s3-website-<AWS-region>.amazonaws.comOR
 - <bucket-name>.s3-website.<AWS-region>.amazonaws.com
- If you get a 403 (Forbidden) error, make sure the bucket policy allows public reads!

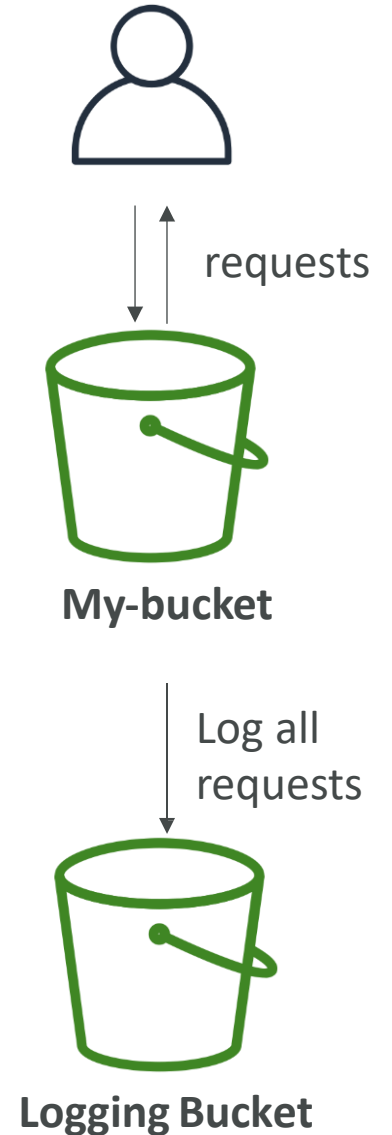
Amazon S3 - Versioning



- You can version your files in Amazon S3
- It is enabled at the bucket level
- Same key overwrite will increment the “version”: 1, 2, 3....
- It is best practice to version your buckets
 - Protect against unintended deletes (ability to restore a version)
 - Easy roll back to previous version
- Notes:
 - Any file that is not versioned prior to enabling versioning will have version “null”
 - Suspending versioning does not delete the previous versions

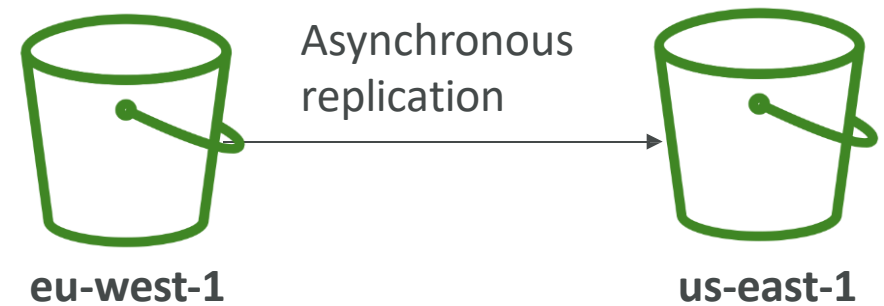
S3 Access Logs

- For audit purpose, you may want to log all access to S3 buckets
- Any request made to S3, from any account, authorized or denied, will be logged into another S3 bucket
- That data can be analyzed using data analysis tools...
- Very helpful to come down to the root cause of an issue, or audit usage, view suspicious patterns, etc...



S3 Replication (CRR & SRR)


- Must enable versioning in source and destination
- Cross Region Replication (CRR)
- Same Region Replication (SRR)
- Buckets can be in different accounts
- Copying is asynchronous
- Must give proper IAM permissions to S3



- CRR - Use cases: compliance, lower latency access, replication across accounts
- SRR – Use cases: log aggregation, live replication between production and test accounts

S3 Storage Classes

- Amazon S3 Standard - General Purpose
 - Amazon S3 Standard-Infrequent Access (IA)
 - Amazon S3 One Zone-Infrequent Access
 - Amazon S3 Intelligent Tiering
 - Amazon Glacier
 - Amazon Glacier Deep Archive

 - Amazon S3 Reduced Redundancy Storage (deprecated - omitted)
- 

S3 Durability and Availability

- Durability:
 - High durability (99.999999999%, 11 9's) of objects across multiple AZ
 - If you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years
 - Same for all storage classes
- Availability:
 - Measures how readily available a service is
 - S3 standard has 99.99% availability, which means it will not be available 53 minutes a year
 - Varies depending on storage class

S3 Standard – General Purposes

- 99.99% Availability
 - Used for frequently accessed data
 - Low latency and high throughput
 - Sustain 2 concurrent facility failures
-
- Use Cases: Big Data analytics, mobile & gaming applications, content distribution...

S3 Standard – Infrequent Access (IA)

- Suitable for data that is less frequently accessed, but requires rapid access when needed
- 99.9% Availability
- Lower cost compared to Amazon S3 Standard, but retrieval fee
- Sustain 2 concurrent facility failures
- Use Cases: As a data store for disaster recovery, backups...

S3 Intelligent-Tiering

- 99.9% Availability
- Same low latency and high throughput performance of S3 Standard
- **Cost-optimized by** automatically moving objects between two access tiers based on changing access patterns:
 - Frequent access
 - Infrequent access
- Resilient against events that impact an entire Availability Zone

S3 One Zone - Infrequent Access (IA)

- Same as IA but data is stored in a single AZ
- 99.5% Availability
- Low latency and high throughput performance
- Lower cost compared to S3-IA (by 20%)
- Use Cases: Storing secondary backup copies of on-premise data, or storing data you can recreate

Amazon Glacier & Glacier Deep Archive



- Low cost object storage (in GB/month) meant for archiving / backup
- Data is retained for the longer term (years)
- Various retrieval options of time + fees for retrieval:
- Amazon Glacier – cheap:
 - Expedited (1 to 5 minutes)
 - Standard (3 to 5 hours)
 - Bulk (5 to 12 hours)
- Amazon Glacier Deep Archive – cheapest:
 - Standard (12 hours)
 - Bulk (48 hours)

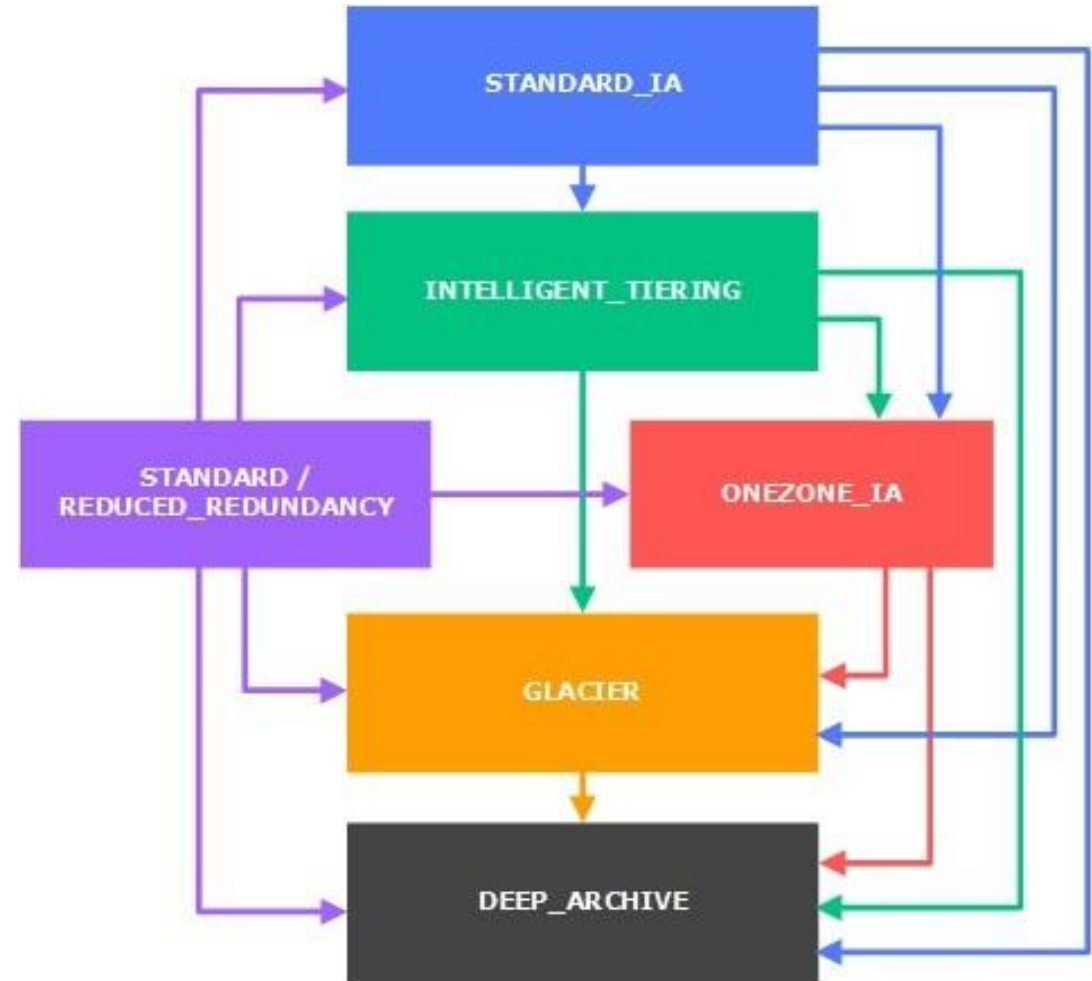
S3 Storage Classes Comparison

	S3 Standard	S3 Intelligent-Tiering	S3 Standard-IA	S3 One Zone-IA	S3 Glacier	S3 Glacier Deep Archive
Designed for durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99.9%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved

<https://aws.amazon.com/s3/storage-classes/>

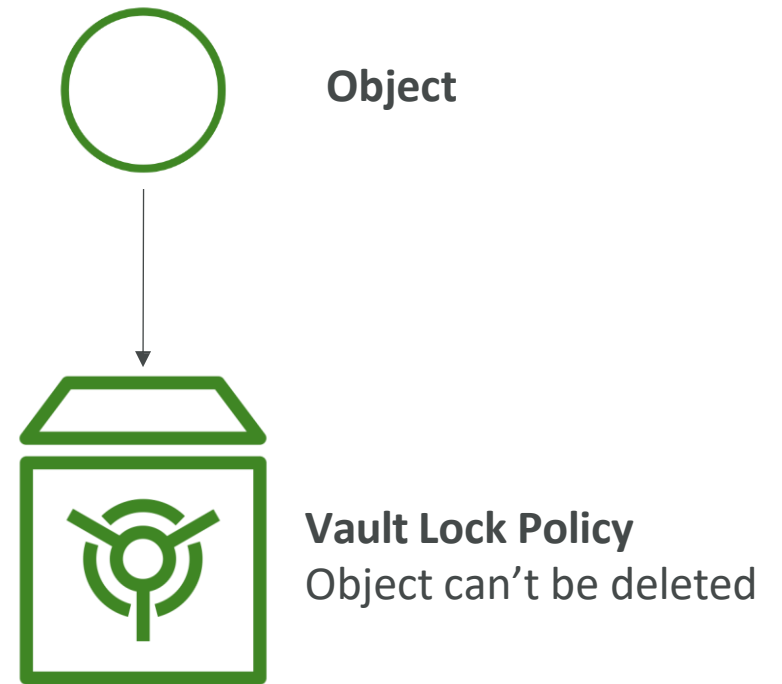
S3 – Moving between storage classes

- You can transition objects between storage classes
- For infrequently accessed object, move them to STANDARD_IA
- For archive objects you don't need in real-time, GLACIER or DEEP_ARCHIVE
- Moving objects can be automated using a lifecycle configuration



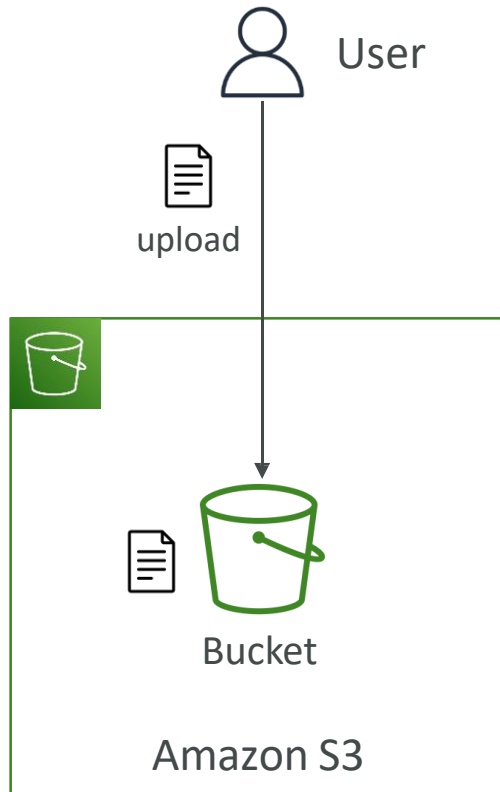
S3 Object Lock & Glacier Vault Lock

- S3 Object Lock
 - Adopt a WORM (Write Once Read Many) model
 - Block an object version deletion for a specified amount of time
- Glacier Vault Lock
 - Adopt a WORM (Write Once Read Many) model
 - Lock the policy for future edits (can no longer be changed)
 - Helpful for compliance and data retention

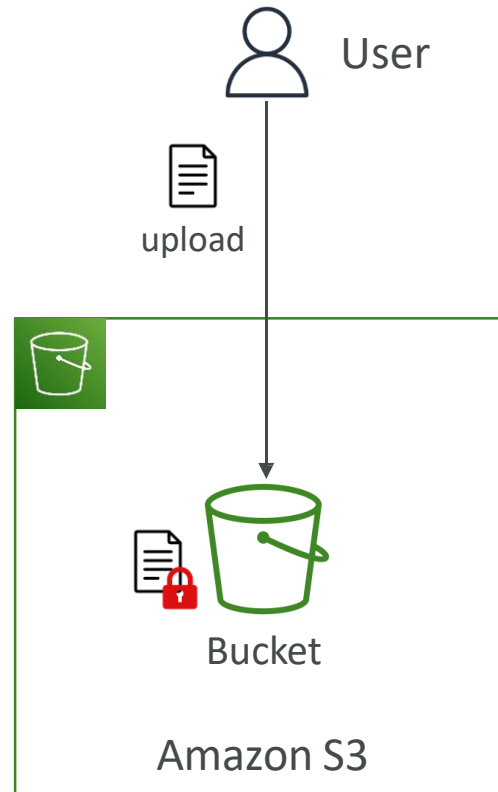


S3 Encryption

No Encryption

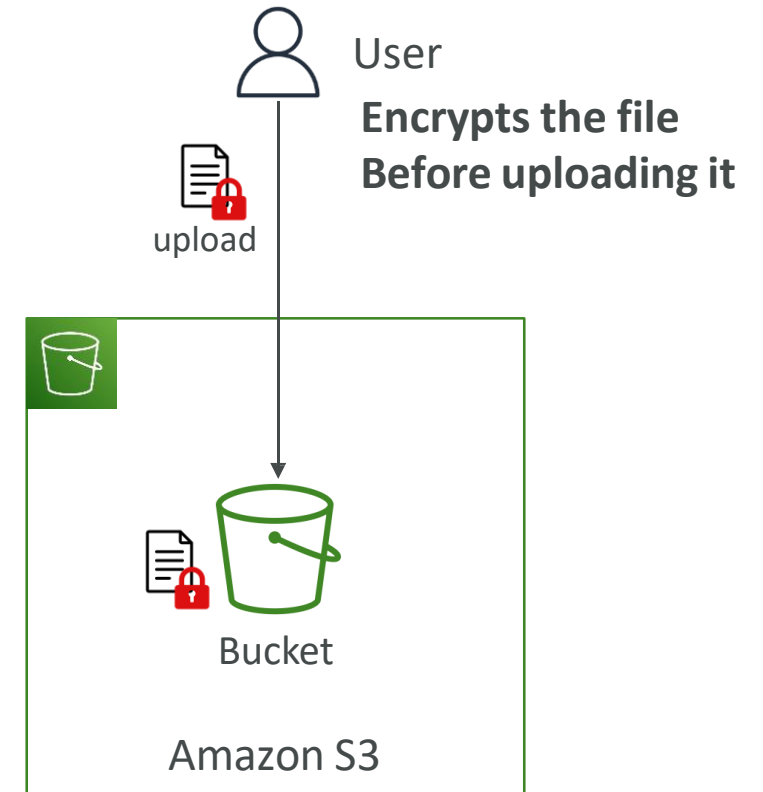


Server-Side Encryption



Server encrypts the file after receiving it

Client-Side Encryption



Shared Responsibility Model for S3



- Infrastructure (global security, durability, availability, sustain concurrent loss of data in two facilities)
- Configuration and vulnerability analysis
- Compliance validation
- S3 Versioning
- S3 Bucket Policies
- S3 Replication Setup
- Logging and Monitoring
- S3 Storage Classes
- Data encryption at rest and in transit

AWS Snow Family

- Highly-secure, portable devices to collect and process data at the edge, and migrate data into and out of AWS

- Data migration:



Snowcone



Snowball Edge



Snowmobile

- Edge computing:



Snowcone



Snowball Edge

Data Migrations with AWS Snow Family

	Time to Transfer		
	100 Mbps	1Gbps	10Gbps
10 TB	12 days	30 hours	3 hours
100 TB	124 days	12 days	30 hours
1 PB	3 years	124 days	12 days

Challenges:

- Limited connectivity
- Limited bandwidth
- High network cost
- Shared bandwidth (can't maximize the line)
- Connection stability

AWS Snow Family: offline devices to perform data migrations

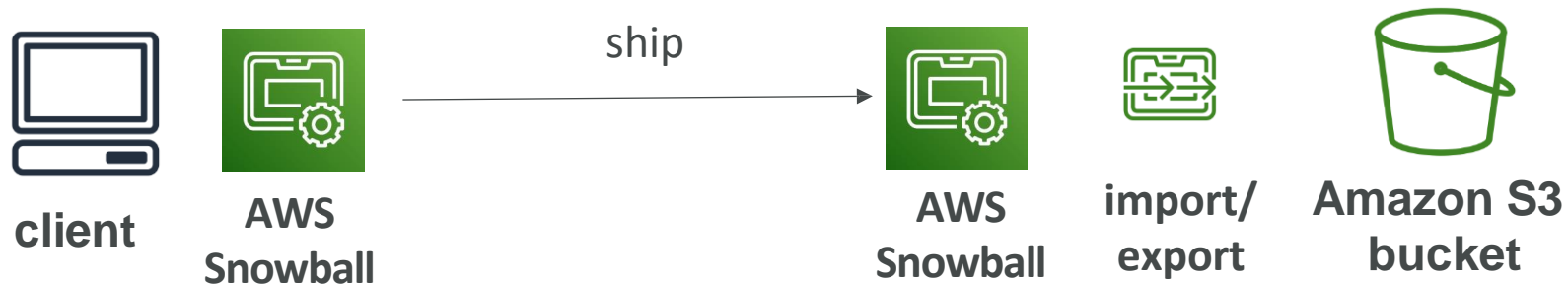
If it takes more than a week to transfer over the network, use Snowball devices!

Diagrams

- Direct upload to S3:



- With Snow Family:



Snowball Edge (for data transfers)



- Physical data transport solution: move TBs or PBs of data in or out of AWS
- Alternative to moving data over the network (and paying network fees)
- Pay per data transfer job
- Provide block storage and Amazon S3-compatible object storage
- Snowball Edge Storage Optimized
 - 80 TB of HDD capacity for block volume and S3 compatible object storage
- Snowball Edge Compute Optimized
 - 42 TB of HDD capacity for block volume and S3 compatible object storage
- Use cases: large data cloud migrations, DC decommission, disaster recovery



AWS Snowcone



- Small, portable computing, anywhere, rugged & secure, withstands harsh environments
 - Light (4.5 pounds, 2.1 kg)
 - Device used for edge computing, storage, and data transfer
 - 8 TBs of usable storage
 - Use Snowcone where Snowball does not fit (space-constrained environment)
 - Must provide your own battery / cables
-
- Can be sent back to AWS offline, or connect it to internet and use AWS DataSync to send data



AWS Snowmobile



- Transfer exabytes of data (1 EB = 1,000 PB = 1,000,000 TBs)
- Each Snowmobile has 100 PB of capacity (use multiple in parallel)
- High security: temperature controlled, GPS, 24/7 video surveillance
- Better than Snowball if you transfer more than 10 PB

AWS Snow Family for Data Migrations



Snowcone




Snowball Edge



Snowmobile

	Snowcone	Snowball Edge Storage Optimized	Snowmobile
Storage Capacity	8 TB usable	80 TB usable	< 100 PB
Migration Size	Up to 24 TB, online and offline	Up to petabytes, offline	Up to exabytes, offline
DataSync agent	Pre-installed		
Storage Clustering		Up to 15 nodes	

Snow Family – Usage Process

1. Request Snowball devices from the AWS console for delivery
 2. Install the snowball client / AWS OpsHub on your servers
 3. Connect the snowball to your servers and copy files using the client
 4. Ship back the device when you're done (goes to the right AWS facility)
 5. Data will be loaded into an S3 bucket
 6. Snowball is completely wiped
- 

What is Edge Computing?

- Process data while it's being created on an edge location
 - A truck on the road, a ship on the sea, a mining station underground...



- These locations may have
 - Limited / no internet access
 - Limited / no easy access to computing power
- We setup a Snowball Edge / Snowcone device to do edge computing
- Use cases of Edge Computing:
 - Preprocess data
 - Machine learning at the edge
 - Transcoding media streams
- Eventually (if need be) we can ship back the device to AWS (for transferring data for example)

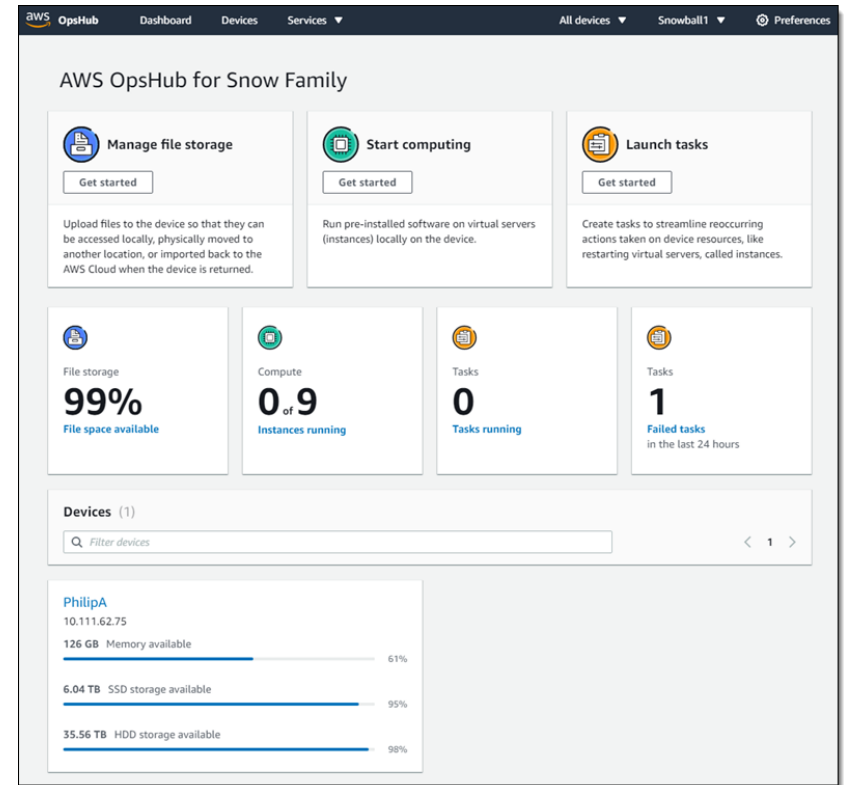
Snow Family – Edge Computing

- Snowcone (smaller)
 - 2 CPUs, 4 GB of memory, wired or wireless access
 - USB-C power using a cord or the optional battery
- Snowball Edge – Compute Optimized
 - 52 vCPUs, 208 GiB of RAM
 - Optional GPU (useful for video processing or machine learning)
 - 42 TB usable storage
- Snowball Edge – Storage Optimized
 - Up to 40 vCPUs, 80 GiB of RAM
 - Object storage clustering available
- All: Can run EC2 Instances & AWS Lambda functions (using AWS IoT Greengrass)
- Long-term deployment options: 1 and 3 years discounted pricing



AWS OpsHub

- Historically, to use Snow Family devices, you needed a CLI (Command Line Interface tool)
- Today, you can use AWS OpsHub (a software you install on your computer / laptop) to manage your Snow Family Device
 - Unlocking and configuring single or clustered devices
 - Transferring files
 - Launching and managing instances running on Snow Family Devices
 - Monitor device metrics (storage capacity, active instances on your device)
 - Launch compatible AWS services on your devices (ex: Amazon EC2 instances, AWS DataSync, Network File System (NFS))



<https://aws.amazon.com/blogs/aws/aws-snowball-edge-update/>

Hybrid Cloud for Storage

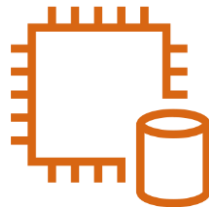
- AWS is pushing for "hybrid cloud"
 - Part of your infrastructure is on-premises
 - Part of your infrastructure is on the cloud
- This can be due to
 - Long cloud migrations
 - Security requirements
 - Compliance requirements
 - IT strategy
- S3 is a proprietary storage technology (unlike EFS / NFS), so how do you expose the S3 data on-premise?
- AWS Storage Gateway!

AWS Storage Cloud Native Options

BLOCK



Amazon EBS



EC2 Instance
Store

FILE



Amazon EFS

OBJECT



Amazon S3



Glacier

AWS Storage Gateway

- Bridge between on-premise data and cloud data in S3
- Hybrid storage service to allow on-premises to seamlessly use the AWS Cloud
- Use cases: disaster recovery, backup & restore, tiered storage
- Types of Storage Gateway:
 - File Gateway
 - Volume Gateway
 - Tape Gateway
- No need to know the types at the exam

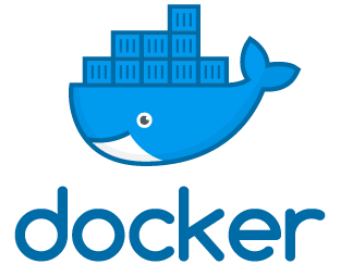


Amazon S3 – Summary

- Buckets vs Objects: global unique name, tied to a region
- S3 security: IAM policy, S3 Bucket Policy (public access), S3 Encryption
- S3 Websites: host a static website on Amazon S3
- S3 Versioning: multiple versions for files, prevent accidental deletes
- S3 Access Logs: log requests made within your S3 bucket
- S3 Replication: same-region or cross-region, must enable versioning
- S3 Storage Classes: Standard, IA, IZ-IA, Intelligent, Glacier, Glacier Deep Archive
- S3 Lifecycle Rules: transition objects between classes
- S3 Glacier Vault Lock / S3 Object Lock: WORM (Write Once Read Many)
- Snow Family: import data onto S3 through a physical device, edge computing
- OpsHub: desktop application to manage Snow Family devices
- Storage Gateway: hybrid solution to extend on-premises storage to S3

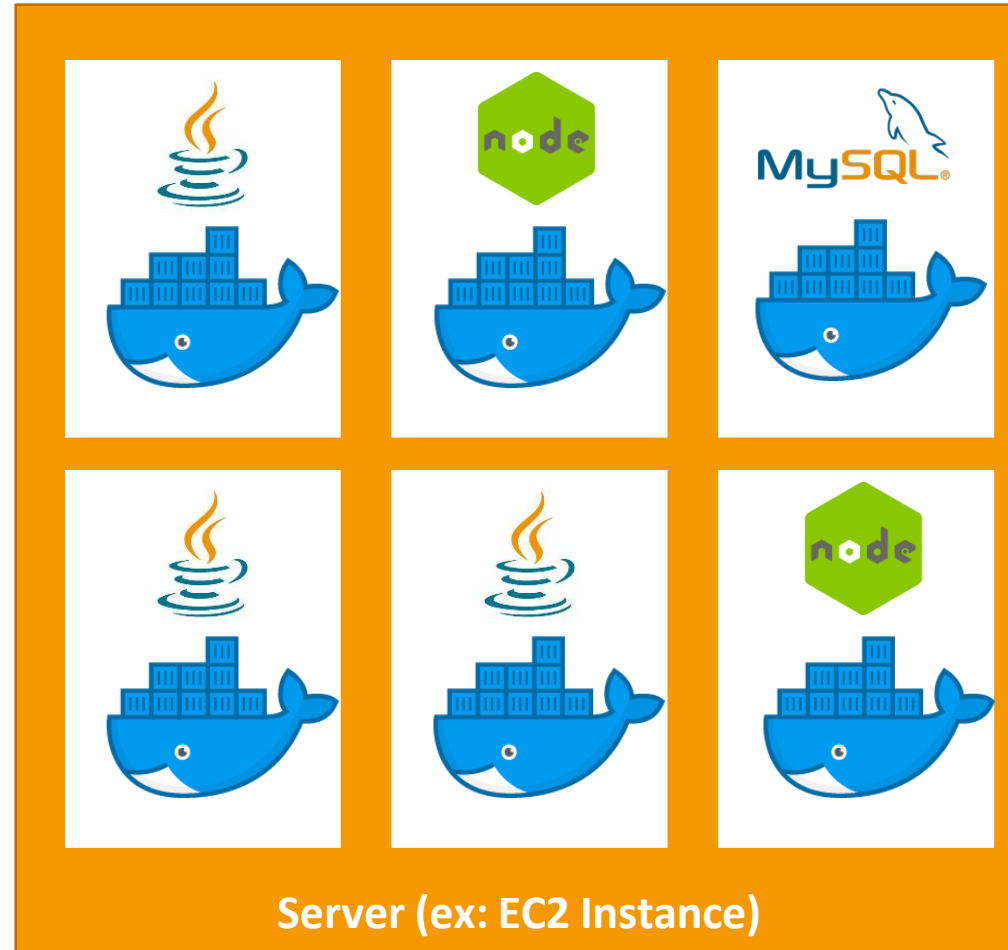
Other Compute Section

What is Docker?



- Docker is a software development platform to deploy apps
- Apps are packaged in containers that can be run on any OS
- Apps run the same, regardless of where they're run
 - Any machine
 - No compatibility issues
 - Predictable behavior
 - Less work
 - Easier to maintain and deploy
 - Works with any language, any OS, any technology
- Scale containers up and down very quickly (seconds)

Docker on an OS

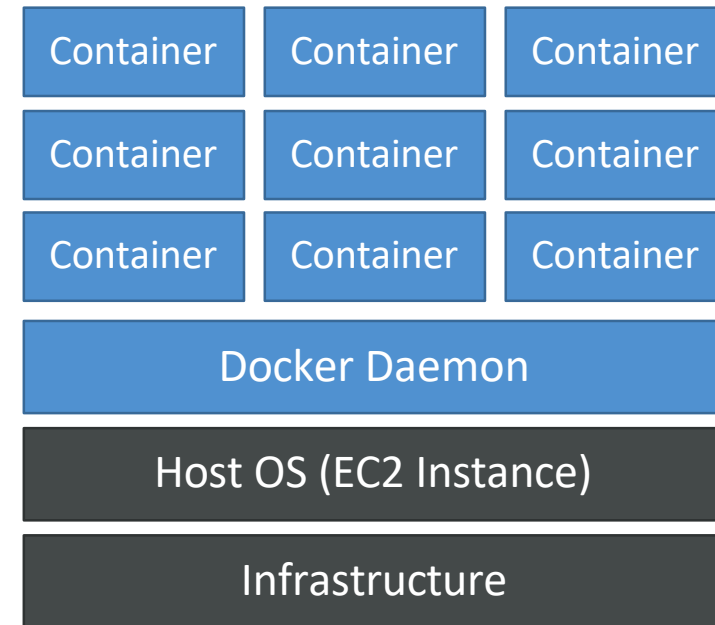
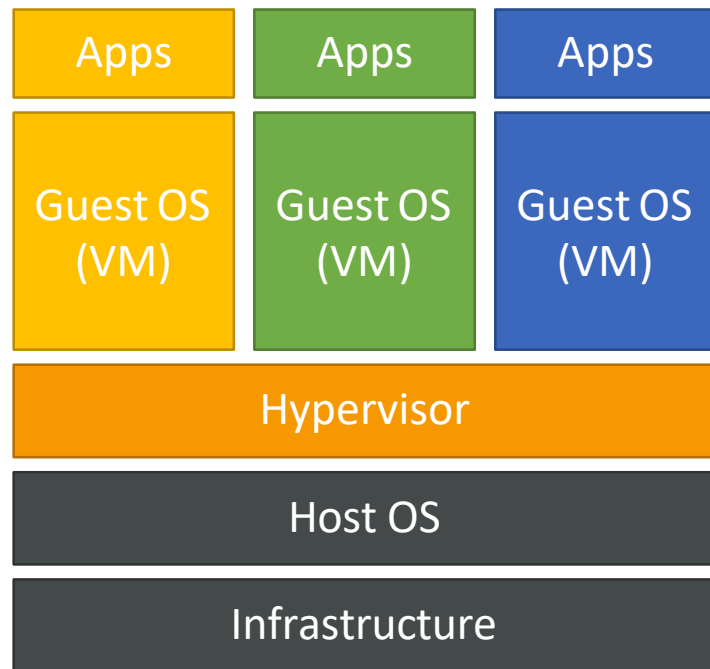


Where Docker images are stored?

- Docker images are stored in Docker Repositories
- Public: Docker Hub <https://hub.docker.com/>
 - Find base images for many technologies or OS:
 - Ubuntu
 - MySQL
 - NodeJS, Java...
- Private: Amazon ECR (Elastic Container Registry)

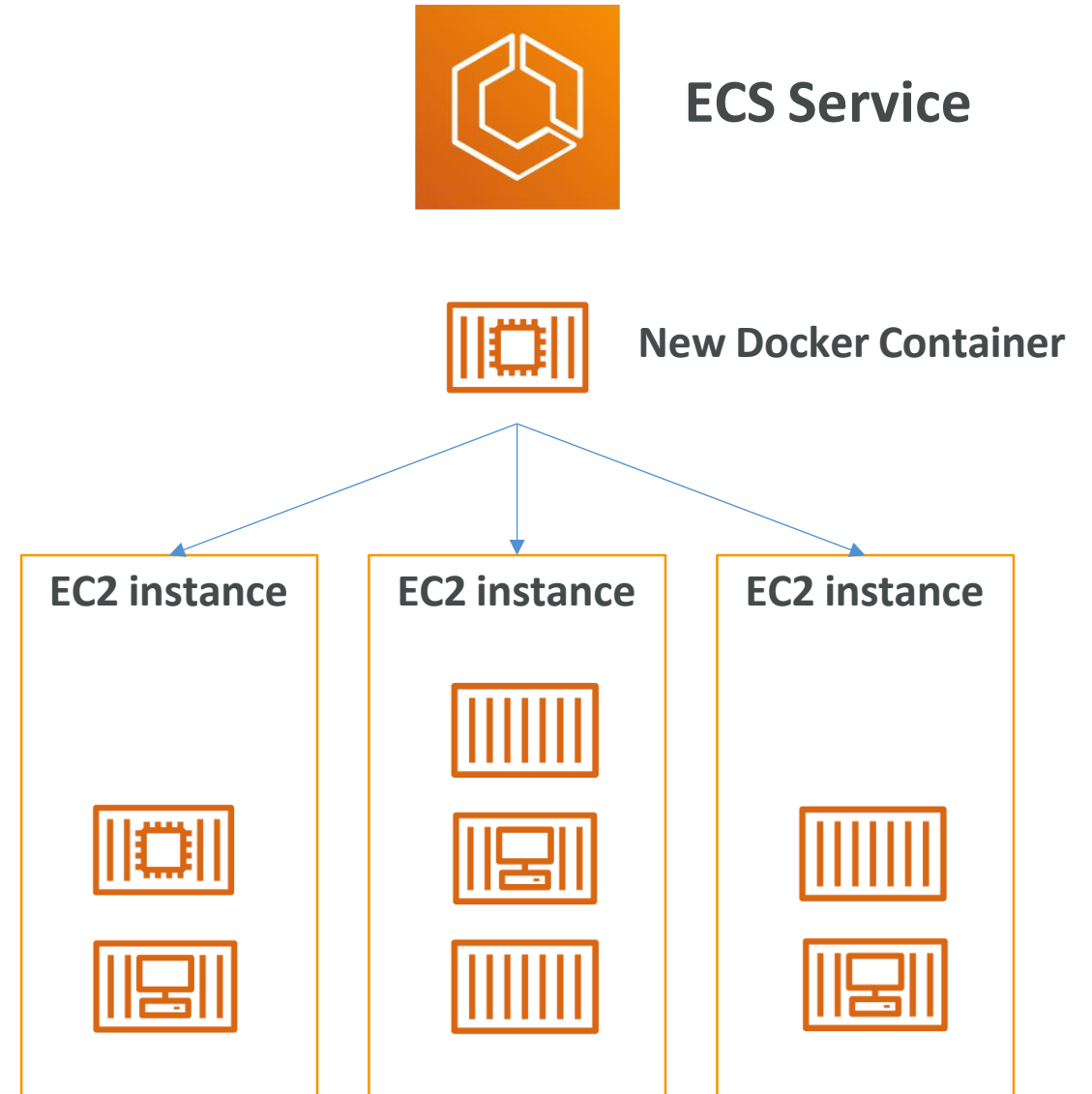
Docker versus Virtual Machines

- Docker is "sort of" a virtualization technology, but not exactly
- Resources are shared with the host => many containers on one server



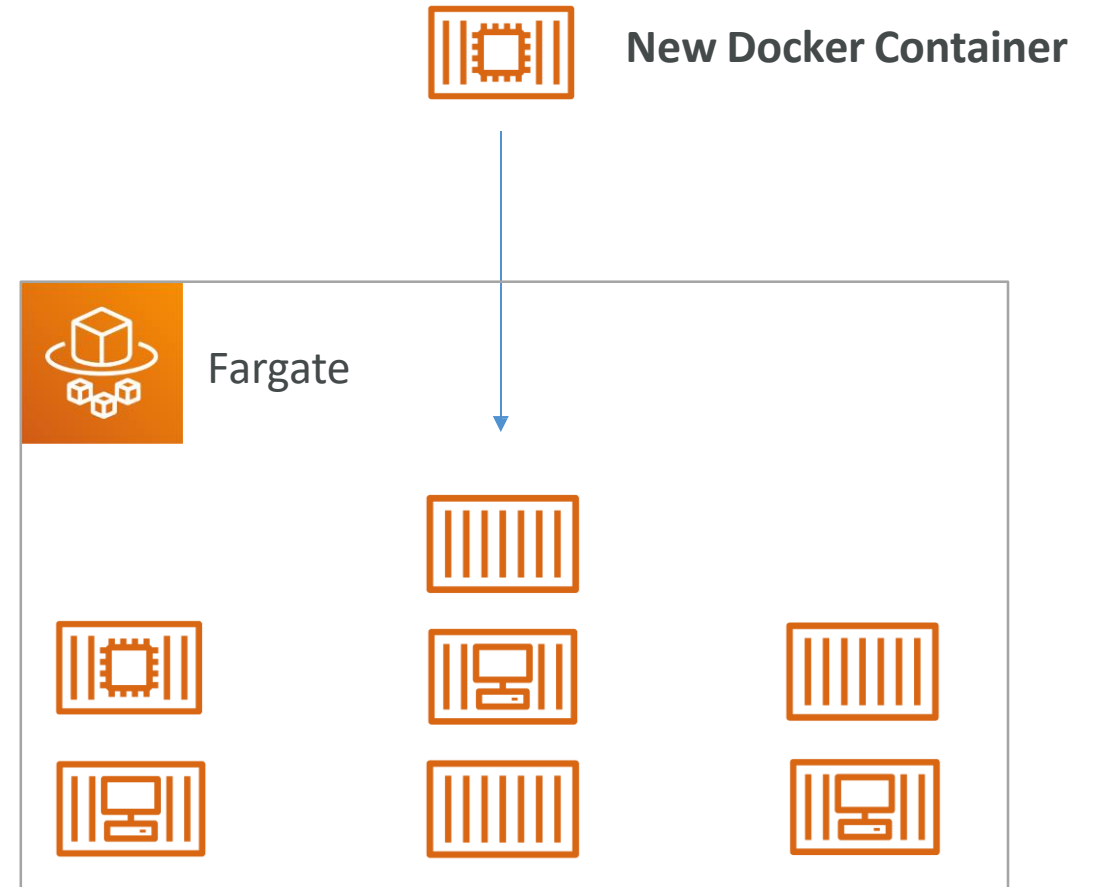
ECS

- ECS = Elastic Container Service
- Launch Docker containers on AWS
- You must provision & maintain the infrastructure (the EC2 instances)
- AWS takes care of starting / stopping containers
- Has integrations with the Application Load Balancer



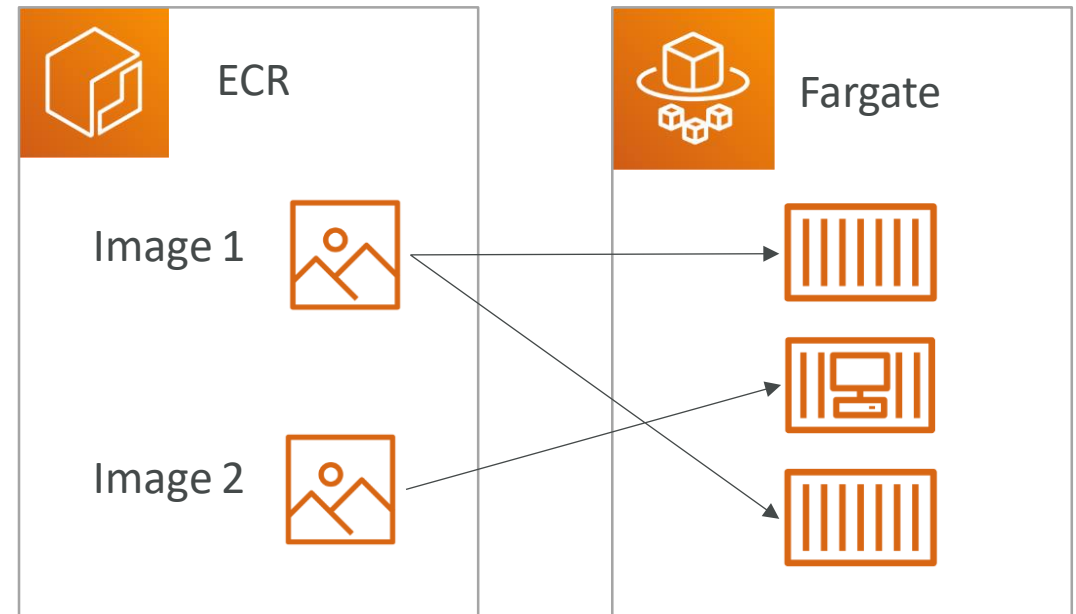
Fargate

- Launch Docker containers on AWS
- You do not provision the infrastructure (no EC2 instances to manage) – simpler!
- Serverless offering
- AWS just runs containers for you based on the CPU / RAM you need



ECR

- Elastic Container Registry
- Private Docker Registry on AWS
- This is where you store your Docker images so they can be run by ECS or Fargate



What's serverless?

- Serverless is a new paradigm in which the developers don't have to manage servers anymore...
- They just deploy code
- They just deploy... functions !
- Initially... Serverless == FaaS (Function as a Service)
- Serverless was pioneered by AWS Lambda but now also includes anything that's managed: “databases, messaging, storage, etc.”
- Serverless does not mean there are no servers...
it means you just don't manage / provision / see them

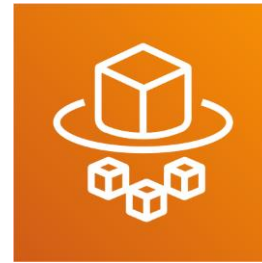
So far in this course...



Amazon S3



DynamoDB

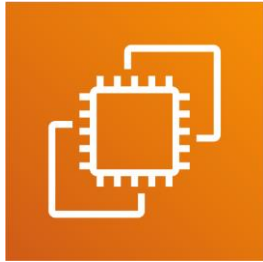


Fargate



Lambda

Why AWS Lambda



Amazon EC2


- Virtual Servers in the Cloud
- Limited by RAM and CPU
- Continuously running
- Scaling means intervention to add / remove servers



Amazon Lambda

- Virtual functions – no servers to manage!
- Limited by time - short executions
- Run on-demand
- Scaling is automated!

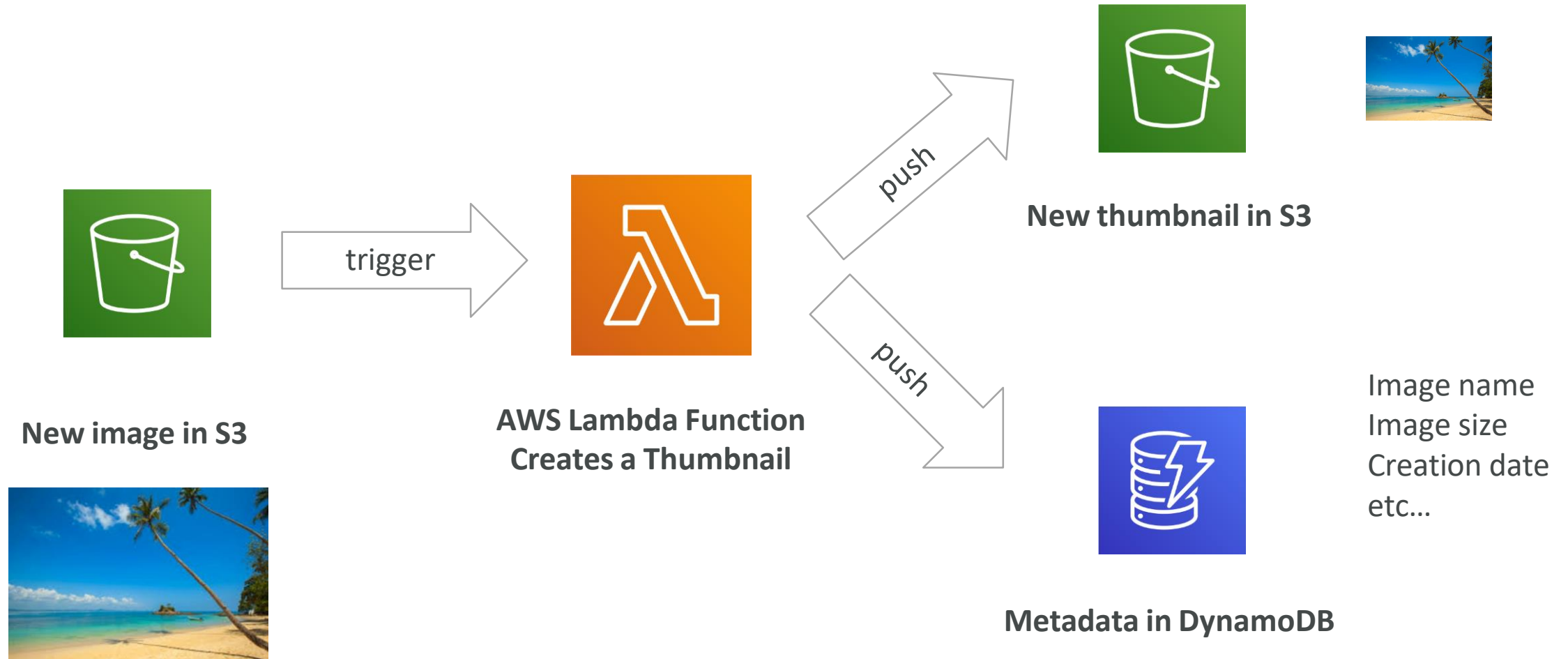
Benefits of AWS Lambda

- Easy Pricing:
 - Pay per request and compute time
 - Free tier of 1,000,000 AWS Lambda requests and 400,000 GBs of compute time
 - Integrated with the whole AWS suite of services
 - Event-Driven: functions get invoked by AWS when needed
 - Integrated with many programming languages
 - Easy monitoring through AWS CloudWatch
 - Easy to get more resources per functions (up to 10GB of RAM!)
 - Increasing RAM will also improve CPU and network!
- 

AWS Lambda language support

- Node.js (JavaScript)
- Python
- Java (Java 8 compatible)
- C# (.NET Core)
- Golang
- C# / Powershell
- Ruby
- Custom Runtime API (community supported, example Rust)
- Lambda Container Image
 - The container image must implement the Lambda Runtime API
 - ECS / Fargate is preferred for running arbitrary Docker images

Example: Serverless Thumbnail creation



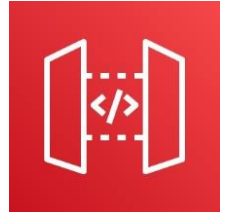
Example: Serverless CRON Job



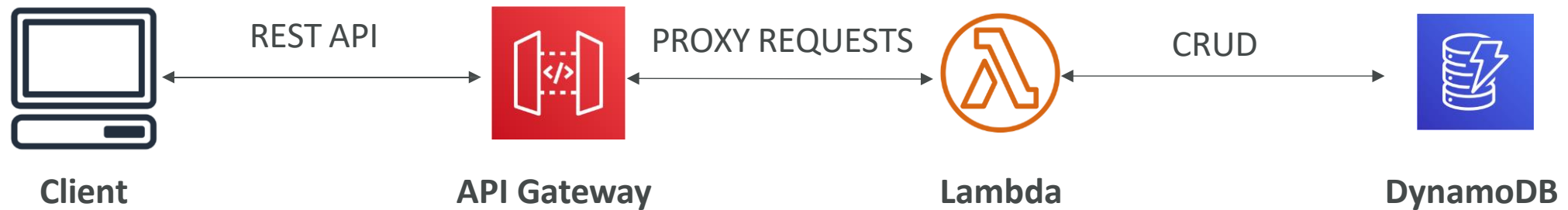
AWS Lambda Pricing: example

- You can find overall pricing information here:
<https://aws.amazon.com/lambda/pricing/>
- Pay per calls:
 - First 1,000,000 requests are free
 - \$0.20 per 1 million requests thereafter (\$0.0000002 per request)
- Pay per duration: (in increment of 1 ms)
 - 400,000 GB-seconds of compute time per month for FREE
 - == 400,000 seconds if function is 1 GB RAM
 - == 3,200,000 seconds if function is 128 MB RAM
 - After that \$1.00 for 600,000 GB-seconds
- It is usually very cheap to run AWS Lambda so it's very popular

Amazon API Gateway



- Example: building a serverless API



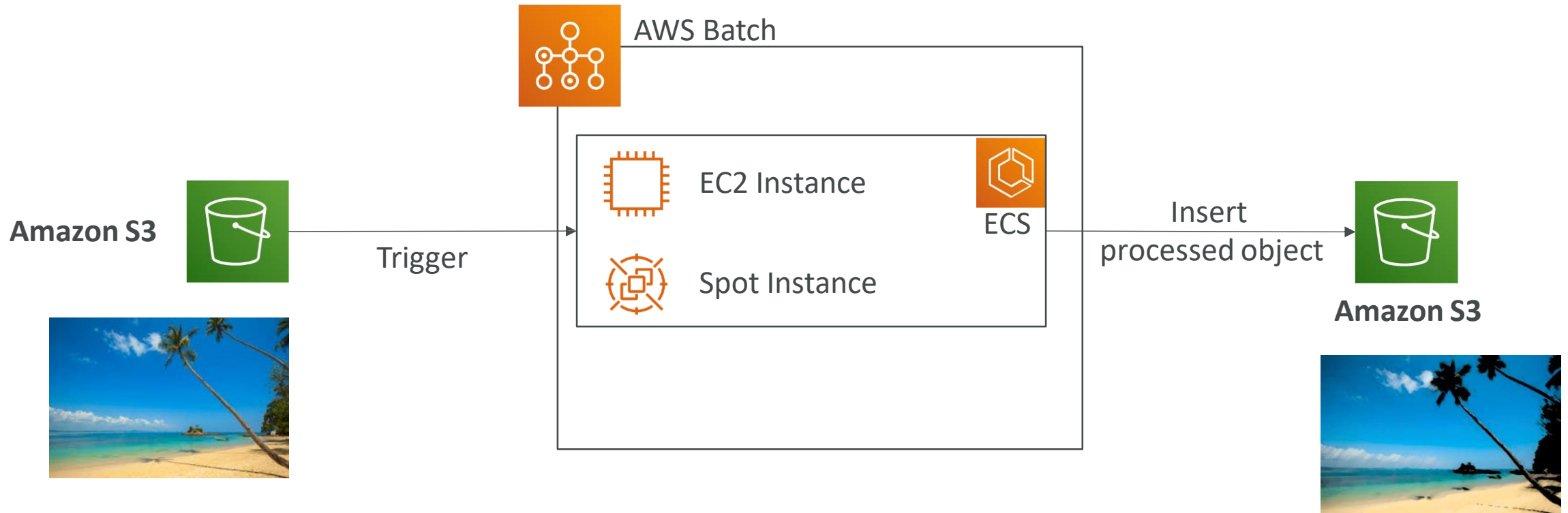
- Fully managed service for developers to easily create, publish, maintain, monitor, and secure APIs
- Serverless and scalable
- Supports RESTful APIs and WebSocket APIs
- Support for security, user authentication, API throttling, API keys, monitoring...

AWS Batch



- Fully managed batch processing at any scale
- Efficiently run 100,000s of computing batch jobs on AWS
- A “batch” job is a job with a start and an end (opposed to continuous)
- Batch will dynamically launch EC2 instances or Spot Instances
- AWS Batch provisions the right amount of compute / memory
- You submit or schedule batch jobs and AWS Batch does the rest!
- Batch jobs are defined as Docker images and run on ECS
- Helpful for cost optimizations and focusing less on the infrastructure

AWS Batch – Simplified Example



Batch vs Lambda

- Lambda:

- Time limit
- Limited runtimes
- Limited temporary disk space
- Serverless



- Batch:

- No time limit
- Any runtime as long as it's packaged as a Docker image
- Rely on EBS / instance store for disk space
- Relies on EC2 (can be managed by AWS)



Amazon Lightsail



- Virtual servers, storage, databases, and networking
- Low & predictable pricing
- Simpler alternative to using EC2, RDS, ELB, EBS, Route 53...
- Great for people with little cloud experience!
- Can setup notifications and monitoring of your Lightsail resources
- Use cases:
 - Simple web applications (has templates for LAMP, Nginx, MEAN, Node.js...)
 - Websites (templates for WordPress, Magento, Plesk, Joomla)
 - Dev / Test environment
- Has high availability but no auto-scaling, limited AWS integrations

Other Compute - Summary

- Docker: container technology to run applications
- ECS: run Docker containers on EC2 instances
- Fargate:
 - Run Docker containers without provisioning the infrastructure
 - Serverless offering (no EC2 instances)
- ECR: Private Docker Images Repository
- Batch: run batch jobs on AWS across managed EC2 instances
- Lightsail: predictable & low pricing for simple application & DB stacks

Lambda Summary

- Lambda is Serverless, Function as a Service, seamless scaling, reactive
- Lambda Billing:
 - By the time run x by the RAM provisioned
 - By the number of invocations
- Language Support: many programming languages except (arbitrary) Docker
- Invocation time: up to 15 minutes
- Use cases:
 - Create Thumbnails for images uploaded onto S3
 - Run a Serverless cron job
- API Gateway: expose Lambda functions as HTTP API

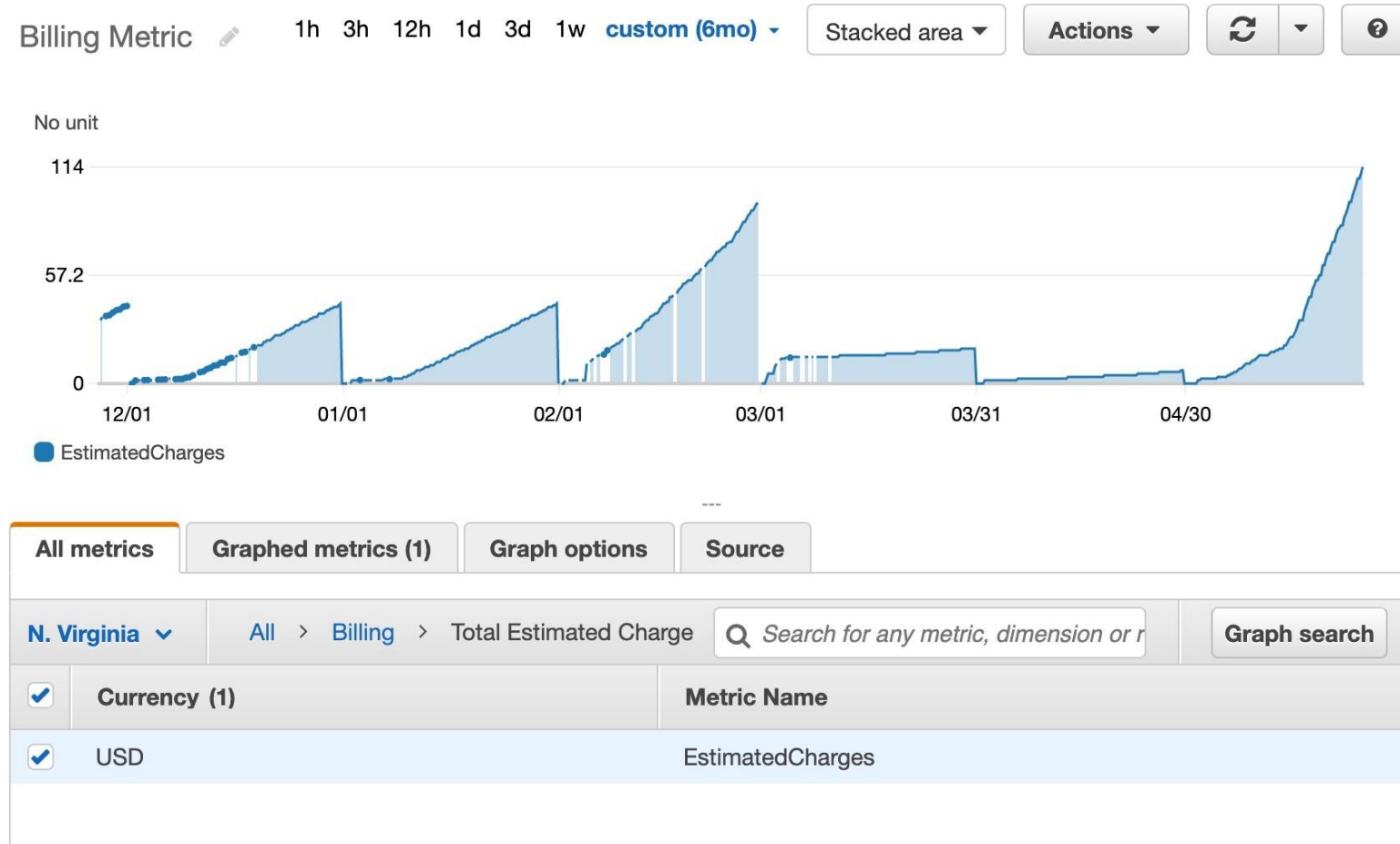
Cloud Monitoring Section

Amazon CloudWatch Metrics




- CloudWatch provides metrics for every services in AWS
- Metric is a variable to monitor (CPUUtilization, NetworkIn...)
- Metrics have timestamps
- Can create CloudWatch dashboards of metrics

Example: CloudWatch Billing metric (us-east-1)



Important Metrics

- EC2 instances: CPU Utilization, Status Checks, Network (not RAM)
 - Default metrics every 5 minutes
 - Option for Detailed Monitoring (\$\$\$): metrics every 1 minute
 - EBS volumes: Disk Read/Writes
 - S3 buckets: BucketSizeBytes, NumberOfObjects, AllRequests
 - Billing: Total Estimated Charge (only in us-east-1)
 - Service Limits: how much you've been using a service API
 - Custom metrics: push your own metrics
- 

Amazon CloudWatch Alarms



- Alarms are used to trigger notifications for any metric
- Alarms actions...
 - Auto Scaling: increase or decrease EC2 instances “desired” count
 - EC2 Actions: stop, terminate, reboot or recover an EC2 instance
 - SNS notifications: send a notification into an SNS topic
- Various options (sampling, %, max, min, etc...)
- Can choose the period on which to evaluate an alarm
- Example: create a billing alarm on the CloudWatch Billing metric
- Alarm States: OK, INSUFFICIENT_DATA, ALARM

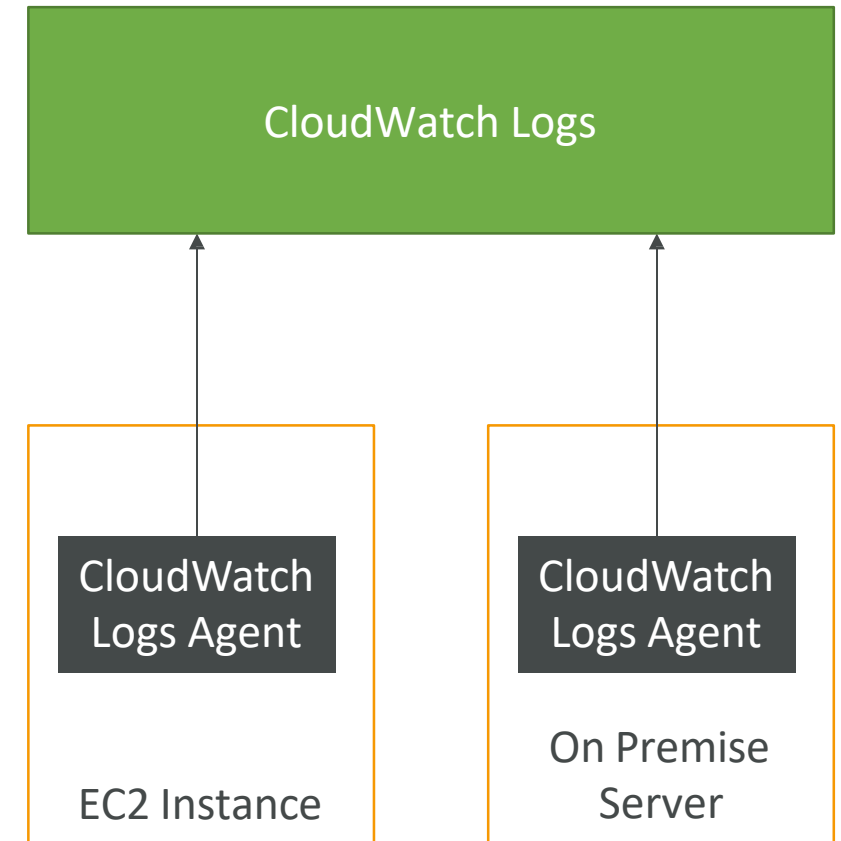
Amazon CloudWatch Logs



- CloudWatch Logs can collect log from:
 - Elastic Beanstalk: collection of logs from application
 - ECS: collection from containers
 - AWS Lambda: collection from function logs
 - CloudTrail based on filter
 - CloudWatch log agents: on EC2 machines or on-premises servers
 - Route53: Log DNS queries
- Enables real-time monitoring of logs
- Adjustable CloudWatch Logs retention

CloudWatch Logs for EC2

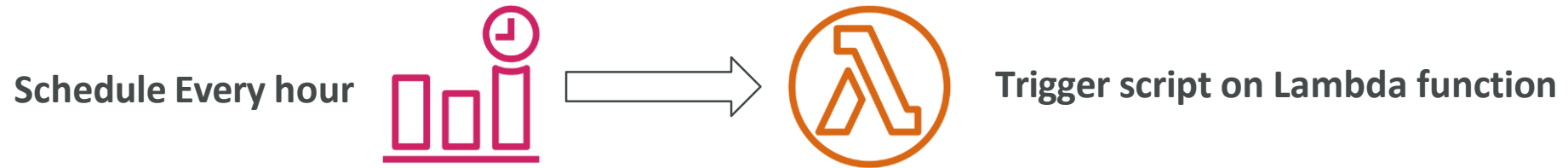
- By default, no logs from your EC2 instance will go to CloudWatch
- You need to run a CloudWatch agent on EC2 to push the log files you want
- Make sure IAM permissions are correct
- The CloudWatch log agent can be setup on-premises too



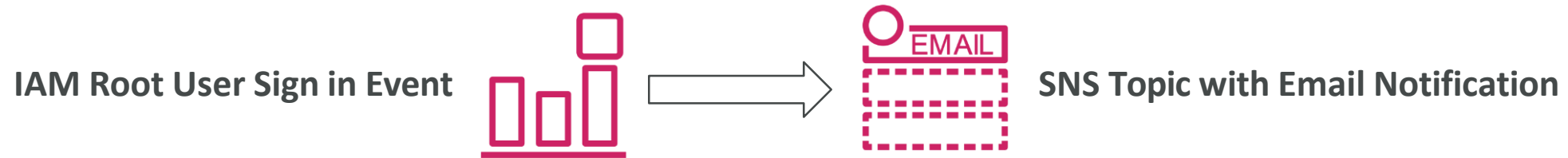
Amazon CloudWatch Events



- Schedule: Cron jobs (scheduled scripts)



- Event Pattern: Event rules to react to a service doing something



- Trigger Lambda functions, send SQS/SNS messages...

Amazon EventBridge



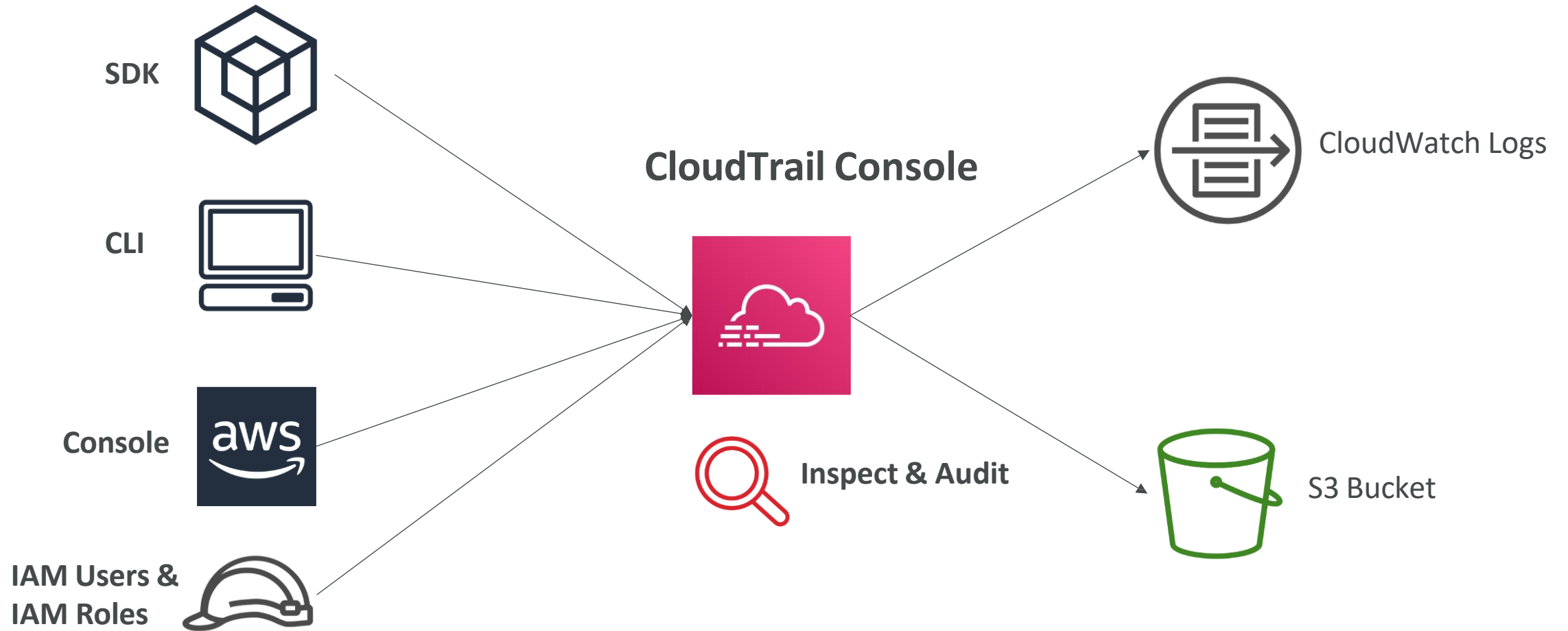
- EventBridge is the next evolution of CloudWatch Events
- Default event bus: generated by AWS services (CloudWatch Events)
- Partner event bus: receive events from SaaS service or applications (Zendesk, DataDog, Segment, Auth0...)
- Custom Event buses: for your own applications
- Schema Registry: model event schema
- EventBridge has a different name to mark the new capabilities
- The CloudWatch Events name will be replaced with EventBridge

AWS CloudTrail

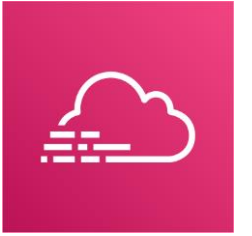


- Provides governance, compliance and audit for your AWS Account
- CloudTrail is enabled by default!
- Get an history of events / API calls made within your AWS Account by:
 - Console
 - SDK
 - CLI
 - AWS Services
- Can put logs from CloudTrail into CloudWatch Logs or S3
- A trail can be applied to All Regions (default) or a single Region.
- If a resource is deleted in AWS, investigate CloudTrail first!

CloudTrail Diagram



CloudTrail Events

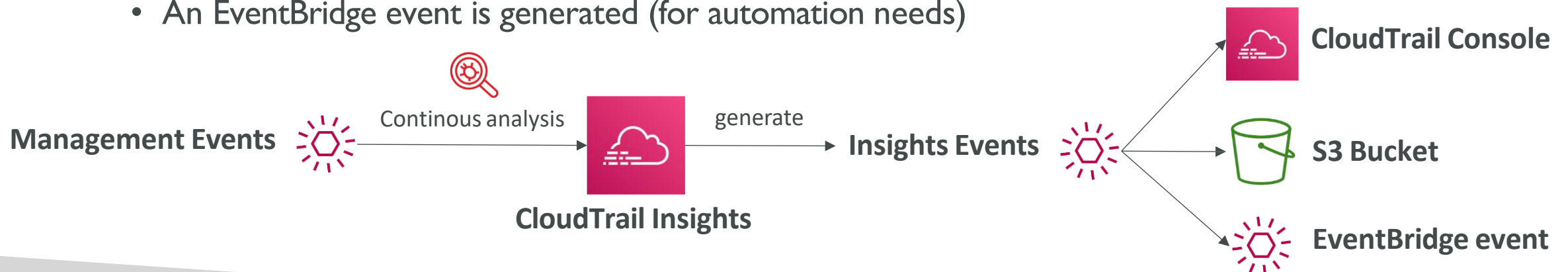


- Management Events:
 - Operations that are performed on resources in your AWS account
 - Examples:
 - Configuring security (IAM `AttachRolePolicy`)
 - Configuring rules for routing data (Amazon EC2 `CreateSubnet`)
 - Setting up logging (AWS CloudTrail `CreateTrail`)
 - By default, trails are configured to log management events.
 - Can separate Read Events (that don't modify resources) from Write Events (that may modify resources)
- Data Events:
 - By default, data events are not logged (because high volume operations)
 - Amazon S3 object-level activity (ex: `GetObject`, `DeleteObject`, `PutObject`): can separate Read and Write Events
 - AWS Lambda function execution activity (the `Invoke` API)
- CloudTrail Insights Events:
 - See next slide J

CloudTrail Insights

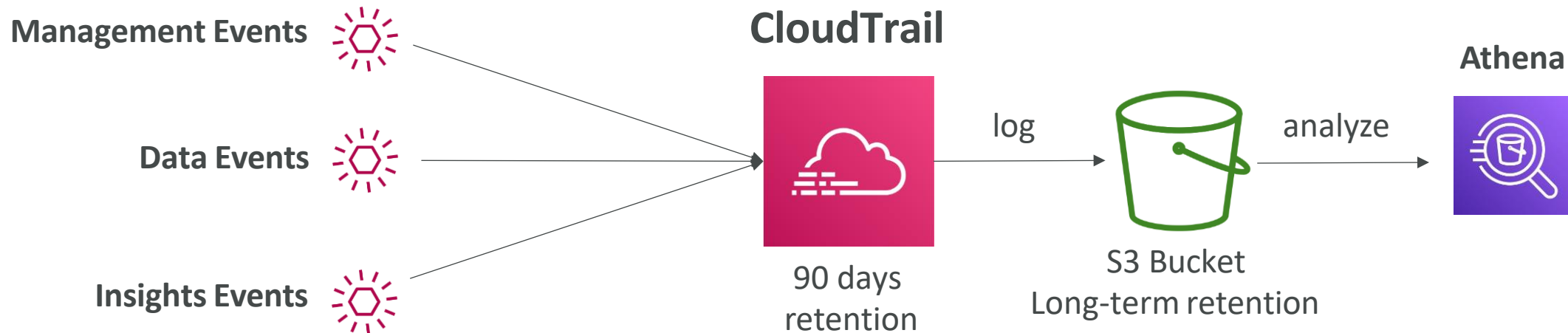


- Enable CloudTrail Insights to detect unusual activity in your account:
 - inaccurate resource provisioning
 - hitting service limits
 - Bursts of AWS IAM actions
 - Gaps in periodic maintenance activity
- CloudTrail Insights analyzes normal management events to create a baseline
- And then continuously analyzes write events to detect unusual patterns
 - Anomalies appear in the CloudTrail console
 - Event is sent to Amazon S3
 - An EventBridge event is generated (for automation needs)



CloudTrail Events Retention

- Events are stored for 90 days in CloudTrail
- To keep events beyond this period, log them to S3 and use Athena



AWS X-Ray



- Debugging in Production, the good old way:
 - Test locally
 - Add log statements everywhere
 - Re-deploy in production
- Log formats differ across applications and log analysis is hard.
- Debugging: one big monolith “easy”, distributed services “hard”
- No common views of your entire architecture
- Enter... AWS X-Ray!

AWS X-Ray

Visual analysis of our applications



AWS X-Ray advantages

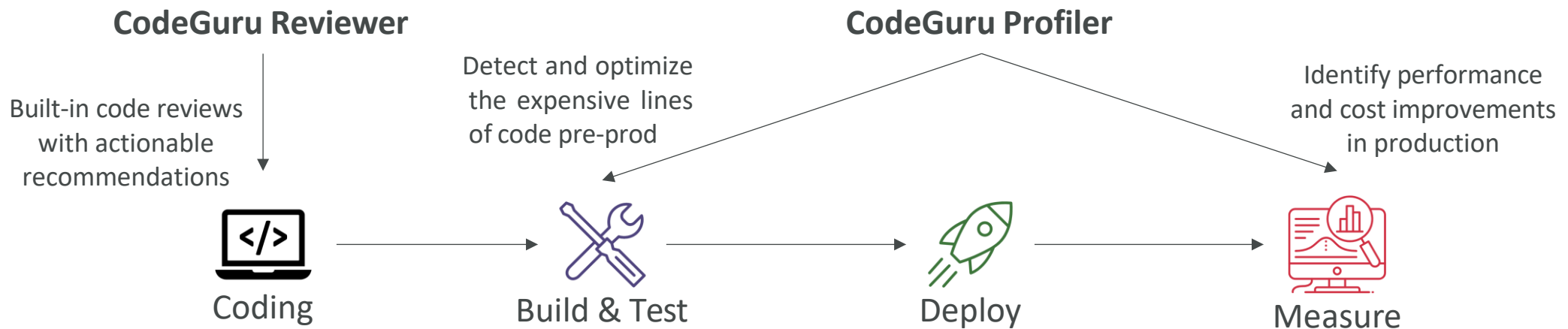


- Troubleshooting performance (bottlenecks)
- Understand dependencies in a microservice architecture
- Pinpoint service issues
- Review request behavior
- Find errors and exceptions
- Are we meeting time SLA?
- Where I am throttled?
- Identify users that are impacted

Amazon CodeGuru



- An ML-powered service for automated code reviews and application performance recommendations
- Provides two functionalities
 - CodeGuru Reviewer: automated code reviews for static code analysis (development)
 - CodeGuru Profiler: visibility/recommendations about application performance during runtime (production)



Amazon CodeGuru Reviewer

- Identify critical issues, security vulnerabilities, and hard-to-find bugs
- Example: common coding best practices, resource leaks, security detection, input validation
- Uses Machine Learning and automated reasoning
- Hard-learned lessons across millions of code reviews on 1000s of open-source and Amazon repositories
- Supports Java and Python
- Integrates with GitHub, Bitbucket, and AWS CodeCommit

The screenshot displays the Amazon CodeGuru Reviewer console interface. At the top, the breadcrumb navigation shows 'CodeGuru > Code reviews > mw2tsa56o0000000'. The main heading is 'RepositoryAnalysis-amazon-codeguru-reviewer-sample-app-master-mw2tsa56o0000000'. Below this, the 'Details' section provides information about the review: Status is 'Completed' (indicated by a green checkmark), Details state 'CodeGuru Reviewer successfully finished reviewing the source code.', and the ARN is 'arn:aws:codeguru-reviewer:us-west-2:033467977803:code-review:RepositoryAnalysis-amazon-codeguru-reviewer-sample-app-master-mw2tsa56o0000000'. Recommendations are listed as 4, with 80 metered lines of code. The review was created on 10 Nov 2020 at 08:08:47 AM GMT-0800 and last updated on 10 Nov 2020 at 08:11:44 AM GMT-0800. The repository is 'amazon-codeguru-reviewer-sample-app' and the branch is 'master'. The 'Recommendations (4)' section shows three items, each with a search bar and a 'Was this helpful?' feedback button. The first recommendation is for 'EventHandler.java Line: 79', suggesting the use of 'waiters' for efficiency. The second is for 'EventHandler.java Line: 100', suggesting pagination. The third is for 'EventHandler.java Line: 100', suggesting the use of the revised 'List Objects API'.

CodeGuru > Code reviews > mw2tsa56o0000000

RepositoryAnalysis-amazon-codeguru-reviewer-sample-app-master-mw2tsa56o0000000

Details

Status	Recommendations	Type
Completed	4	RepositoryAnalysis
Details	Metered lines of code	Provider
CodeGuru Reviewer successfully finished reviewing the source code.	80	GitHub
ARN	Time created	Repository
arn:aws:codeguru-reviewer:us-west-2:033467977803:code-review:RepositoryAnalysis-amazon-codeguru-reviewer-sample-app-master-mw2tsa56o0000000	10 Nov 2020 08:08:47 AM GMT-0800	amazon-codeguru-reviewer-sample-app
	Last updated	Branch name
	10 Nov 2020 08:11:44 AM GMT-0800	master

Recommendations (4)

Search recommendations

EventHandler.java Line: 79

This code appears to be waiting for a resource before it runs. You could use the waiters feature to help improve efficiency. Consider using ObjectExists or ObjectNotExists. For more information, see <https://aws.amazon.com/blogs/developer/waiters-in-the-aws-sdk-for-java/>

Was this helpful?

EventHandler.java Line: 100

This code might not produce accurate results if the operation returns paginated results instead of all results. Consider adding another call to check for additional results.

Was this helpful?

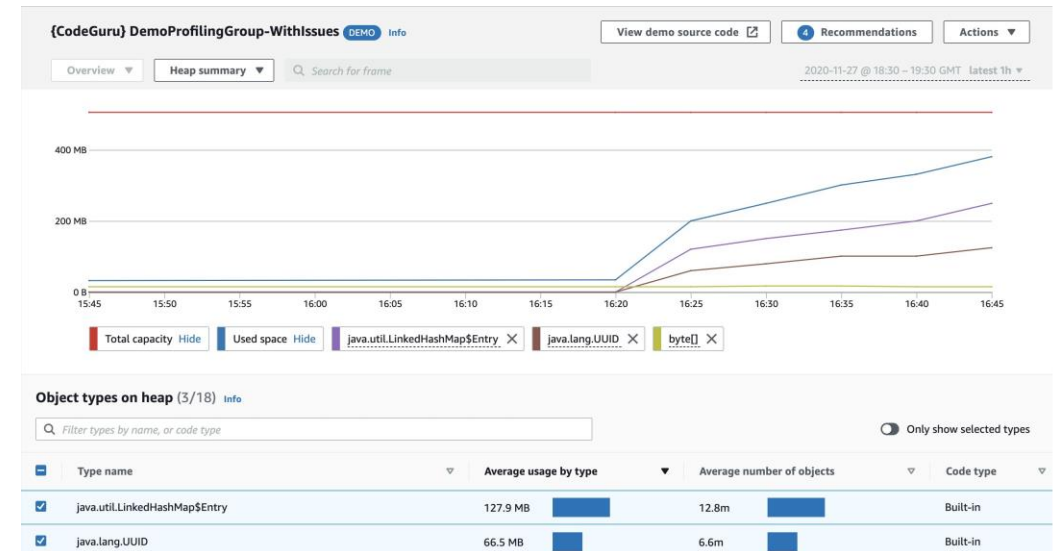
EventHandler.java Line: 100

This code uses an outdated API. ListObjectsV2 is the revised List Objects API, and we recommend you use this revised API for new application developments.

Was this helpful?

Amazon CodeGuru Profiler

- Helps understand the runtime behavior of your application
- Example: identify if your application is consuming excessive CPU capacity on a logging routine
- Features:
 - Identify and remove code inefficiencies
 - Improve application performance (e.g., reduce CPU utilization)
 - Decrease compute costs
 - Provides heap summary (identify which objects using up memory)
 - Anomaly Detection
- Support applications running on AWS or on-premise
- Minimal overhead on application



AWS Status - Service Health Dashboard



- Shows all regions, all services health
 - Shows historical information for each day
 - Has an RSS feed you can subscribe to
-
- <https://status.aws.amazon.com/>



[Amazon Web Services](#) » Service Health Dashboard

Get a personalized view of AWS service health

[Open the Personal Health Dashboard](#)

Current Status - May 26, 2020 PDT

Amazon Web Services publishes our most up-to-the-minute information on service availability in the table below. Check back here any time to get current status information, or subscribe to an RSS feed to be notified of interruptions to each individual service. If you are experiencing a real-time, operational issue with one of our services that is not described below, please inform us by clicking on the "Contact Us" link to submit a service issue report. All dates and times are Pacific Time (PST/PDT).

North America			South America	Europe	Africa	Asia Pacific	Middle East	Contact Us
Recent Events		Details	RSS					
✓ No recent events.								
Remaining Services		Details	RSS					
✓	Alexa for Business (N. Virginia)	Service is operating normally						
✓	Amazon API Gateway (Montreal)	Service is operating normally						
✓	Amazon API Gateway (N. California)	Service is operating normally						
✓	Amazon API Gateway (N. Virginia)	Service is operating normally						
✓	Amazon API Gateway (Ohio)	Service is operating normally						
✓	Amazon API Gateway (Oregon)	Service is operating normally						
✓	Amazon AppStream 2.0 (N. Virginia)	Service is operating normally						
✓	Amazon AppStream 2.0 (Oregon)	Service is operating normally						
✓	Amazon Athena (Montreal)	Service is operating normally						

AWS Personal Health Dashboard

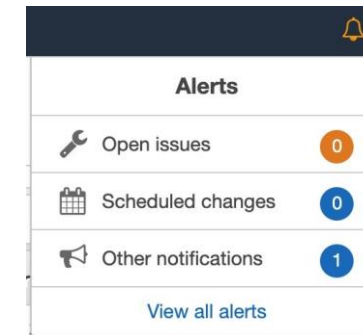





- AWS Personal Health Dashboard provides alerts and remediation guidance when AWS is experiencing events that may impact you.
- While the Service Health Dashboard displays the general status of AWS services, Personal Health Dashboard gives you a personalized view into the performance and availability of the AWS services underlying your AWS resources.
- The dashboard displays relevant and timely information to help you manage events in progress and provides proactive notification to help you plan for scheduled activities.

AWS Personal Health Dashboard










- Global service <https://phd.aws.amazon.com/>
- Shows how AWS outages directly impact you & your AWS resources
- Alert, remediation, proactive, scheduled activities



Alerts		
	Open issues	0
	Scheduled changes	0
	Other notifications	1
View all alerts		

Event log



	Event	Status	Region/AZ ⓘ	Start time	Last update time	Affected resources	Event category
<input type="radio"/>	ElasticContainerRegistry operational issue	Closed	us-west-2	May 22, 2020 at 11:48:49 PM U...	May 22, 2020 at 11:49:31 PM U...	-	 Issue
<input type="radio"/>	CodeBuild operational notification	-	-	May 21, 2020 at 11:20:00 PM U...	May 21, 2020 at 11:35:26 PM U...	1 entity	 Notification
<input type="radio"/>	ElasticsearchService operational issue	Closed	us-east-1	May 21, 2020 at 3:44:30 PM UT...	May 21, 2020 at 4:38:20 PM UT...	-	 Issue
<input type="radio"/>	Batch operational issue	Closed	us-west-1	May 10, 2020 at 3:38:49 AM UT...	May 10, 2020 at 5:55:46 AM UT...	-	 Issue
<input type="radio"/>	ElasticContainerService operational issue	Closed	us-west-1	May 10, 2020 at 3:31:30 AM UT...	May 10, 2020 at 5:52:25 AM UT...	-	 Issue
<input type="radio"/>	CloudFormation operational issue	Closed	us-west-2	April 30, 2020 at 9:47:10 PM UT...	April 30, 2020 at 11:11:31 PM U...	-	 Issue
<input type="radio"/>	CloudFront operational issue	Closed	-	April 21, 2020 at 11:57:30 PM U...	April 22, 2020 at 12:28:15 AM U...	-	 Issue

Monitoring Summary

- CloudWatch:
 - Metrics: monitor the performance of AWS services and billing metrics
 - Alarms: automate notification, perform EC2 action, notify to SNS based on metric
 - Logs: collect log files from EC2 instances, servers, Lambda functions...
 - Events (or EventBridge): react to events in AWS, or trigger a rule on a schedule
- CloudTrail: audit API calls made within your AWS account
- CloudTrail Insights: automated analysis of your CloudTrail Events
- X-Ray: trace requests made through your distributed applications
- Service Health Dashboard: status of all AWS services across all regions
- Personal Health Dashboard: AWS events that impact your infrastructure
- Amazon CodeGuru: automated code reviews and application performance recommendations