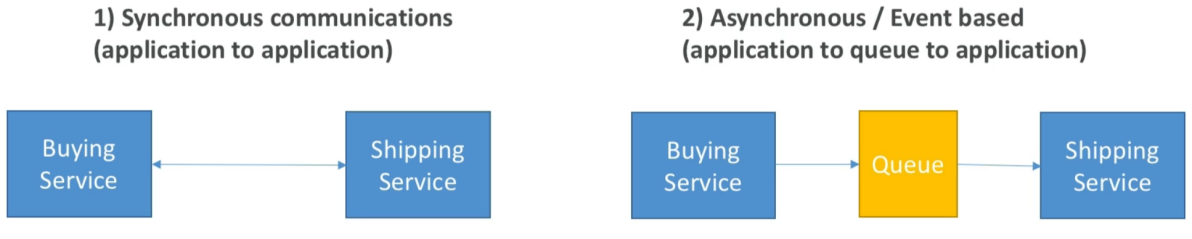
SAA-6

AWS Integration & Messaging

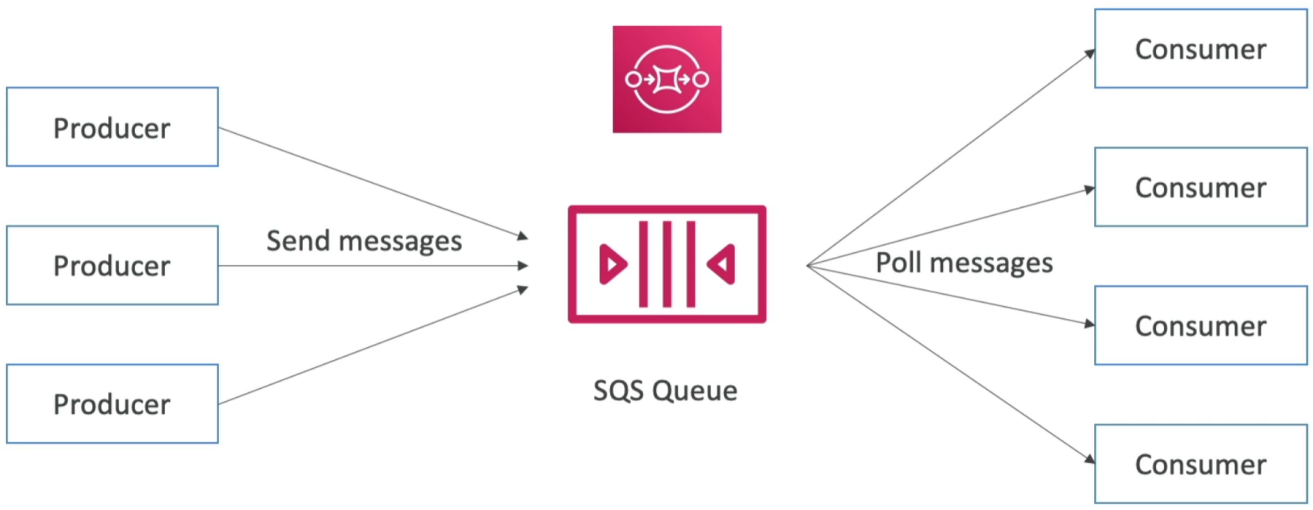
Section Introduction

- When we start deploying multiple applications, they will inevitably need to communicate with one another
- There are two patterns of application communication



- Syschronous between applications can be problematic if there are sudden spikes of traffic
- What if you need to suddenly encode 1000 videos but usually it's 10?
- In that case, it's better to **decouple** your applications,
  - using SQS: queue model
  - using SNS: pub/sub model
  - using Kinesis: real-time streaming model
- These services can scale independently from our application

Amazon SQS
Waht's a queue?



Amazon SQS - Standard Queue

- Oldest offering (over 10 years old)
- Fully managed service,used to **decouple applications**
- Attributes:
  - Unlimited throughput, unlimited number of messages in queue
  - Default retention of messages: 4 days, maximum of 14 days
  - Low latency (< 10 ms on publish and receive)
  - Limitation of 256KB per message sent

- Can have duplicate messages (at least once delivery, occasionally)
- Can have out of order messages (best effort ordering)

SQS - Producing Messages

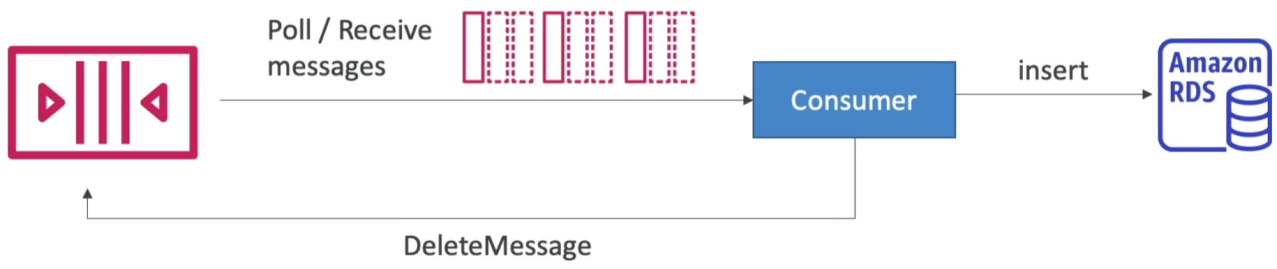
- Produced to SQS using the SDK (SendMessage API)
- The message is **persisted** in SQS until a consummer deletes it
- Message retention: default 4 days, up to 14 days

- Example: send an order to be processed

    - Order id
    - Customer id
    - Any attributes you want
- SQS standard: unlimited throughput

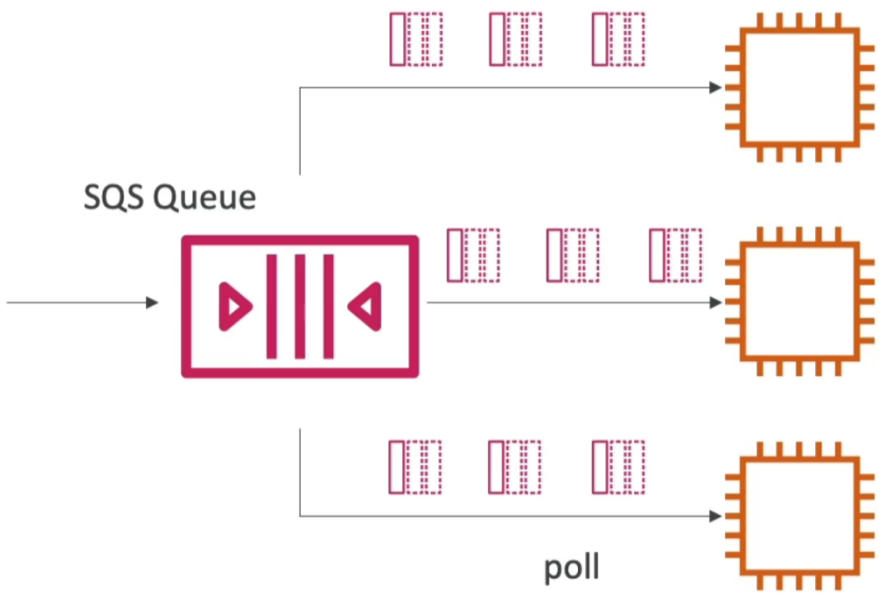
SQS - Consuming Messages

- Consummers (running on EC2 instances, servers, or AWS Lambda)...
- Poll SQS for messages (receive up to 10 messages at a time)
- Process the messages (example: insert the message into an RDS database)
- Delete the messages using the DeleteMessage API

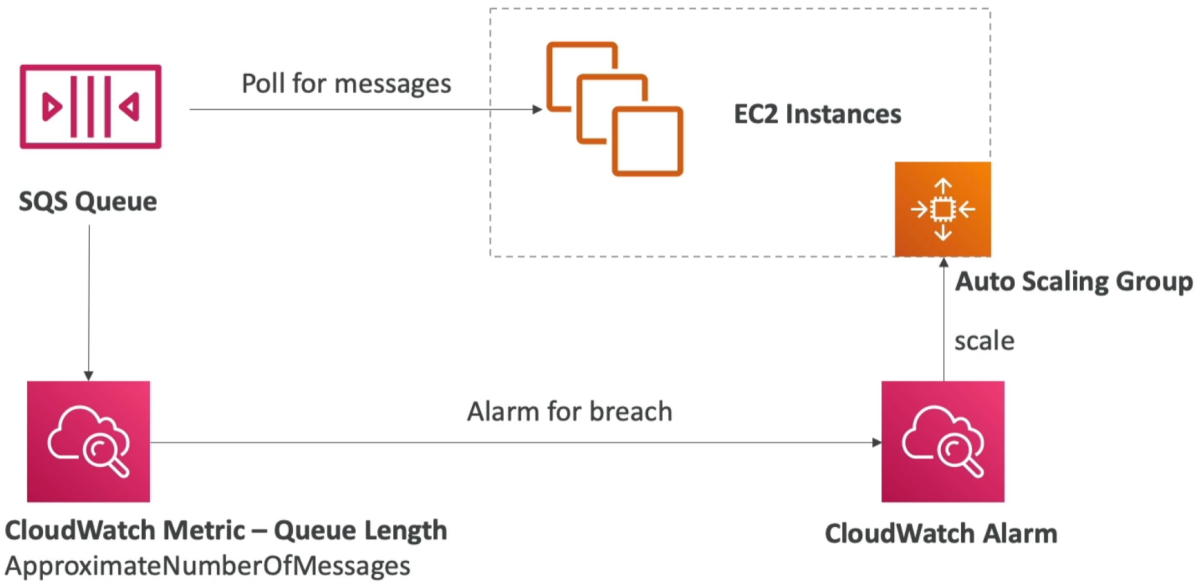


SQS - Multiple EC2 Instances Consumers

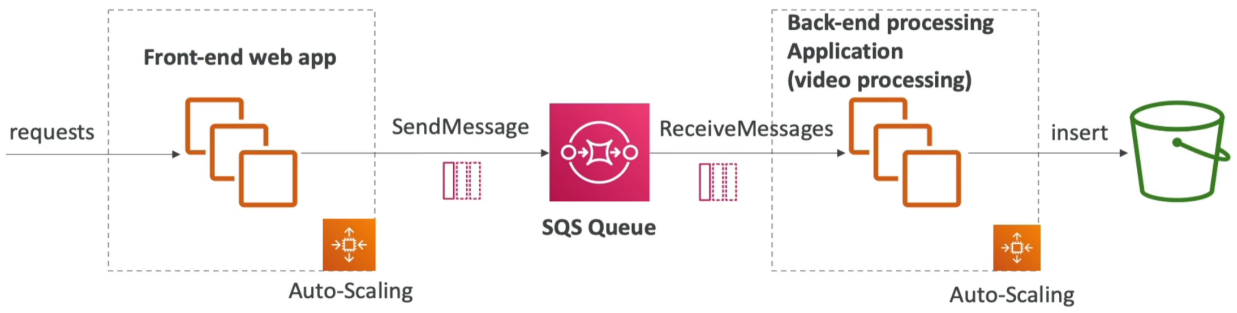
- Consumers receive and process messages in parallel
- At least once delivery
- Best-effort message ordering
- Consumers delete messages after processing them
- We can scale consumers horizontally to improve throughput of processing



SQS with Auto Scaling Group (ASG)



SQS to decouple between application tiers



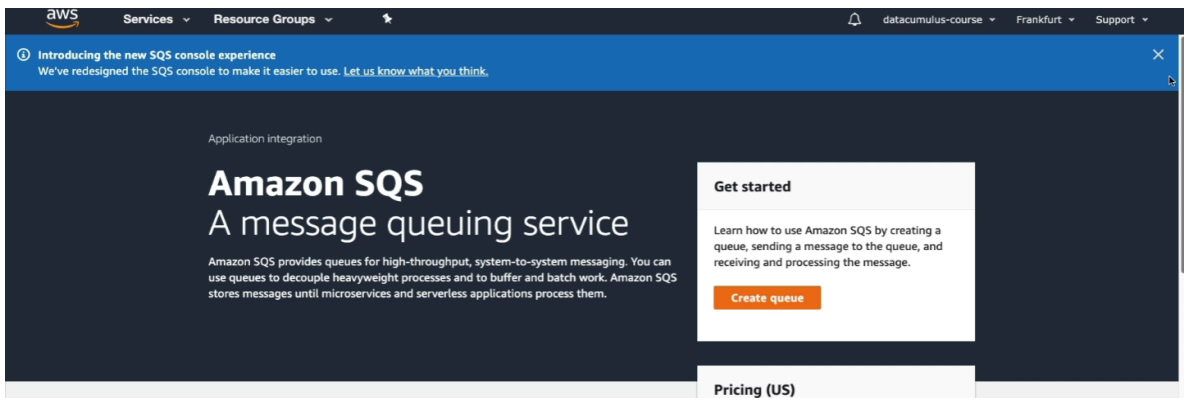
Amazon SQS - Security

- **Encryption:**
  - In-flight encryption using HTTPS API
  - At-rest encryption using KMS keys
  - Client-side encryption if the client wants to perform encryption/decryption itself

- **Access Controls:** IAM policies to regulate access to the SQS API

- **SQS Access Policies** (similar to S3 bucket policies)
  - Useful for cross-account access to SQS queues
  - Useful for allowing other services (SNS, S3 ...) to write to an SQS queue

Hands On



Create queue

Details

Type
Choose the queue type for your application or cloud infrastructure.

You can't change the queue type after you create a queue.

Standard Info
At-least-once delivery, message ordering isn't preserved
- At-least once delivery
- Best-effort ordering

FIFO Info
First-in-first-out delivery, message ordering is preserved
- First-in-first-out delivery
- Exactly-once processing

Name

MyQueue

A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (_).

Configuration

Set the maximum message size, visibility to other consumers, and message retention. Info

Visibility timeout Info
30 Seconds
Should be between 0 seconds and 12 hours.

Message retention period Info
4 Days
Should be between 1 minute and 14 days.

Delivery delay Info
0 Seconds
Should be between 0 seconds and 15 minutes.

Maximum message size Info
256 KB
Should be between 1 KB and 256 KB.

Receive message wait time Info
0 Seconds
Should be between 0 and 20 seconds.

Access policy

Define who can access your queue. Info

Choose method

Basic
Use simple criteria to define a basic access policy.

Advanced
Use a JSON object to define an advanced access policy.

Define who can send messages to the queue
- Only the queue owner
  Only the owner of the queue can send messages to the queue.
- Only the specified AWS accounts, IAM users and roles
  Only the specified AWS account IDs, IAM users and roles can send messages to the queue.

Define who can receive messages from the queue
- Only the queue owner
  Only the owner of the queue can receive messages from the queue.
- Only the specified AWS accounts, IAM users and roles
  Only the specified AWS account IDs, IAM users and roles can receive messages from the queue.

JSON (read-only)

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "387124123361"
      },
      "Action": [
        "SQS:*"
      ],
      "Resource": "arn:aws:sqs:eu-central-
1:387124123361:DemoQueue"
    }
```

Encryption - *Optional*

Amazon SQS provides in-transit encryption by default. To add at-rest encryption to your queue, enable server-side encryption. Info

Server-side encryption
- Disabled
- Enabled

Choose CMK
- Choose a CMK alias
- Enter the CMK alias

Customer master key Info
(Default) alias/aws/sqs

Description
Default master key that protects my SQS messages when no other key is defined

Account
387124123361

CMK ARN
arn:aws:kms:eu-central-1:387124123361:key/c7582c86-f57a-42c5-8c91-ca9617d7fa56

Data key reuse period
5 Minutes
Should be between 1 minute and 24 hours.

▶ **Dead-letter queue** - *Optional*

Send undeliverable messages to a dead-letter queue. Info

▶ **Tags** - *Optional*

A tag is a label assigned to an AWS resource. Use tags to search and filter your resources or track your AWS costs. Learn more

Cancel | Create queue

DemoQueue

Edit | Delete | Purge | Send and receive messages

Details Info

Name	Type	ARN
DemoQueue	Standard	arn:aws:sqs:eu-central-1:387124123361:DemoQueue
Encryption	**URL**	**Dead-letter queue**
Enabled	https://sqs.eu-central-1.amazonaws.com/387124123361/DemoQueue	Disabled

▶ More

Send and receive messages

Send messages to and receive messages from a queue.

Send message Info — Clear content | Send message

✓ Your message has been sent and is ready to be received. — View details

Message body

Enter the message to send to the queue.

hello world!

Delivery delay Info

0 Seconds

Should be between 0 seconds and 15 minutes.

▶ Message attributes - *Optional* Info

Receive messages Info — Edit poll settings | Stop polling | Poll for messages

Messages available	Polling duration	Maximum message count	Polling progress
1	30	10	0% (0 receives/second)

Messages (0) — View details | Delete

Search messages

1

Receive messages Info — Edit poll settings | Stop polling | Poll for messages

Messages available	Polling duration	Maximum message count	Polling progress
1	30	10	0% (0 receives/second)

Messages (0) — View details | Delete

Search messages

1

ID	Sent	Size	Receive count

No messages. To view messages in the queue, poll for messages.

Poll for messages

Receive messages Info — Edit poll settings | Stop polling | Poll for messages

Messages available	Polling duration	Maximum message count	Polling progress
1	30	10	23% (1 receives/second)

Messages (1) — View details | Delete

Search messages

1

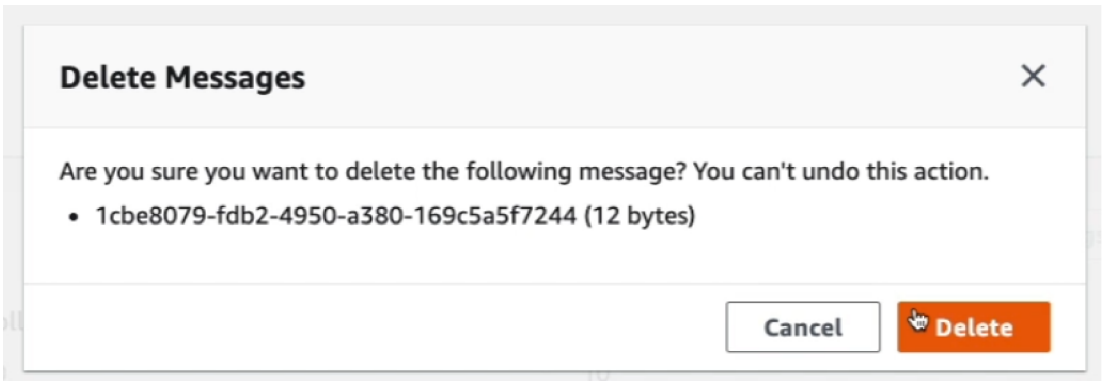
ID	Sent	Size	Receive count
1cbe8079-fdb2-4950-a380-169c5a5f7244	02/08/2020, 16:45:45	12 bytes	1

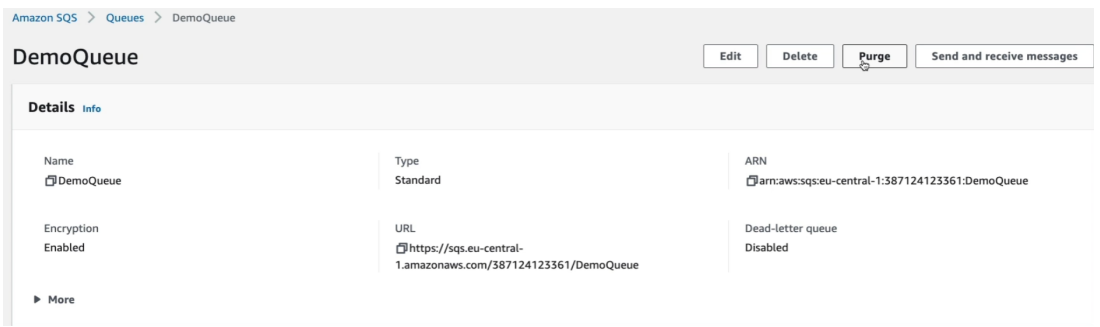
Message: 1cbe8079-fdb2-4950-a380-169c5a5f7244

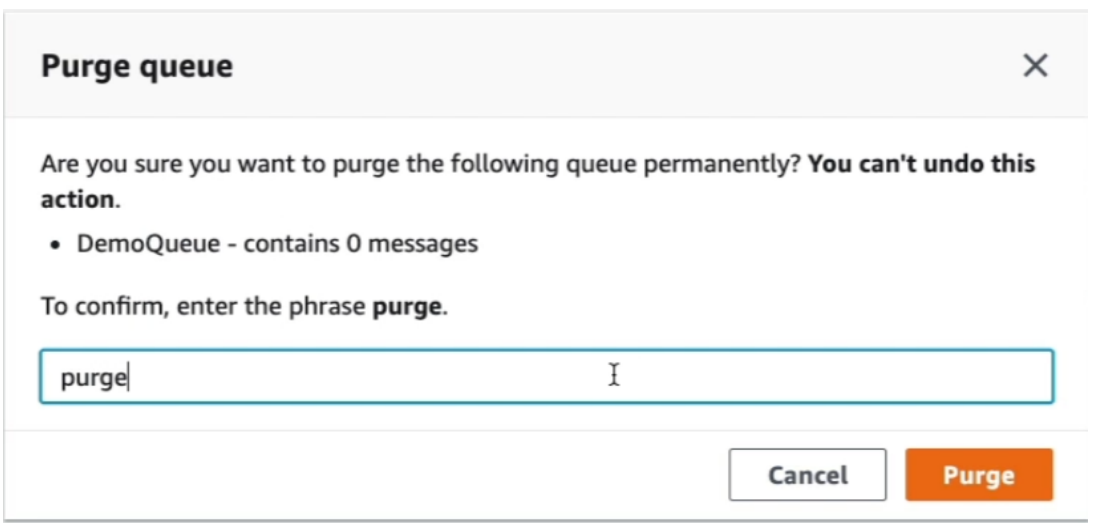
Details | Body | Attributes

ID	Size	MD5 of message body	Sender account ID
1cbe8079-fdb2-4950-a380-169c5a5f7244	12 bytes	fc3ff98e8c6a0d3087d515c0473f8677	387124123361
Sent	**First received**	**Receive count**	**Message attributes count**
02/08/2020, 16:45:45	02/08/2020, 16:45:56	1	-
Message attributes size	**MD5 of message attributes**		
-	-		

Done

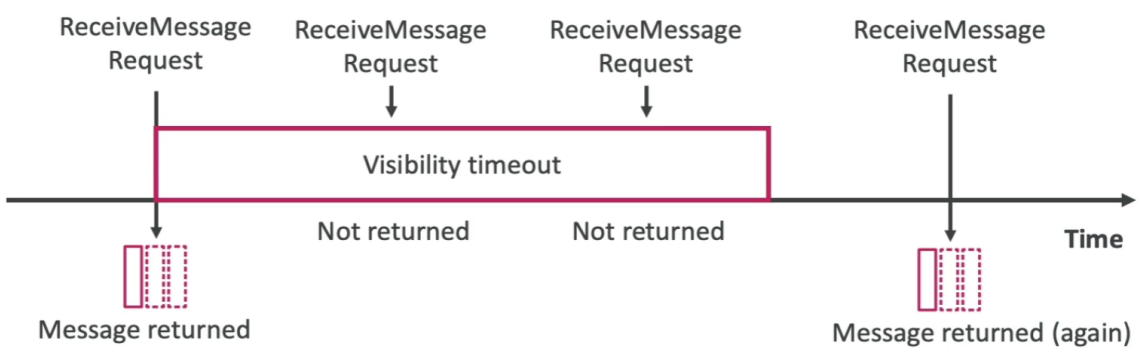






SQS - Message Visibility Timeout

- After a message is polled by a consumer, it becomes invisible to other consumers
- By default, the "message visibility timeout" is 30 seconds
- That means the message has 30 seconds to be processed
- After the message visibility timeout is over, the message is "visible" in SQS

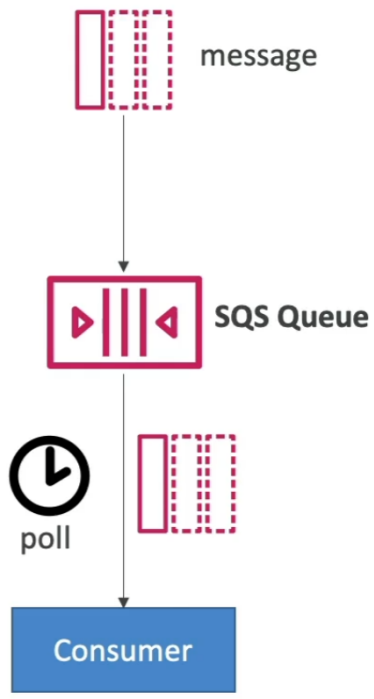


- If a message is not processed within the visibility timeout, it will be processed **twice**
- A consumer could call the **ChangeMessageVisibility** API to get more time
- If visibility timeout is high (hours), and consumer crashes, re-processing will take time
- If visibility timeout is too low (seconds), we may get duplicates

Amazon SQS - Long Polling

- When a consumer requests messages from the queue, it can optionally "wait" for messages to arrive if there are none in the queue
- This is called Long Polling
- LongPolling decreases the number of API calls made to SQS while increasing the efficiency and latency of your application
- The wait time can be between 1 sec to 20 sec (20 sec preferable)
- Long Polling is preferable to Short Polling

- Long polling can be enabled at the queue level or at the API level using **WaitTimeSeconds**

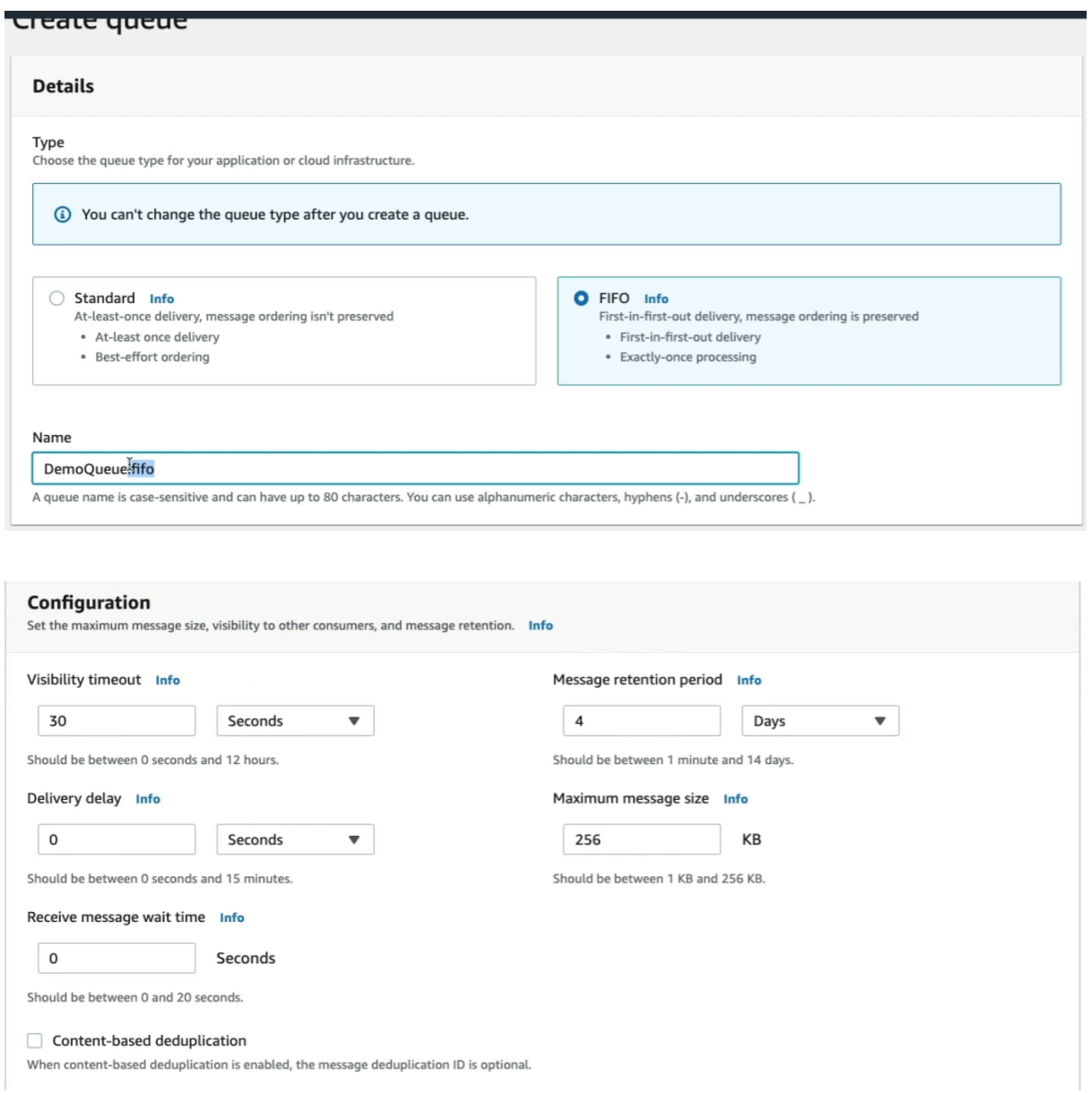


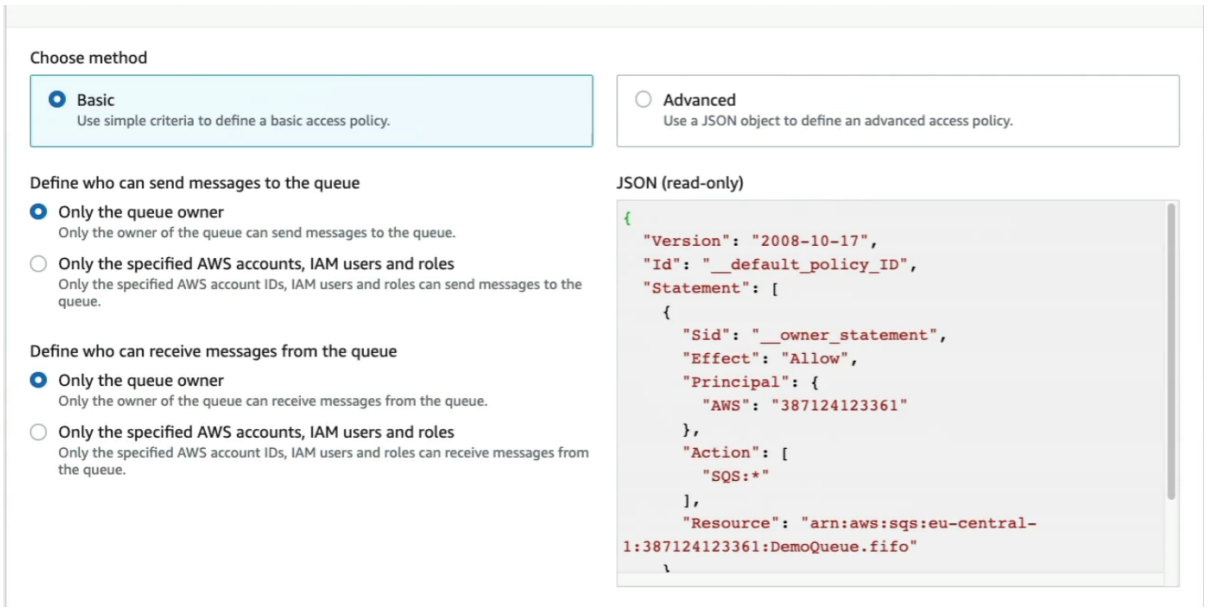
Amazon SQS - FIFO Queue

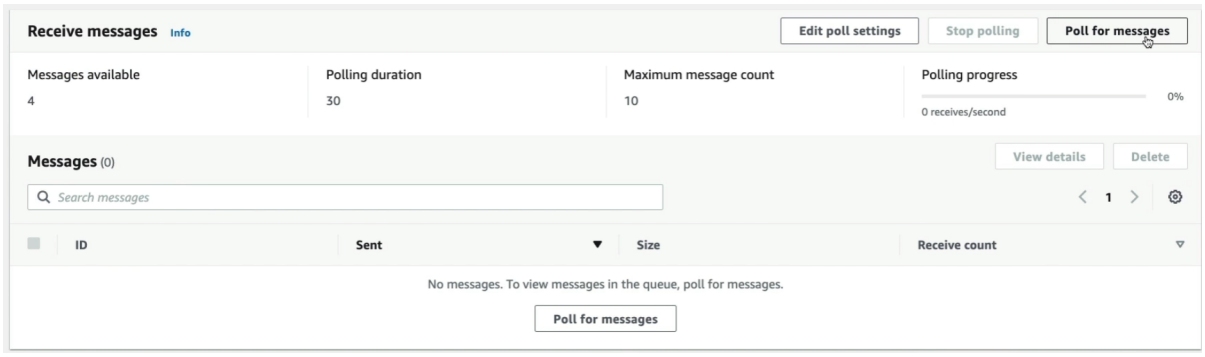
- FIFO = First In First Out (ordering of message in the queue)

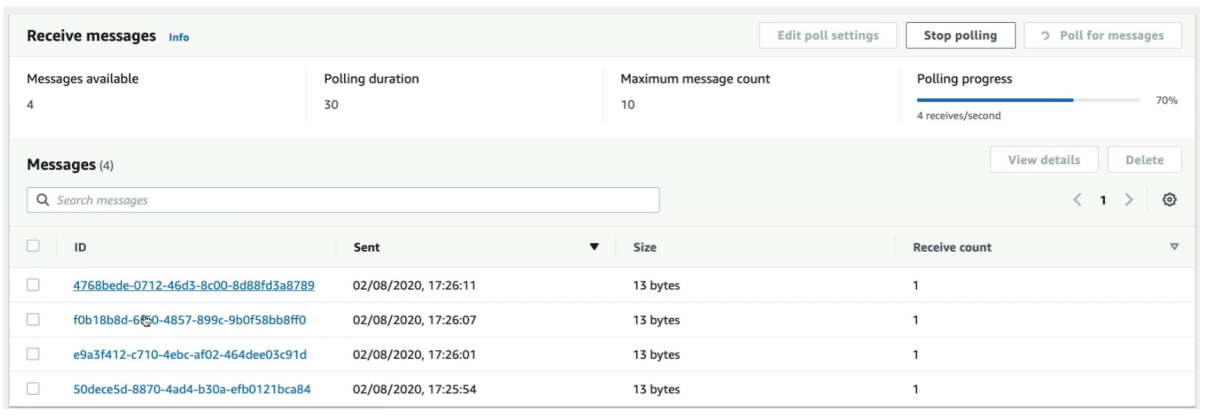
- Limited throughput: 300 msg/s without batching, 3000 msg/s with
- Exactly-once send capability (by removing duplicates)
- Messages are processed in order by the consumer

Hands On

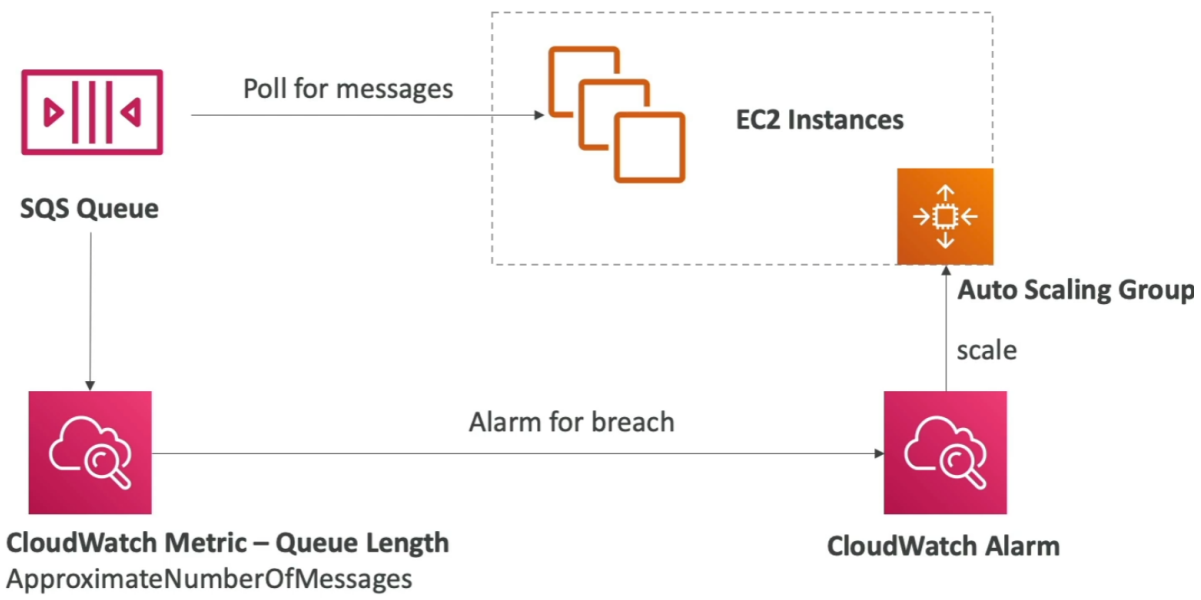




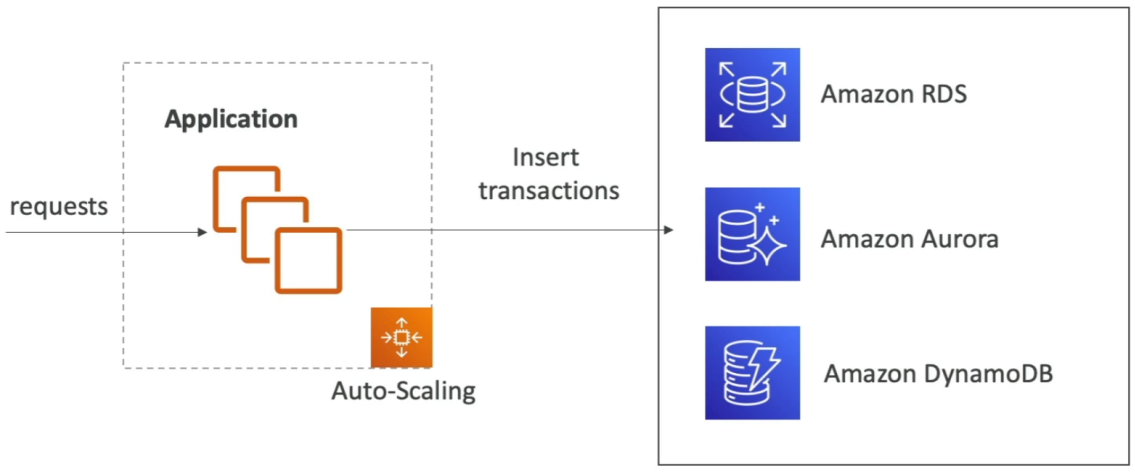




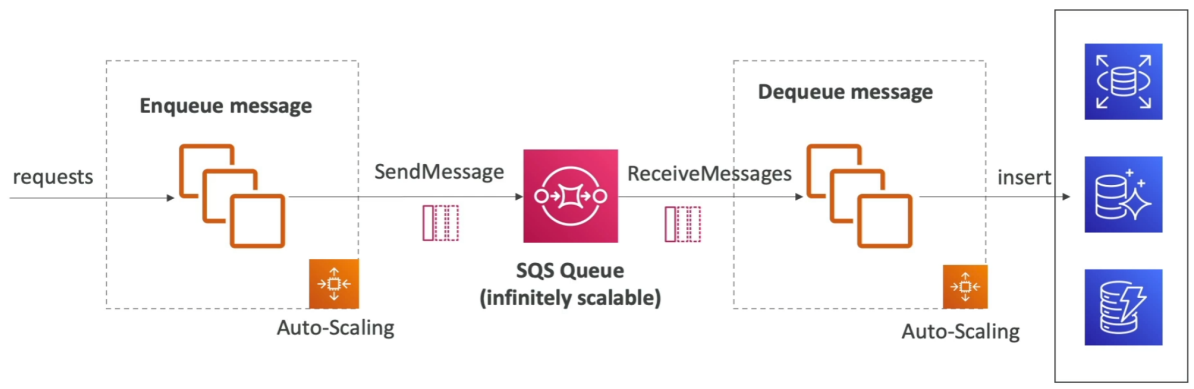
SQS with Auto Scaling Group (ASG)



If the load is too big, some transactions may be lost



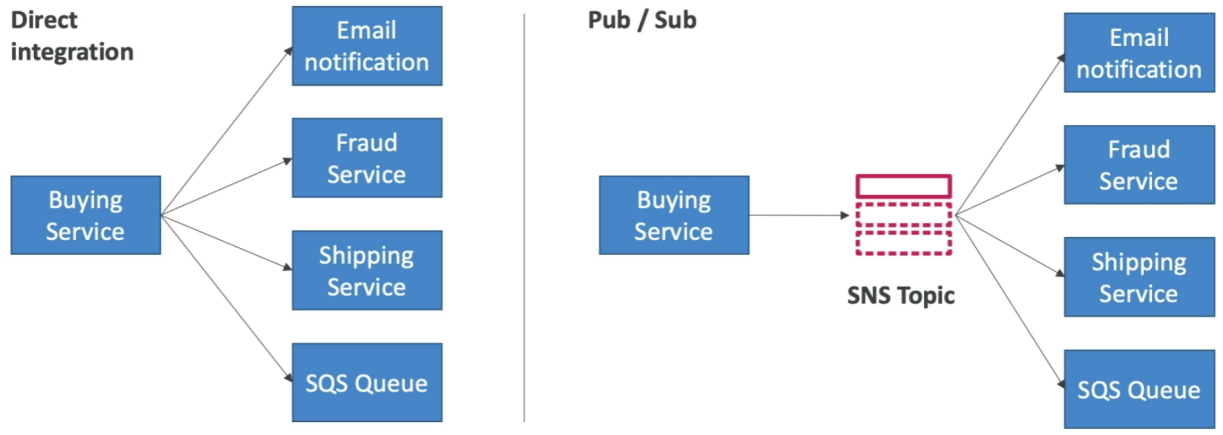
SQS as a buffer to database writes



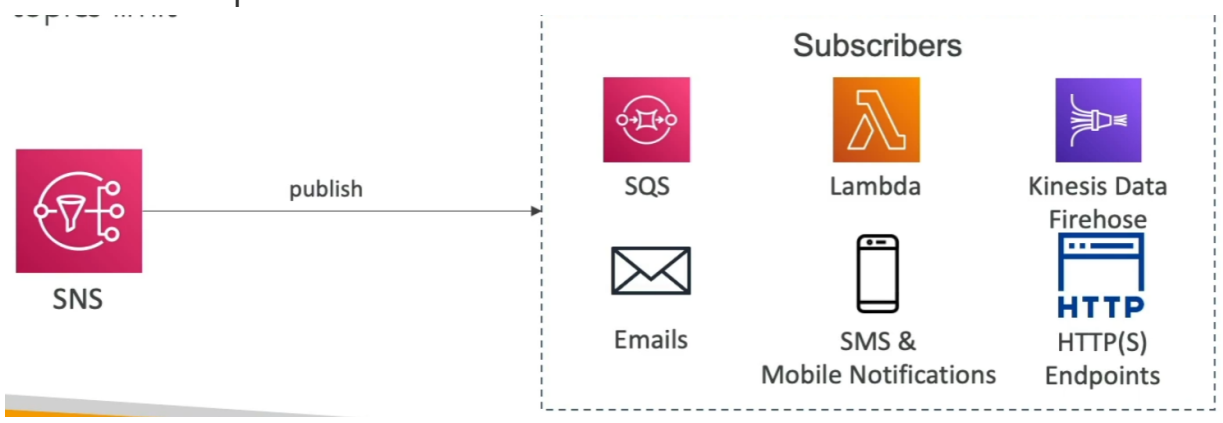
SQS to decouple between application

Amazon SNS

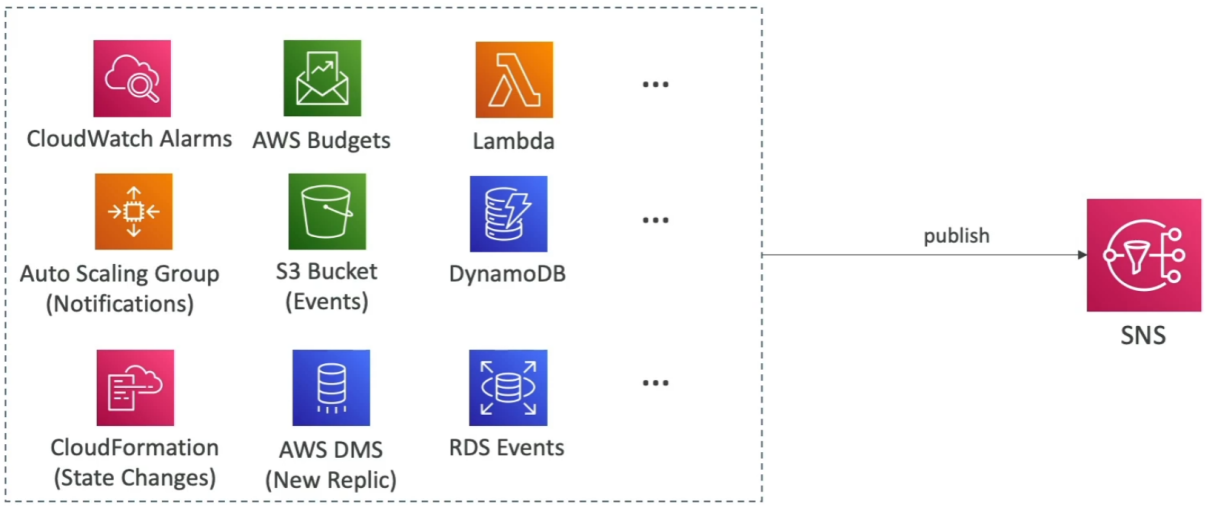
What if you want to send one message to many receivers?



- The "event producer" only sends message to one SNS topic
- As many "event receivers" (subscriptions) as we want to listen to the SNS topic notifications
- Each subscriber to the topic will get all the message (note: new feature to filter messages)
- Up to 12500000 subscriptions per topic
- 100000 topic limit



SNS integrates with a lot of AWS services



Many AWS services can send data directly to SNS for notifications

AWS SNS - How to publish

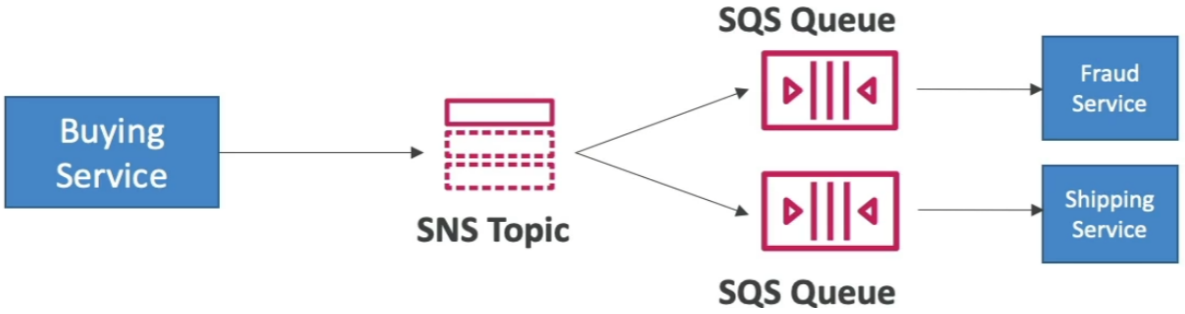
- Topic Publish (using the SDK)
  - Create a topic
  - Create a subscription (or many)
  - Publish to the topic
- Direct Publish (for mobile apps SDK)
  - Create a platform application
  - Create a platform endpoint
  - Publish to the platform endpoint
  - Works with Google GCM,Apple APNS,Amazon ADM...

Amazon SNS - Security

- **Encryption:**
  - In-flight encryption using HTTPS API
  - At-rest encryption using KMS keys
  - Client-side encryption if the client wants to perform encryption/decryption itself

- **Access Controls:IAM policies to regulate access to the SNS API**
- **SNS Access Policies (similar to S3 bucket policies)**
  - Useful for cross-account access to SNS topics
  - Useful for allowing other services (S3 ...) to write to an SNS topic

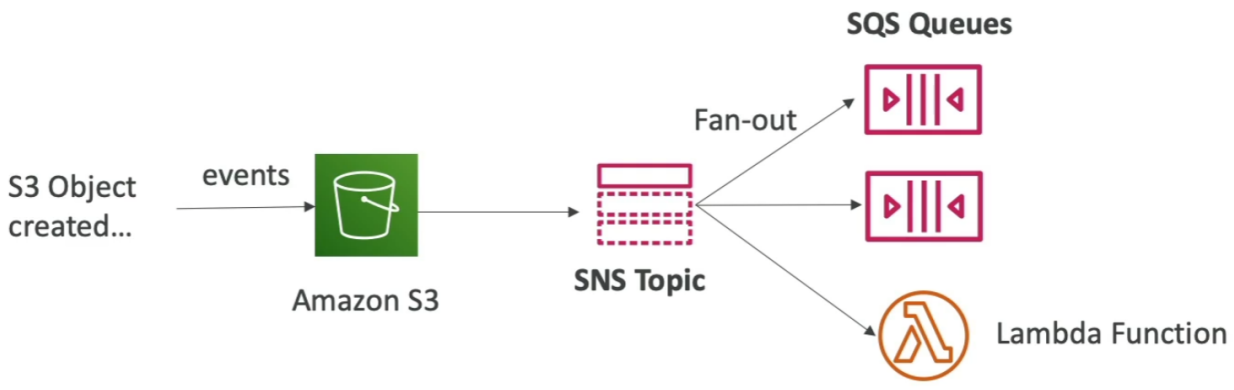
SNS + SQS: Fan Out



- Push once in SNS,receive in all SQS queues that are subscribers
- Fully decoupled, no data loss
- SQS allows for:data persistence, delayed processing and retries of work
- Ability to add more SQS subscribers over time
- Make sure your SQS queue access policy allows for SNS to write

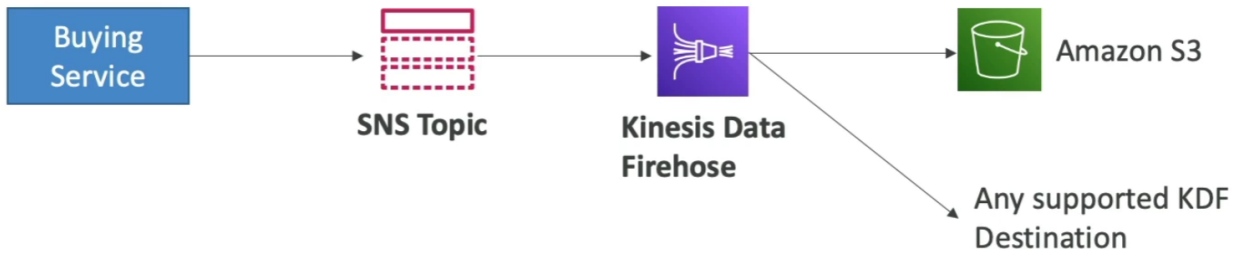
Application: S3 Events to multiple queues

- For the same combination of : event type (e.g. object create) and **prefix** (e.g. images/) you can only have one S3 Event rule
- If you want to send the same S3 event to many SQS queues, use fan-out



Application: SNS to Amazon S3 through Kinesis Data Firehose

- SNS can send to Kinesis and therefore we can have the following solutions architecture:



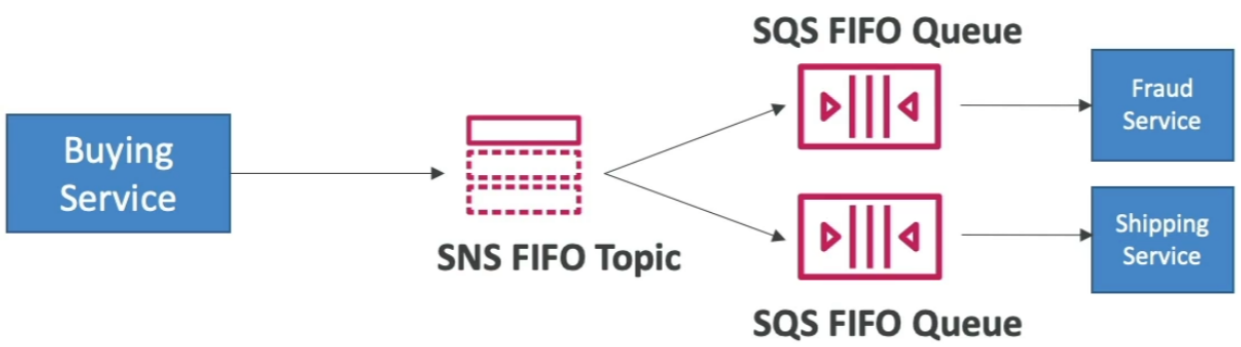
Amazon SNS - FIFO Topic

- FIFO = First In First Out (ordering of message in the topic)

- Similar features as SQS FIFO:
  - Ordering by Message Group ID (all messages in the same group are ordered)
  - Deduplication using a Deduplication ID or Content Based Deduplication
- Can only have SQS FIFO queues as subscribers
- Limited throughput (same throughput as SQS FIFO)

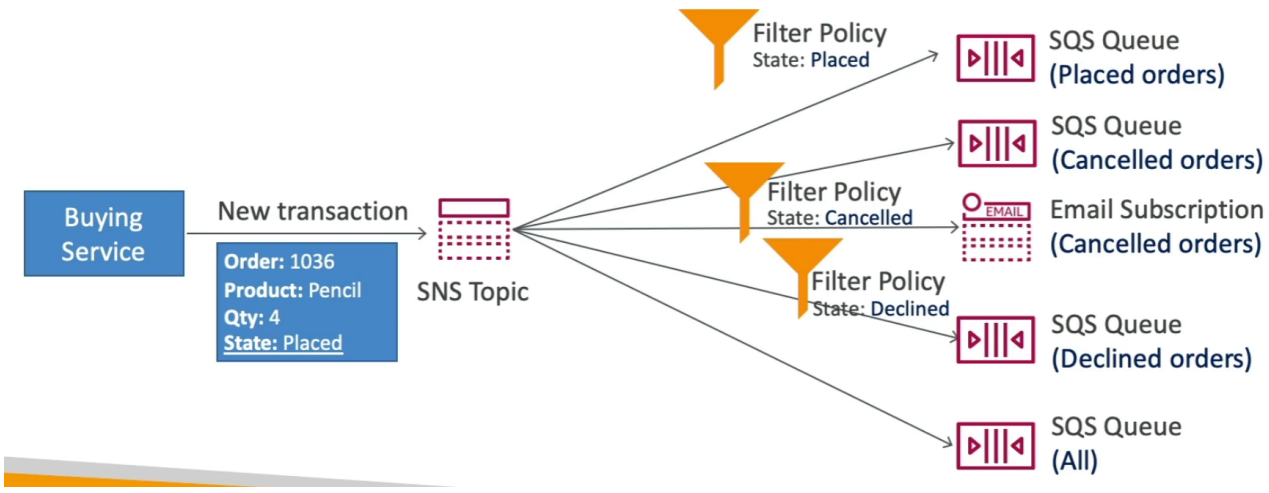
SNS FIFO + SQS FIFO: Fan Out

- In case you need fan out + ordering + deduplication

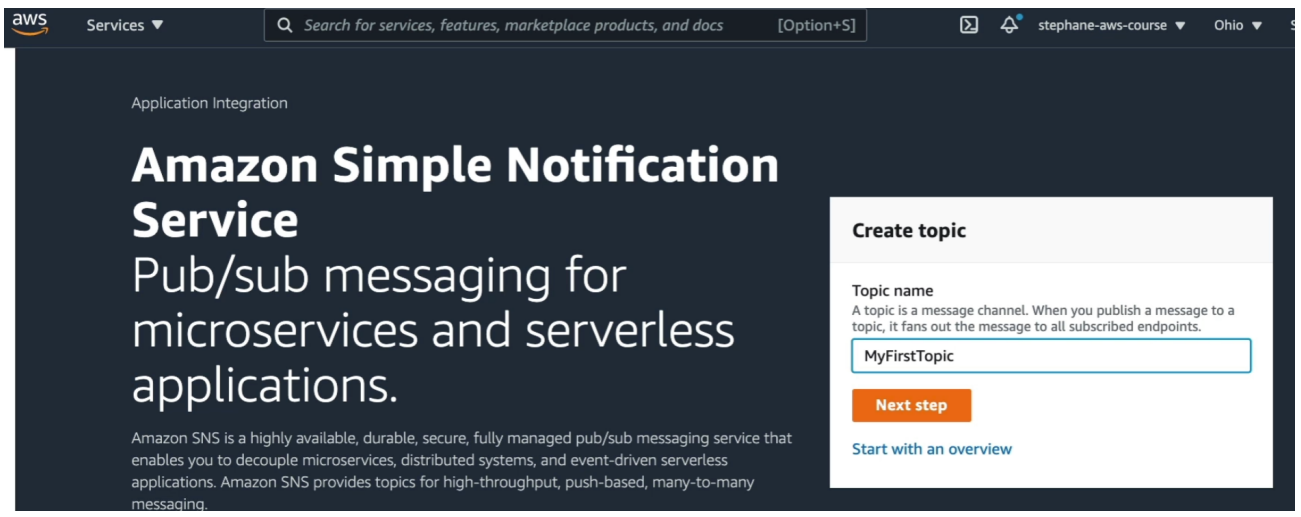


SNS - Message Filtering

- JSON policy used to filter message sent to SNS topic's subscriptions
- If a subscription doesn't have a filter policy, it receives every message



Hands On



Create topic

Details

Type Info

Topic type cannot be modified after topic is created

- FIFO (first-in, first-out)
  - Strictly-preserved message ordering
  - Exactly-once message delivery
  - High throughput, up to 300 publishes/second
  - Subscription protocols: SQS
- Standard
  - Best-effort message ordering
  - At-least once message delivery
  - Highest throughput in publishes/second
  - Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

MyFirstTopic

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - *optional*

To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. Info

My Topic

Maximum 100 characters, including hyphens (-) and underscores (_).

▸ **Encryption - *optional***

Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

▸ **Access policy - *optional***

This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic. Info

▸ **Delivery retry policy (HTTP/S) - *optional***

The policy defines how Amazon SNS retries failed deliveries to HTTP/S endpoints. To modify the default settings, expand this section. Info

▸ **Delivery status logging - *optional***

These settings configure the logging of message delivery status to CloudWatch Logs. Info

▸ **Tags - *optional***

A tag is a metadata label that you can assign to an Amazon SNS topic. Each tag consists of a key and an optional value. You can use tags to search and filter your topics and track your costs. Learn more

Cancel Create topic

Subscriptions (0)

Edit Delete Request confirmation Confirm subscription Create subscription

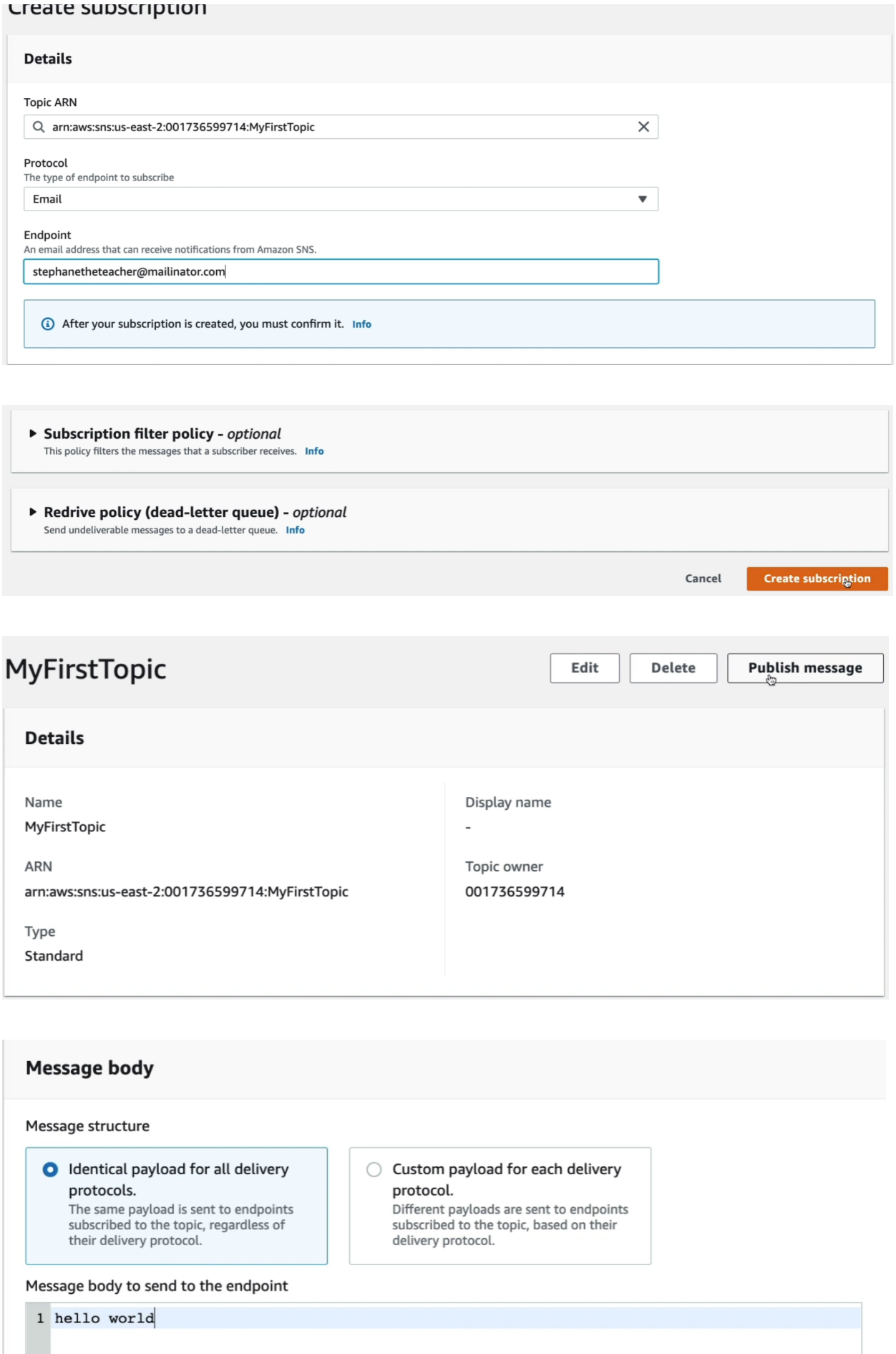
Search 1

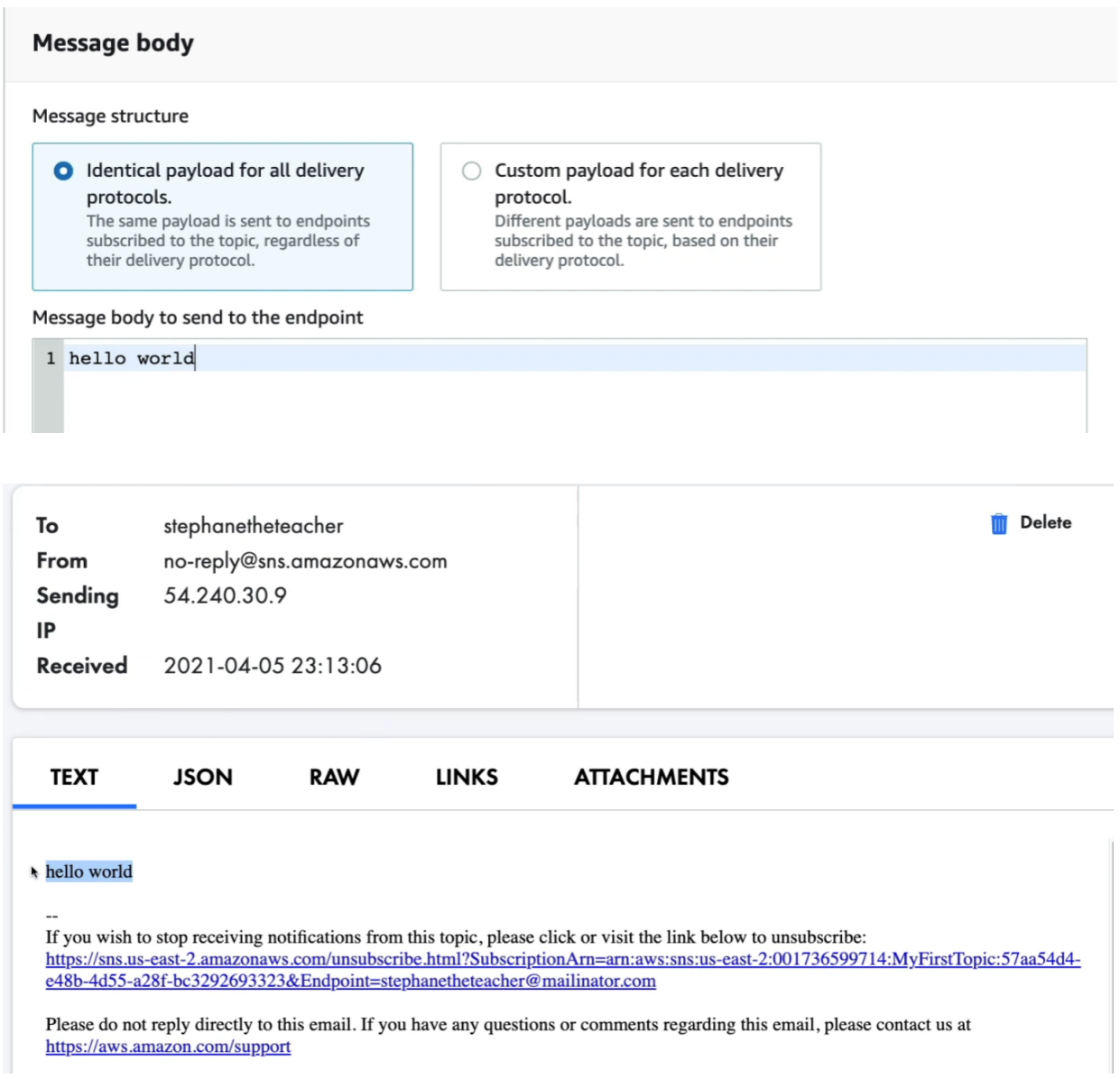
ID	Endpoint	Status	Protocol

No subscriptions found

You don't have any subscriptions to this topic.

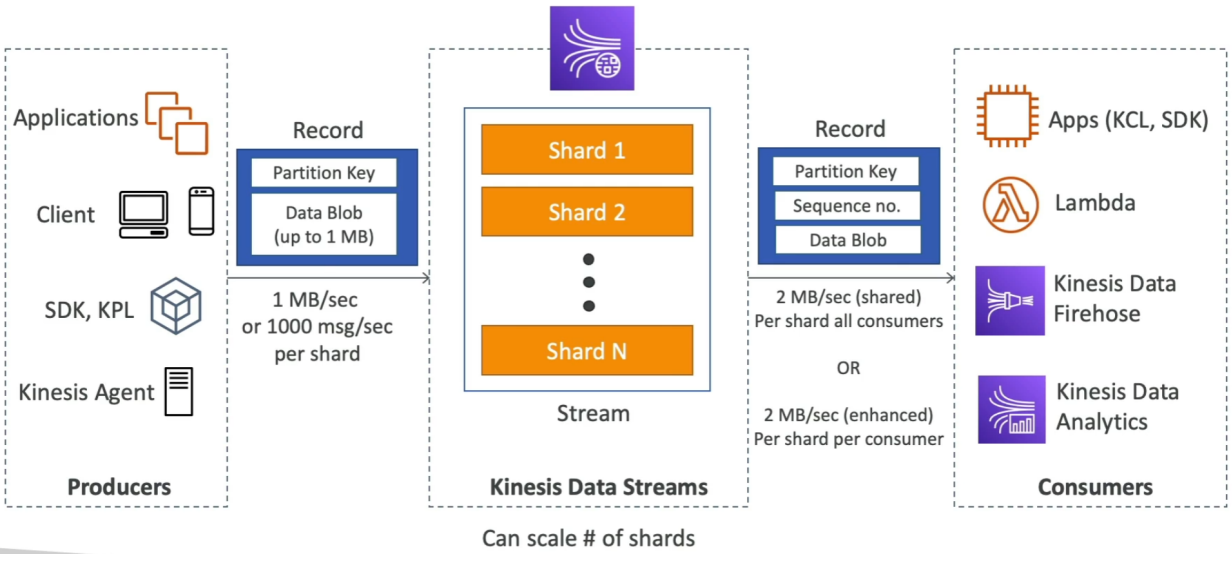
Create subscription



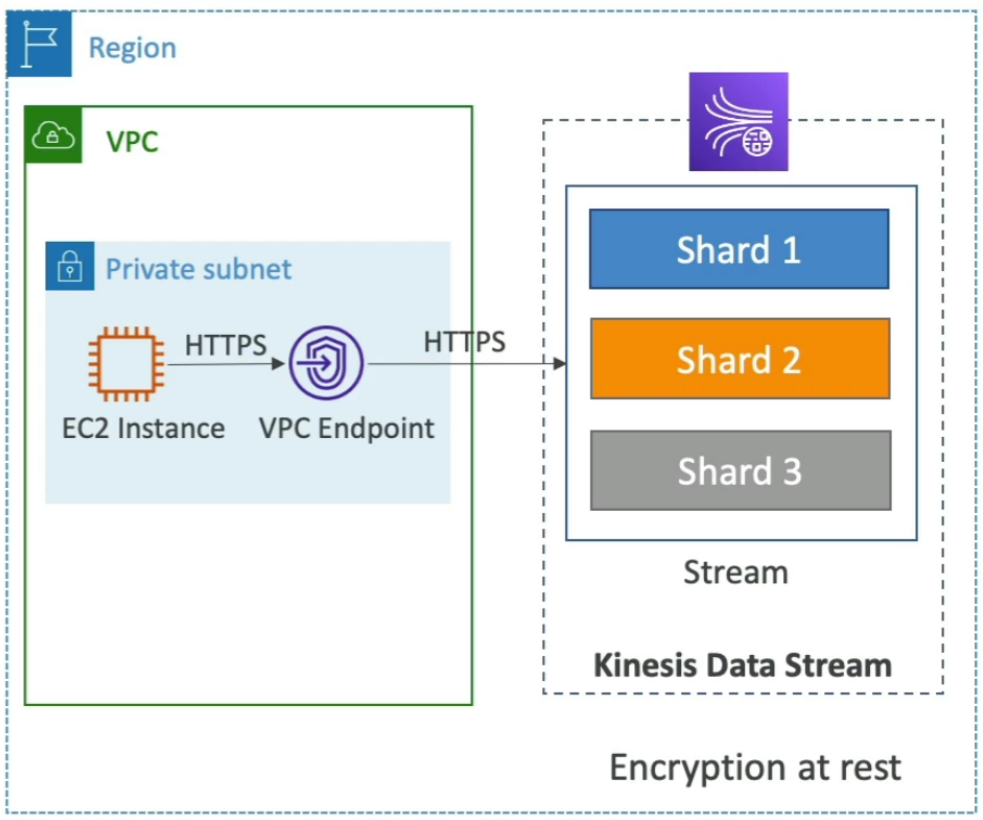


Kinesis Overview

- Makes it easy to collect,process, and analyze streaming data in real-time
- Ingest real-time data such as: Application logs, Metrics, Website clickstreams,IoT telemetry data ...
- **Kinesis Data Streams**: capture, process, and store data streams



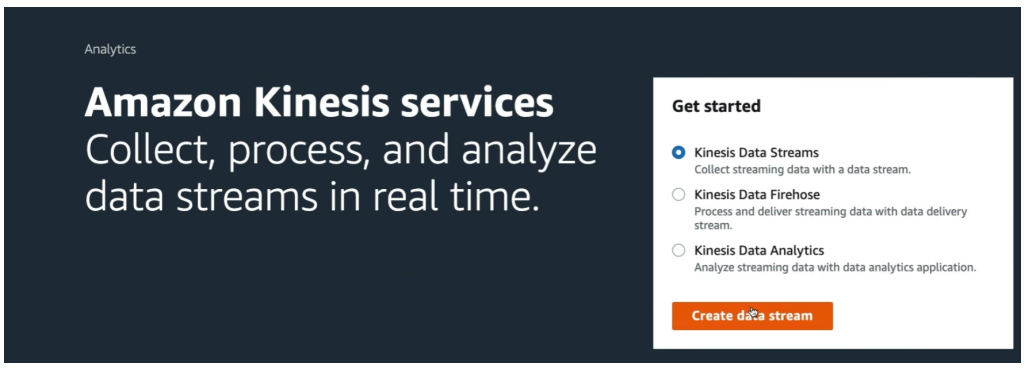
  - Retention between 1 day to 365 days
  - Ability to reprocess (replay) data
  - Once data is inserted in Kinesis, it can't be deleted (immutability)
  - Data that shares the same partition goes to the same shard (ordering)
  - Producers: AWS SDK, Kinesis Producer Library (KPL), Kinesis Agent
  - Consumers:
    - write your own: Kinesis Client Library(KCL),AWS SDK
    - Managed:AWS Lambda, Kinesis Data Firehose, Kinesis Data Analytics
  - Capacity Modes
    - **Provisioned mode**:
      - You choose the number of shards provisioned, scale manually or using API
      - Each shard gets 1MB/s in (or 1000 records per second)
      - Each shard gets 2MB/s out (classic or enhanced fan-out consumer)
      - You pay per shard provisioned per hour
    - **On-demand mode**:
      - No need to provision or manage the capacity
      - Default capacity provisioned (4 MB/s in or 4000 records per second)
      - Scales automatically based on observed throughput peak during the last 30 days
      - Pay per stream per hour & data in/out per GB
  - Security

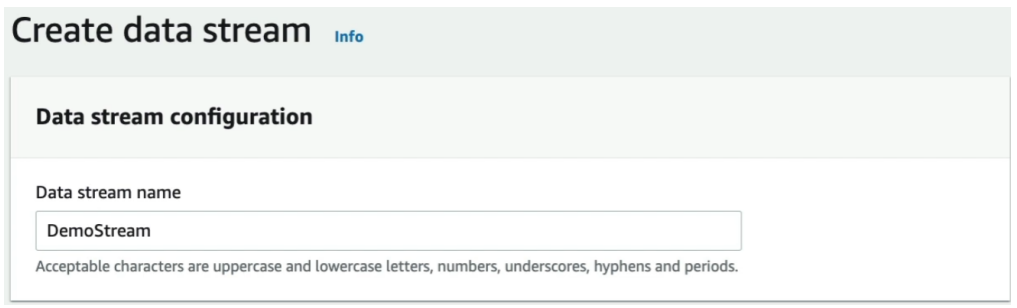


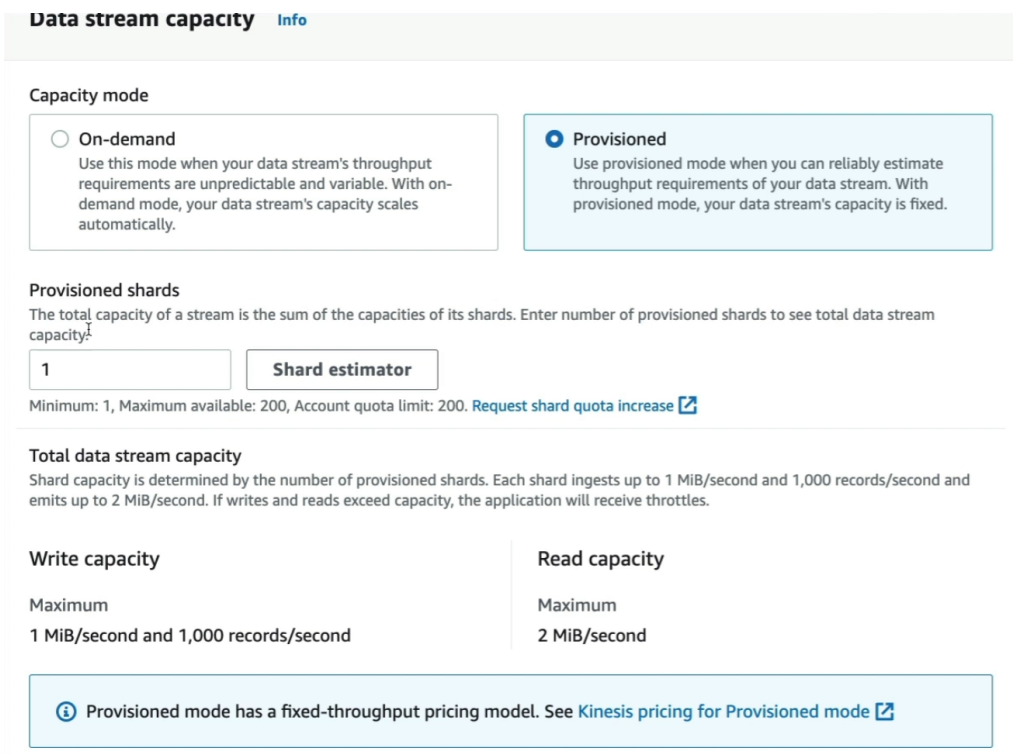
    - Control access / authorization using IAM policies
    - Encryption in flight using HTTPS endpoints
    - Encryption at rest using KMS
    - You can implement encryption/decryption of data on client side (harder)
    - VPC Endpoints available for Kinesis to access within VPC

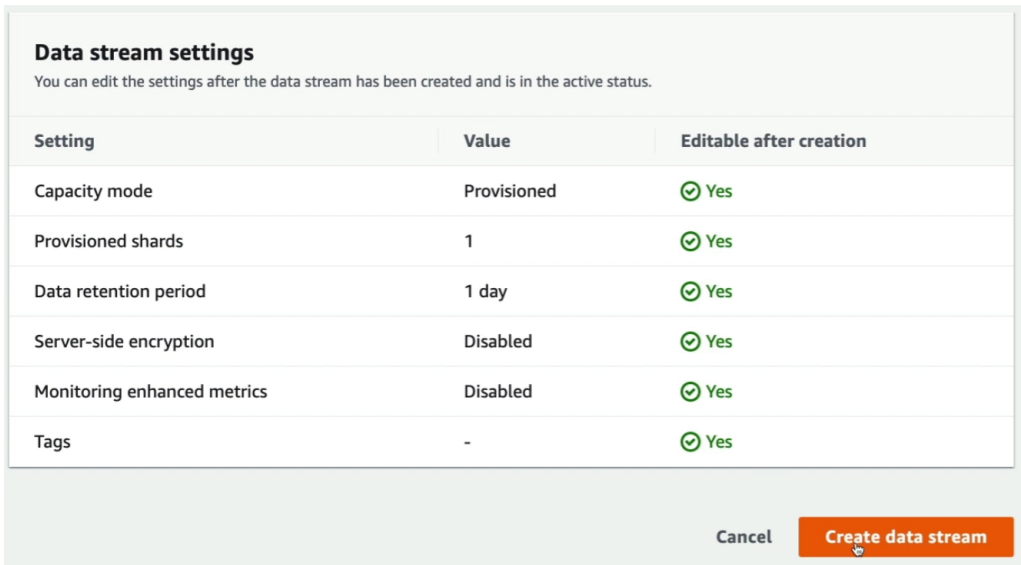
- Monitor API calls using CloudTrail

- Hands on


Analytics
Amazon Kinesis services
Collect, process, and analyze data streams in real time.
Get started
Kinesis Data Streams
Collect streaming data with a data stream.
Kinesis Data Firehose
Process and deliver streaming data with data delivery stream.
Kinesis Data Analytics
Analyze streaming data with data analytics application.
Create data stream


Create data stream Info
Data stream configuration
Data stream name
DemoStream
Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens and periods.


Data stream capacity Info
Capacity mode
On-demand
Use this mode when your data stream's throughput requirements are unpredictable and variable. With on-demand mode, your data stream's capacity scales automatically.
Provisioned
Use provisioned mode when you can reliably estimate throughput requirements of your data stream. With provisioned mode, your data stream's capacity is fixed.
Provisioned shards
The total capacity of a stream is the sum of the capacities of its shards. Enter number of provisioned shards to see total data stream capacity.
1
Shard estimator
Minimum: 1, Maximum available: 200, Account quota limit: 200. Request shard quota increase
Total data stream capacity
Shard capacity is determined by the number of provisioned shards. Each shard ingests up to 1 MiB/second and 1,000 records/second and emits up to 2 MiB/second. If writes and reads exceed capacity, the application will receive throttles.
Write capacity
Maximum
1 MiB/second and 1,000 records/second
Read capacity
Maximum
2 MiB/second
Provisioned mode has a fixed-throughput pricing model. See Kinesis pricing for Provisioned mode


Data stream settings
You can edit the settings after the data stream has been created and is in the active status.
Setting
Value
Editable after creation
Capacity mode
Provisioned
Yes
Provisioned shards
1
Yes
Data retention period
1 day
Yes
Server-side encryption
Disabled
Yes
Monitoring enhanced metrics
Disabled
Yes
Tags
-
Yes
Cancel
Create data stream

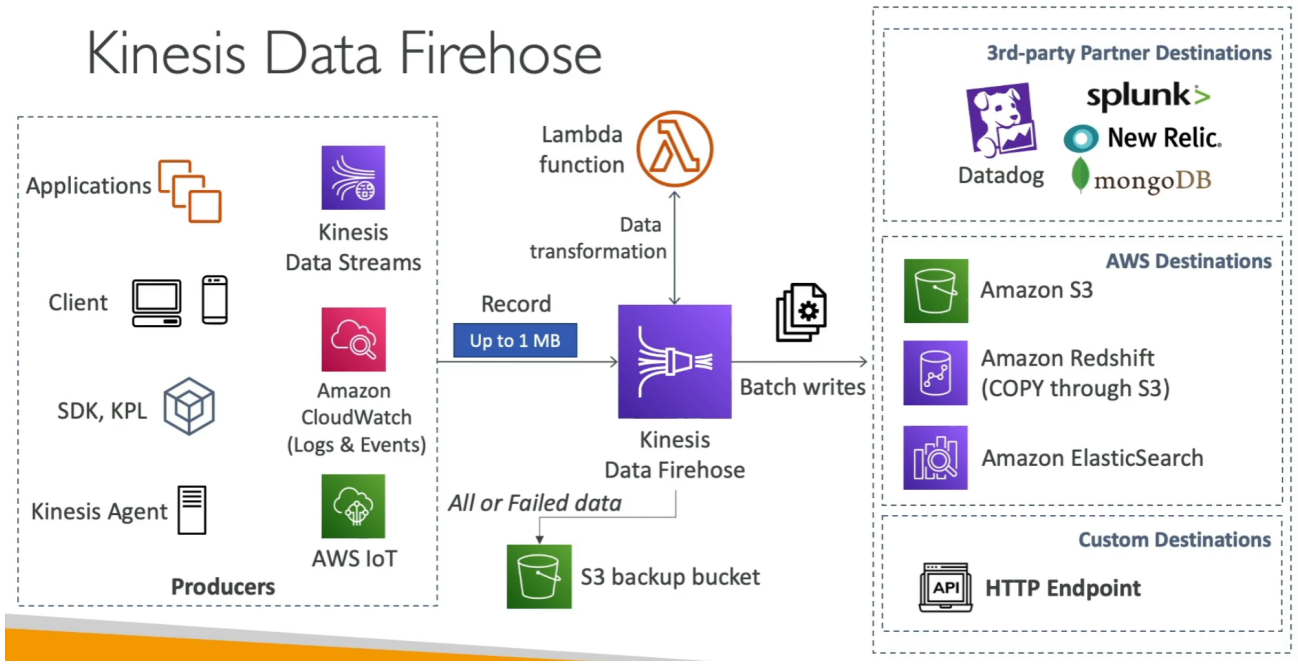
```
[cloudshell-user@ip-10-0-39-58 ~]$ Try these commands to get started:
aws help  or  aws <command> help  or  aws <command> --cli-auto-prompt
[cloudshell-user@ip-10-0-39-58 ~]$ aws --version
aws-cli/2.1.16 Python/3.7.3 Linux/4.14.225-168.357.amzn2.x86_64 exec-env/CloudShel
l exe/x86_64.amzn.2 prompt/off
[cloudshell-user@ip-10-0-39-58 ~]$ aws kinesis put-record --stream-name DemoStream
 --partition-key user1 --data "user signup" --cli-binary-format raw-in-base64-out
{
    "ShardId": "shardId-000000000000",
    "SequenceNumber": "49617390934629201455926329624637508189499828916235272194"
}
[cloudshell-user@ip-10-0-39-58 ~]$ aws kinesis put-record --stream-name DemoStream
 --partition-key user1 --data "user signup" --cli-binary-format raw-in-base64-out
{
    "ShardId": "shardId-000000000000",
    "SequenceNumber": "49617390934629201455926329624638717115319444438763175938"
}
[cloudshell-user@ip-10-0-39-58 ~]$ aws kinesis put-record --stream-name DemoStream
 --partition-key user1 --data "user login" --cli-binary-format raw-in-base64-out
{
    "ShardId": "shardId-000000000000",
    "SequenceNumber": "49617390934629201455926329624639926041139059686413172738"
}
[cloudshell-user@ip-10-0-39-58 ~]$ aws kinesis put-record --stream-name DemoStream
 --partition-key user1 --data "user logout" --cli-binary-format raw-in-base64-out
```

```
[cloudshell-user@ip-10-0-39-58 ~]$ aws kinesis describe-stream --stream-name DemoS
tream
{
    "StreamDescription": {
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "StartingHashKey": "0",
                    "EndingHashKey": "340282366920938463463374607431768211455"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber": "49617390934629201455926329620716961
756489569047916052482"
                }
            }
        ],
        "StreamARN": "arn:aws:kinesis:us-east-1:001736599714:stream/DemoStream",
        "StreamName": "DemoStream",
        "StreamStatus": "ACTIVE",
        "RetentionPeriodHours": 24,
        "EnhancedMonitoring": [
            {
                "ShardLevelMetrics": []
```

```
[cloudshell-user@ip-10-0-39-58 ~]$ aws kinesis get-shard-iterator --stream-name De
moStream --shard-id shardId-000000000000 --shard-iterator-type TRIM_HORIZON
{
    "ShardIterator": "AAAAAAAAAAGtYgUq0OwFKpIqqDHDoswgADDT3VzUW6qPzgR04TPuWEbCgUZP
VjI2adlKwjAUsIIpjLYOaC82nhF+Ml2nlT5WxK3WLltulh34OZrsVw9Q0gT6QpaZlkqi6wSKfb2yrqxj6S
hbHueiUuqQMH/5hW2NRaLhgdxm6thQtoZF+BS+ljcye7BJV1iwNKz29KH96EqGnn9UcxLyZoSLWXD/KV1u
ck88QV+u02S/HQ1+v/xKmw=="
}
[cloudshell-user@ip-10-0-39-58 ~]$ aws kinesis get-records --shard-iterator "AAAAA
AAAAAGtYgUq0OwFKpIqqDHDoswgADDT3VzUW6qPzgR04TPuWEbCgUZPVjI2adlKwjAUsIIpjLYOaC82nhF
+Ml2nlT5WxK3WLltulh34OZrsVw9Q0gT6QpaZlkqi6wSKfb2yrqxj6ShbHueiUuqQMH/5hW2NRaLhgdxm6
thQtoZF+BS+ljcye7BJV1iwNKz29KH96EqGnn9UcxLyZoSLWXD/KV1uck88QV+u02S/HQ1+v/xKmw=="
```

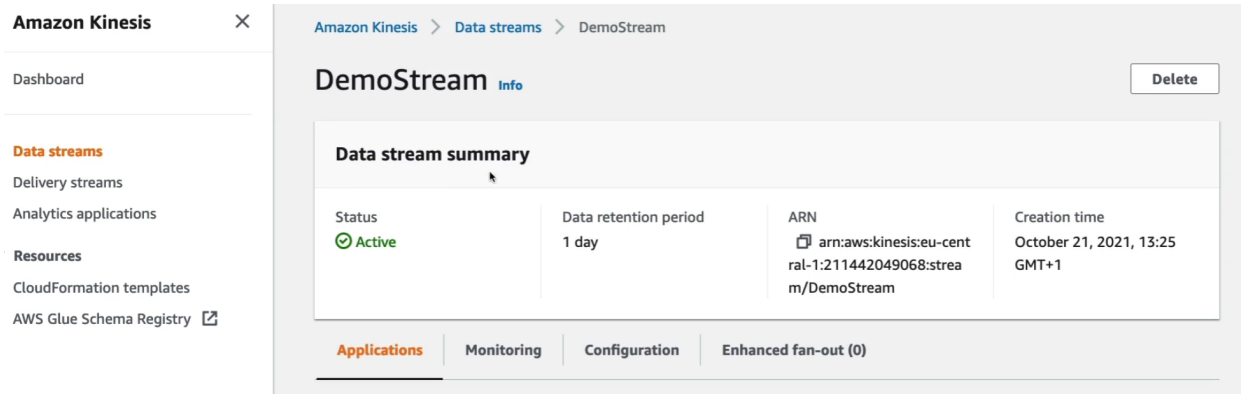
```
{
    "Records": [
        {
            "SequenceNumber": "49617390934629201455926329624637508189499828916235272194",
            "ApproximateArrivalTimestamp": "2021-04-16T17:30:24.565000+00:00",
            "Data": "dXNlciBzaWdudXA=",
            "PartitionKey": "user1"
        },
        {
            "SequenceNumber": "49617390934629201455926329624638717115319444438763175938",
            "ApproximateArrivalTimestamp": "2021-04-16T17:30:37.358000+00:00",
            "Data": "dXNlciBzaWdudXA=",
            "PartitionKey": "user1"
        },
        {
            "SequenceNumber": "49617390934629201455926329624639926041139059686413172738",
            "ApproximateArrivalTimestamp": "2021-04-16T17:30:45.978000+00:00",
            "Data": "dXNlciBsb2dpbg==",
            "PartitionKey": "user1"
        },
        {
```

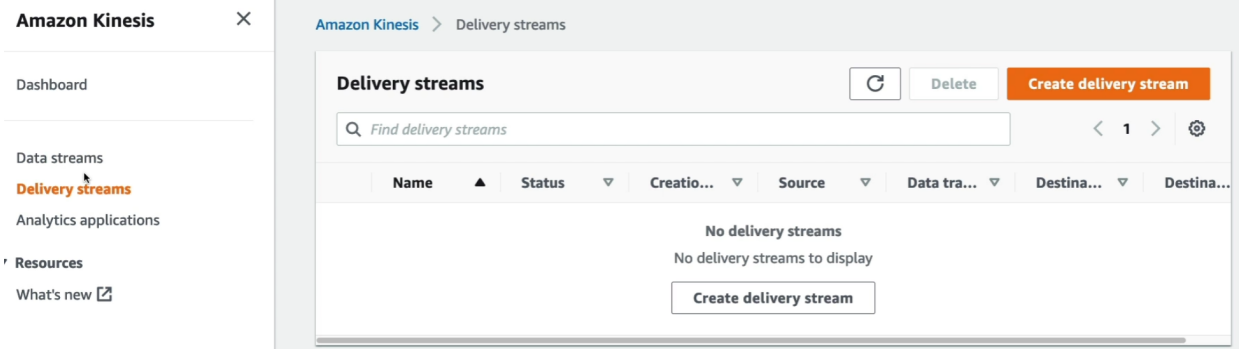
- **Kinesis Data Firehose**: load data streams into AWS data stores

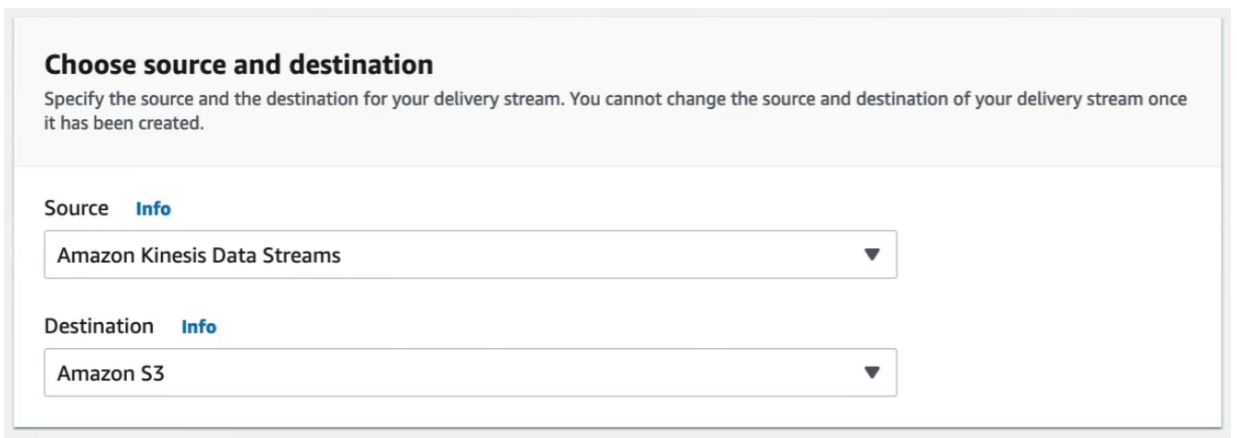


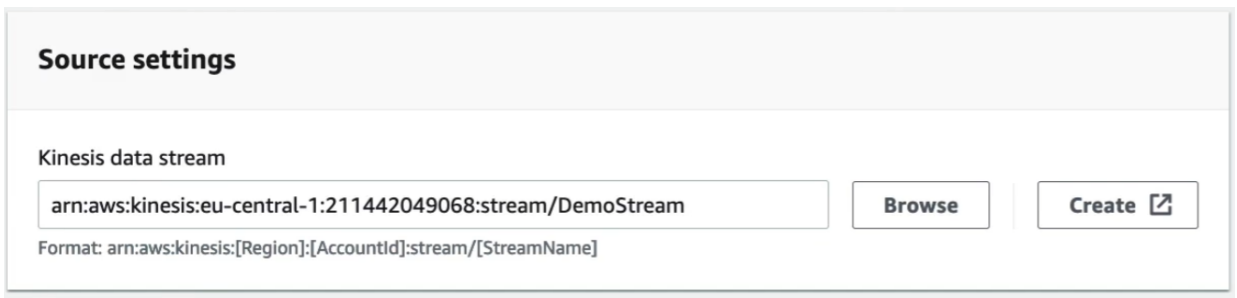
  - Fully Managed Service,no administration, automatic scaling. serverless
    - AWS: Redshift / Amazon S3 / ElasticSearch
    - 3rd party partner: Splunk / MongoDB / DataDog / NewRelic /...
    - Custom: send to any HTTP endpoint
  - Pay for data going through Firehose
  - **Near Real Time**
    - 60 seconds latency minimum for non full batches
    - Or minimum 1 MB of data at at a time
  - Supports many data formats, conversions, transformations,compression
  - Supports custom data transformations using AWS Lambda
  - Can send failed or all data to a backup S3 bucket

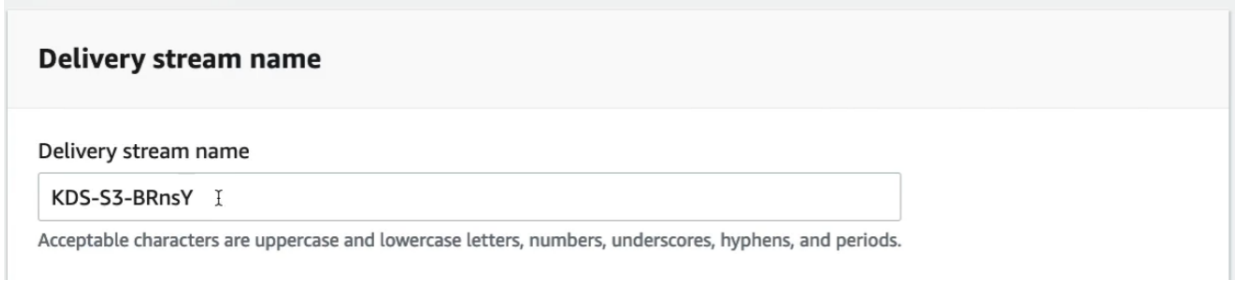
- Hands on

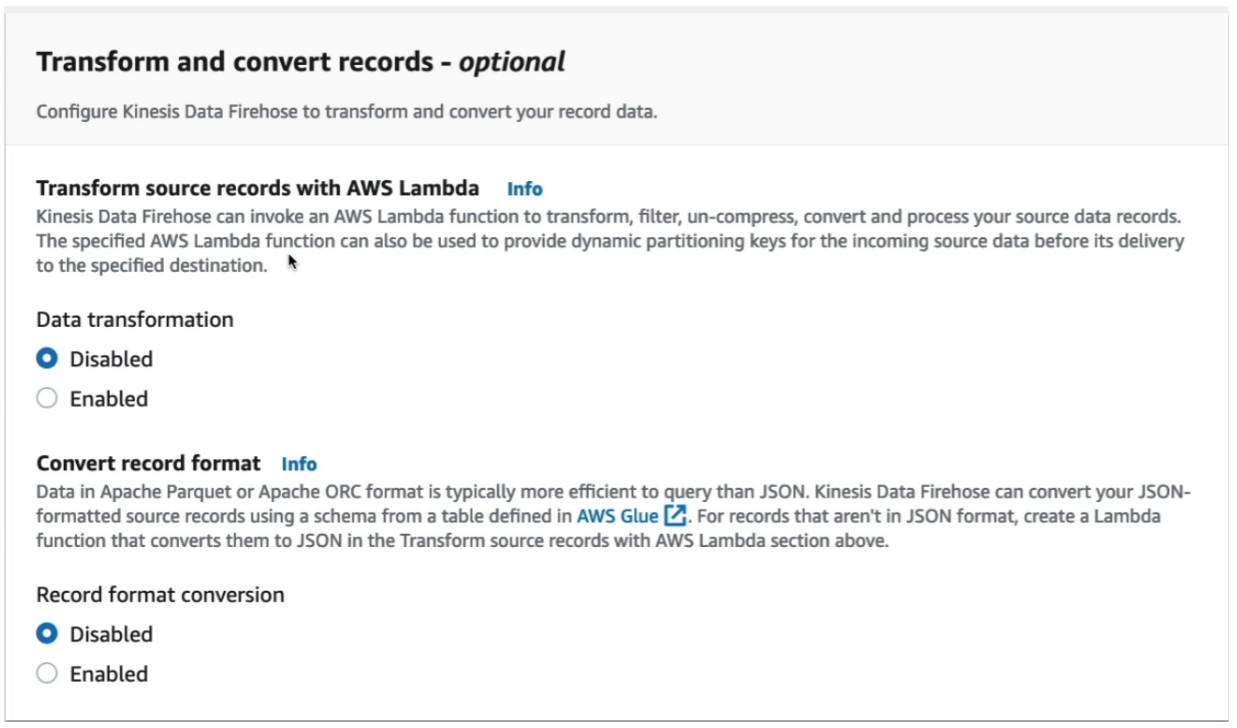

Amazon Kinesis
Dashboard
Data streams
Delivery streams
Analytics applications
Resources
CloudFormation templates
AWS Glue Schema Registry
Amazon Kinesis > Data streams > DemoStream
DemoStream Info
Delete
Data stream summary
Status
Active
Data retention period
1 day
ARN
arn:aws:kinesis:eu-central-1:211442049068:stream/DemoStream
Creation time
October 21, 2021, 13:25 GMT+1
Applications
Monitoring
Configuration
Enhanced fan-out (0)


Amazon Kinesis
Dashboard
Data streams
Delivery streams
Analytics applications
Resources
What's new
Amazon Kinesis > Delivery streams
Delivery streams
Delete
Create delivery stream
Find delivery streams
1
Name
Status
Creatio...
Source
Data tra...
Destina...
Destina...
No delivery streams
No delivery streams to display
Create delivery stream


Choose source and destination
Specify the source and the destination for your delivery stream. You cannot change the source and destination of your delivery stream once it has been created.
Source Info
Amazon Kinesis Data Streams
Destination Info
Amazon S3


Source settings
Kinesis data stream
arn:aws:kinesis:eu-central-1:211442049068:stream/DemoStream
Browse
Create
Format: arn:aws:kinesis:[Region]:[AccountId]:stream/[StreamName]


Delivery stream name
Delivery stream name
KDS-S3-BRnsY
Acceptable characters are uppercase and lowercase letters, numbers, underscores, hyphens, and periods.


Transform and convert records - optional
Configure Kinesis Data Firehose to transform and convert your record data.
Transform source records with AWS Lambda Info
Kinesis Data Firehose can invoke an AWS Lambda function to transform, filter, un-compress, convert and process your source data records. The specified AWS Lambda function can also be used to provide dynamic partitioning keys for the incoming source data before its delivery to the specified destination.
Data transformation
Disabled
Enabled
Convert record format Info
Data in Apache Parquet or Apache ORC format is typically more efficient to query than JSON. Kinesis Data Firehose can convert your JSON-formatted source records using a schema from a table defined in AWS Glue. For records that aren't in JSON format, create a Lambda function that converts them to JSON in the Transform source records with AWS Lambda section above.
Record format conversion
Disabled
Enabled

Destination settings Info

Specify the destination settings for your delivery stream.

S3 bucket

s3://demo-firehose-stephane-v3 | Browse | Create

Format: s3://bucket

Dynamic partitioning Info

Dynamic partitioning enables you to create targeted data sets by partitioning streaming S3 data based on partitioning keys. You can partition your source data with inline parsing and/or the specified AWS Lambda function. You can enable dynamic partitioning only when you create a new delivery stream. You cannot enable dynamic partitioning for an existing delivery stream. Enabling dynamic partitioning incurs additional costs per GiB of partitioned data. For more information, see Kinesis Data Firehose pricing.

- (•) Disabled
- ( ) Enabled

S3 bucket prefix - *optional*

By default, Kinesis Data Firehose appends the prefix "YYYY/MM/dd/HH" (in UTC) to the data it delivers to Amazon S3. You can override this default by specifying a custom prefix that includes expressions that are evaluated at runtime.

Enter a prefix

Amazon CloudWatch error logging Info

Choose Enabled if you want Kinesis Data Firehose to log record delivery errors to CloudWatch Logs.

- ( ) Disabled
- (•) Enabled

Permissions Info

Kinesis Data Firehose uses this IAM role for all the permissions that the delivery stream needs. To specify different roles for the different permissions, use the API or the CLI.

- (•) Create or update IAM role **KinesisFirehoseServiceRole-KDS-S3-BRn-eu-central-1-1634819284161**
  Creates a new role or updates an existing one and adds the required policies to it, and enables Kinesis Data Firehose to assume it.
- ( ) Choose existing IAM role
  The role that you choose must have policies that include the permissions that Kinesis Data Firehose needs.

Tags Info

You can add tags to organize your AWS resources, track costs, and control access.

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel | Create delivery stream

```
[cloudshell-user@ip-10-0-38-107 ~]$ aws kinesis put-record --stream-name DemoStream --partition-key user1 --data "user signup" --cli-binary-format raw-in-base64-out
{
    "ShardId": "shardId-000000000000",
    "SequenceNumber": "49623180259820468866284917982806735146293046538896670722"
}
[cloudshell-user@ip-10-0-38-107 ~]$ aws kinesis put-record --stream-name DemoStream --partition-key user1 --data "user login" --cli-binary-format raw-in-base64-out
{
    "ShardId": "shardId-000000000000",
    "SequenceNumber": "49623180259820468866284917982807944072112661580388237314"
}
[cloudshell-user@ip-10-0-38-107 ~]$ aws kinesis put-record --stream-name DemoStream --partition-key user1 --data "user logout" --cli-binary-format raw-in-base64-out
{
    "ShardId": "shardId-000000000000",
    "SequenceNumber": "49623180259820468866284917982809152997932276553160327170"
}
[cloudshell-user@ip-10-0-38-107 ~]$
```

Amazon S3 > demo-firehose-stephane-v3

demo-firehose-stephane-v3 Info

Objects | Properties | Permissions | Metrics | Management | Access Points

Objects (1)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Copy S3 URI | Copy URL | Download | Open | Delete | Actions

Create folder | Upload

Find objects by prefix 1

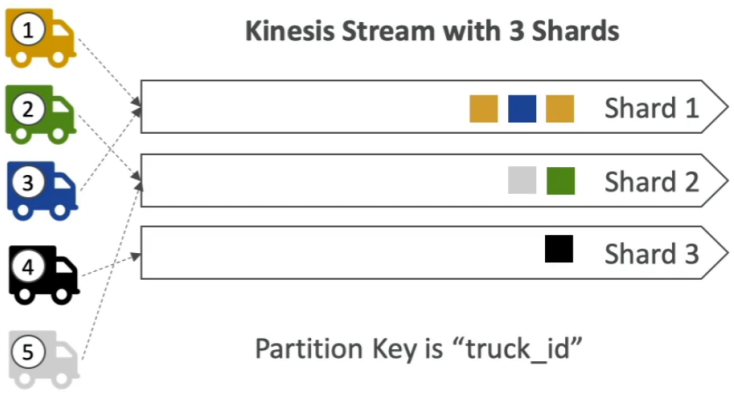
Name	Type	Last modified	Size	Storage class
2021/	Folder	-	-	-

Kinesis Data Streams vs Firehose

Kinesis Data Streams	Kinesis Data Firehose
• Streaming service for ingest at scale • Write custom code (producer / consumer) • Real-time (~200ms) • Manage scaling (shard splitting / merging) • Data storage for 1 to 365 days • Supports replay capability	• Load streaming data into S3 / Redshift / ES / 3rd party / custom HTTP • Fully managed • Near real-time (buffer time min. 60 sec) • Automatic scaling • No data storage • Doesn't support replay capability

- **Kinesis Data Analytics**: analyze data streams with SQL or Apache Flink
- **Kinesis Video Streams**: capture,process, and store video streams

Ordering data into Kinesis

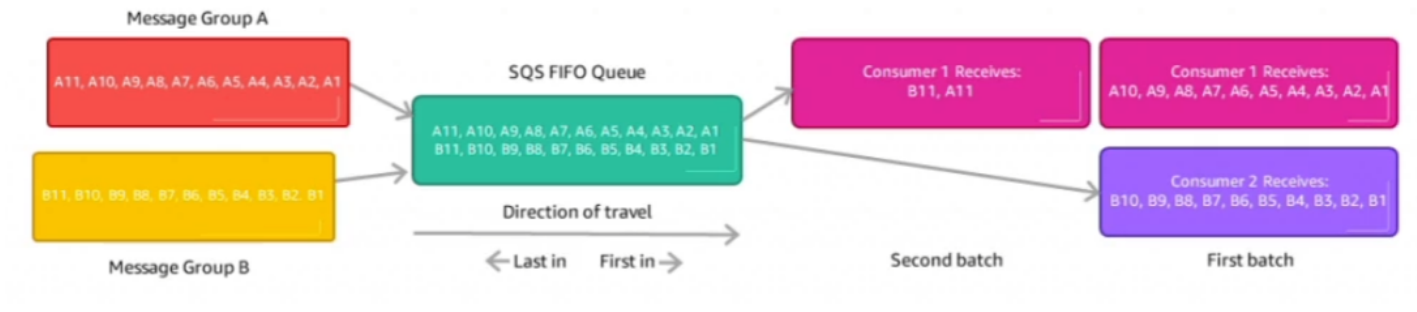


- Imagine you have 100 trucks (truck_1, truck_2, ..., truck_100) on the road sending their GPS positions regularly into AWS
- You want to consume the data in order for each truck, so that you can track their movement accurately.
- How should you send that data into Kinesis?
- **Answer: send using a "Partition Key" value of the "truck_id"**
- **The same key will always go to the same shard**

Ordering data into SQS

- For SQS standard, there is no ordering.
- For SQS FIFO , if you don't use a Group ID, messages are consumed in the order they are sent, **with only one consumer**

- You want to scale the number of consumers, but you want messages to be "grouped" when they are related to each other
- Then you use a Group ID (similar to Partition Key in Kinesis)



Kinesis vs SQS ordering

- **Let's assume 100 trucks, 5 kinesis shards, 1 SQS FIFO**
- Kinesis Data Streams:
  - On average you'll have 20 trucks per shard
  - Trucks will have their data ordered within each shard

    - The maximum amount of consumers in parallel we can have is 5
    - Can receive up to 5 MB/s of data

- SQS FIFO
    - You only have one SQS FIFO queue
    - You will have 100 Group ID
    - You can have up to 100 Consumers (due to the 100 Group ID)
    - You have up to 300 messages per second (or 3000 if using batching)

SQS vs SNS vs Kinesis

SQS	SNS	Kinesis

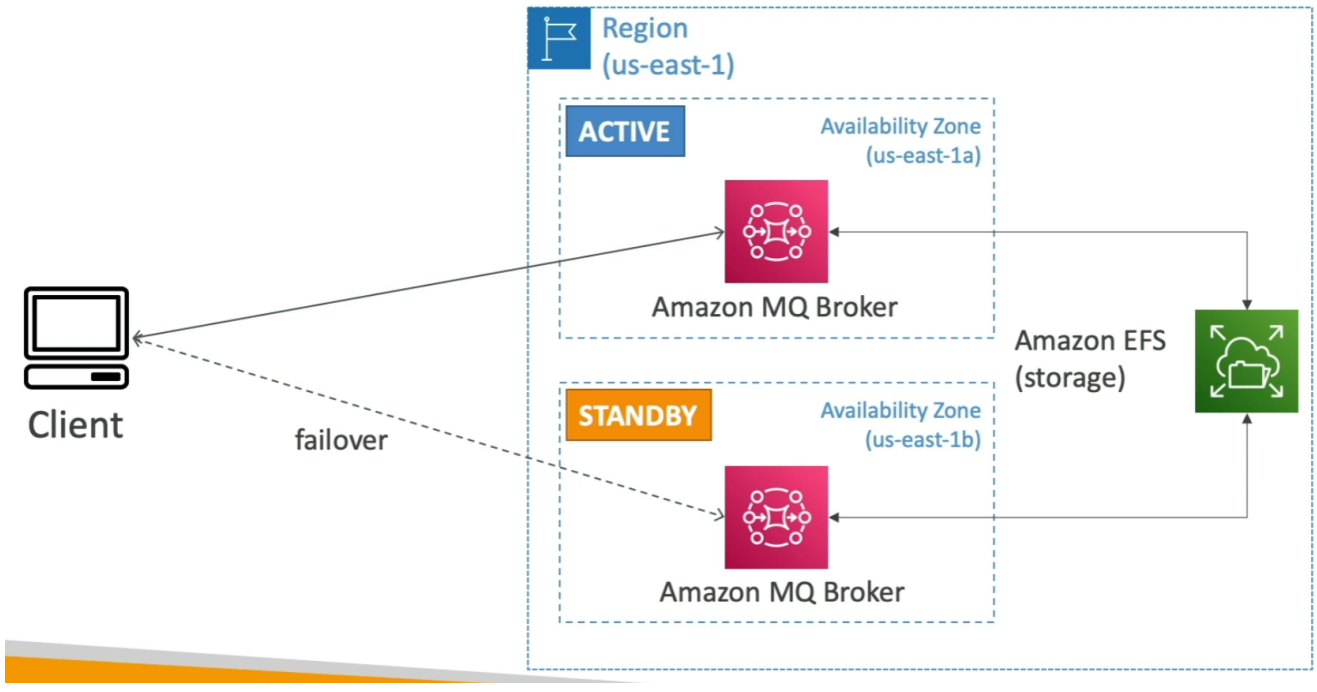
• Consumer "pull data"
• Data is deleted after being consumed
• Can have as many workers(consumers) as we want
• No need to provision throughput
• Ordering guarantees only on FIFO queues
• Individual message delay capability
	• Push data to many subscribers
	• Up to 12500000 subscribers
	• Data is not persisted (lost if not delivered)
	• Pub/Sub
	• Up to 100000 topics
	• No need to provision throughput
	• Integrates with SQS for fan-out architecture pattern
	• FIFO capability for SQS FIFO
		• Standard: pull data
		  ◦ 2MB per shard
		• Enhanced-fan out: push data
		  ◦ 2 MB per shard per consumer
		• Possibility to replay data
		• Meant for real-time big data, analytics and ETL
		• Ordering at the shard level
		• Data expires after X days
		• Provisioned mdoe or on-demand capacity mode

Amazon MQ

- SQS, SNS are "cloud-native" services: proprietary protocols form AWS
- Traditional applications running from on-premises may use open protocols such as: MQTT,AMQP,STOMP,Openwrite,WSS
- When migrating to the cloud, instead of re-engineering the application to use SQS and SNS, we can use Amazon MQ
- Amazon MQ is a managed message broker service for

- Amazon MQ doesn't "scale" as much as SQS / SNS
- Amazon MQ runs on servers, can run in Multi-AZ with failover
- Amazon MQ has both queue feature (~SQS) and topic features (~SNS)

Amazon MQ - High Availability



Amazon Elastic Container Service (Amazon ECS)
Amazon's own container platform

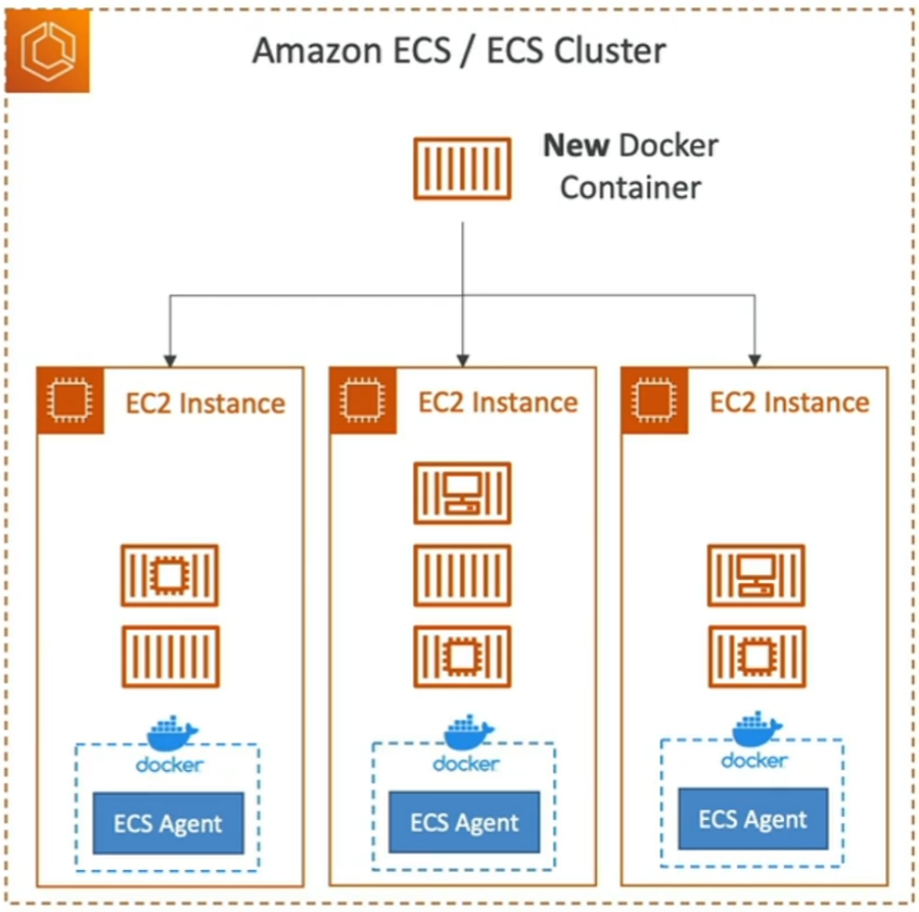
Amzon Elastic Kubernetes Service (Amazon EKS)
Amazon's managed Kubernetes(open source)

AWS Fargate
Amazon's own Serverless container platform
Works with ECS and with EKS

Amazon ECR:
Store container images

Amazon ECS - EC2 Launch Type

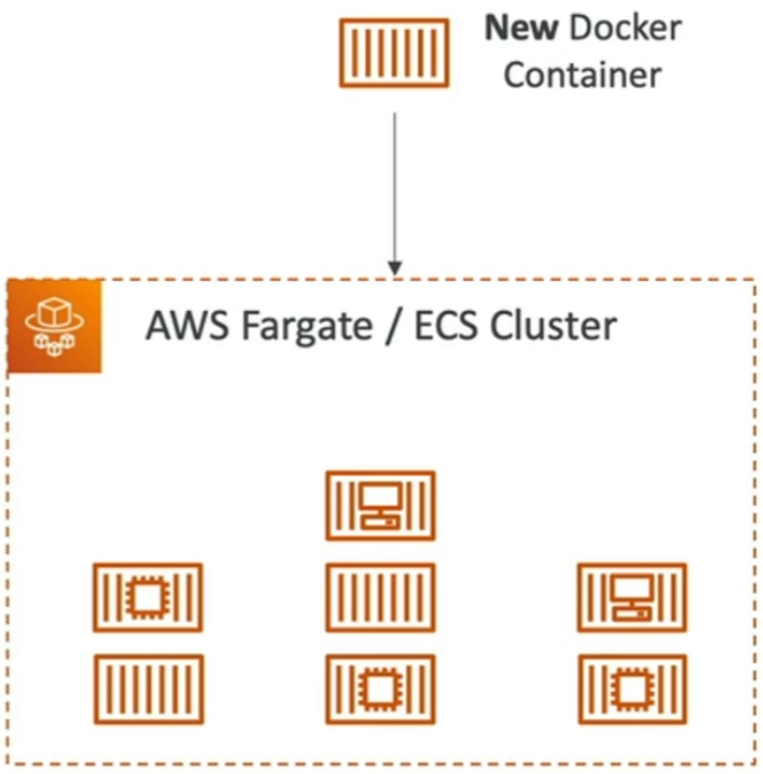
- ECS = Elastic Container Service
- Launch Docker container on AWS = Launch ECS Tasks on ECS Clusters
- EC2 Launch Type: you must provision & maintain the infrastructure (the EC2 instances)
- Each EC2 Instance must run the ECS Agent to register in the ECS Cluster
- AWS takes care of starting / stoping containers



Amazon ECS - Fargate Launch Type

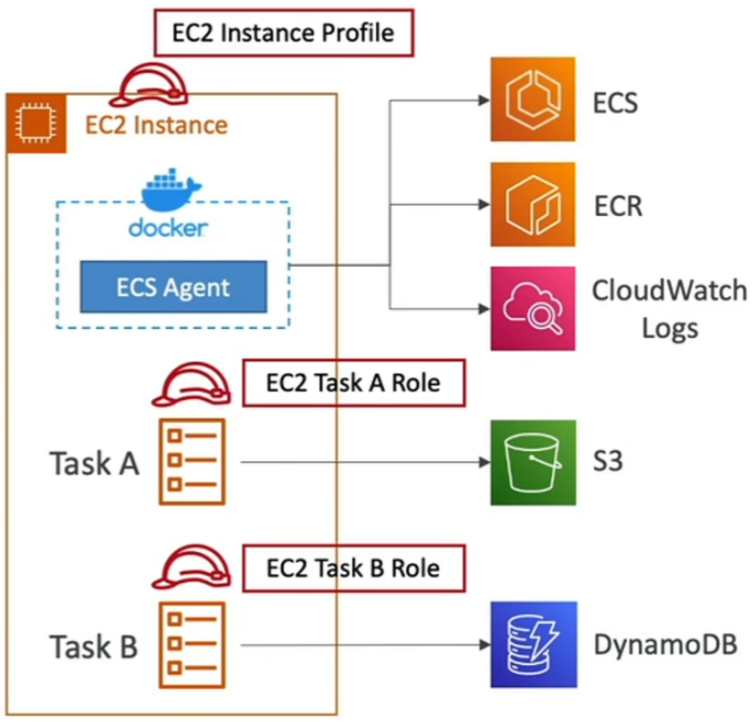
- Launch Docker containers on AWS
- **You do not provision the infrastructure (no EC2 instances to manage)**
- **It's all Serverless!**

- You just create task definitions
- AWS just runs ECS Tasks for you based on the CPU / RAM you need
- To scale, just increase the number of task. Simple - no more EC2 instances



Amazon ECS - IAM Roles for ECS

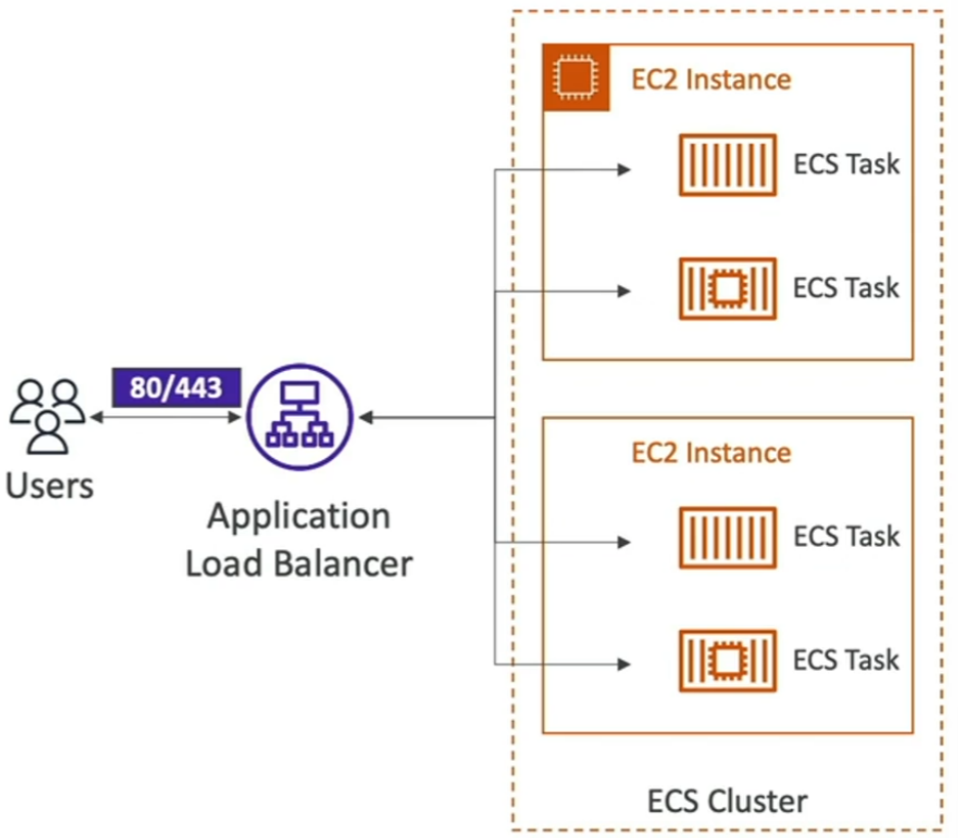
- EC2 Instance Profile (EC2 Launch Type only):
  - Used by the ECS agent
  - Makes API calls to ECS service
  - Send container logs to CloudWatch Logs
  - Pull Docker image from ECR
  - Reference sensitive data in Secrets Manager or SSM Parameter Store
- ECS Task Role:
  - Allows each task to have a specific role
  - Use different roles for the different ECS Services you run
  - Task Role is defined in the task definition
-



Amazon ECS - Load Balancer Integrations

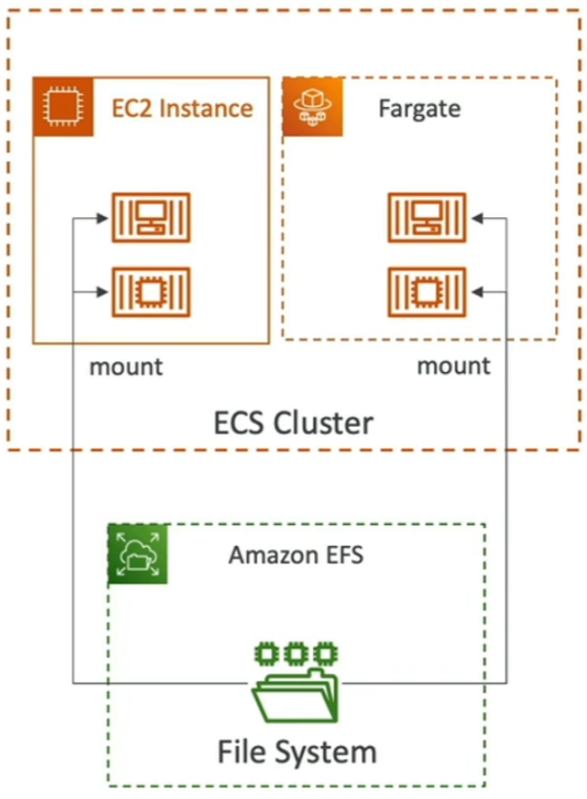
- Application Load Balancer supported and works for most use cases
- Network Load Balancer recommended only for high throughput / high performance use cases,or to pair it with AWS Private Link

- **Elastic Load Balancer** supported but not recommended (no advanced features - no Fargate)



Amazon ECS - Data Volumes (EFS)

- Mount EFS file systems onto ECS tasks
- Works for both **EC2** and **Fargate** launch types
- Tasks running in any AZ will share the same data in the EFS file system
- **Fargate + EFS = Serverless**
- Use cases: persistent multi-AZ shared storage for your containers
- Note:
  - Amazon S3 cannot be mounted as a file system
-



-

ECS Service Auto Scaling

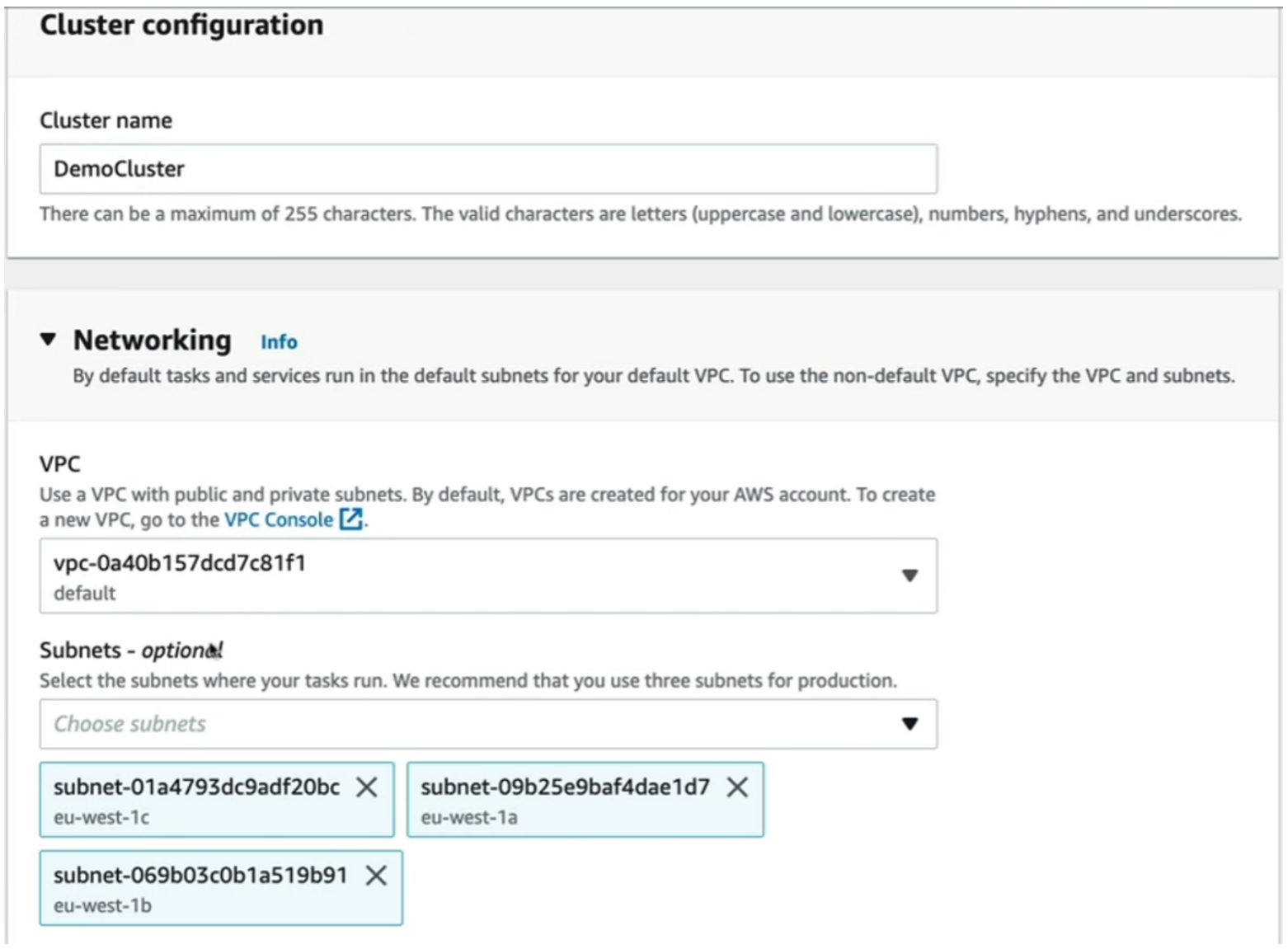
- Automatically increase/decrease the desired number of ECS tasks
- Amazon ECS Auto Scaling uses AWS Application Auto Scaling
  - ECS Service Average CPU Utilization
  - ECS Service Average Memory Utilization - Scale on RAM
  - ALB Request Count Per Target - metric coming from the ALB
- **Target Tracking** - scale based on target value for a specific CloudWatch metric
- **Step Scaling** - scale based on a specified CloudWatch Alarm
- **Scheduled Scaling** - scale based on a specified data/time (predictable changes)

- ECS Service Auto Scaling (task level) ≠EC2 Auto Scaling (EC2 instance level)
- Fargate Auto Scaling is much easier to setup (because Serverless)

EC2 Launch Type - Auto Scaling EC2 Instances

- Accommodate EC2 Service Scaling by adding underlying EC2 Instances
- **Auto Scaling Group Scaling**
  - Scale your ASG based on CPU Utilization
  - Add EC2 instances over time
- **ECS Cluster Capacity Provider**
  - Used to automatically provision and scale the infrastructure for your ECS Tasks
  - Capacity Provider paired with an Auto Scaling Group
  - Add EC2 Instances when you're missing capacity (CPU,RAM...)

Hands on



Amazon EC2 instances

Manual configurations. Use for large workloads with consistent resource demands.

Auto Scaling group (ASG) Info

Use Auto Scaling groups to scale the Amazon EC2 instances in the cluster.

Create new ASG

Operating system/Architecture

Choose the Windows operating system or Linux architecture for your instance.

Amazon Linux 2

EC2 instance type

Choose based on the workloads you plan to run on this cluster.

t2.micro

Desired capacity

Specify the number of instances to launch in your cluster.

Minimum: 0

Maximum: 5

SSH Key pair

Create a key pair in the EC2 console, consisting of a private key and a public key, that you use to prove your identity when connecting to an instance.

None - unable to SSH

External instances using ECS Anywhere

Manual configurations. Use to add data center compute.

Monitoring - *optional* Info

Container Insights is off by default. When you use Container Insights, there is a cost associated with it.

Tags - *optional* Info

Tags help you to identify and organize your clusters.

Cancel Create

Cluster DemoCluster has been created successfully.

Amazon Elastic Container Service > Clusters

All Clusters Info

Clusters (1)

Create cluster

Search clusters

1

DemoCluster

No default found

Services

No services deployed in cluster.

Tasks

No tasks running in this cluster.

EC2 > Auto Scaling groups

Auto Scaling groups (1)

Edit Delete Create an Auto Scaling group

Search your Auto Scaling groups

1

Name	Launch template/configuration	Instances	Status	Desired capacity
Infra-ECS-Cluster	ECSLaunchTemplate_egbArPGLqbbz | Ver	0	-	0

EC2 > Auto Scaling groups > Infra-ECS-Cluster-0cc5f695-1b06-49a1-9d5f-f1c495ba4dc8-ECSAutoScalingGroup-1PMXWABH1NKSG

Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh

Group details

Edit

Desired capacity
0

Minimum capacity
0

Maximum capacity
5

Auto Scaling group name
Infra-ECS-Cluster-0cc5f695-1b06-49a1-9d5f-f1c495ba4dc8-ECSAutoScalingGroup-1PMXWABH1NKSG

Date created
Thu Apr 07 2022 23:24:42 GMT+0100 (Western European Summer Time)

Amazon Resource Name (ARN)
arn:aws:autoscaling:eu-west-1:783768293452:autoScalingGroup:80f4be37-38a3-48d3-9919-cdfd5f7f8f39:autoScalingGroupName/Infra-ECS-Cluster-0cc5f695-1b06-49a1-9d5f-f1c495ba4dc8-ECSAutoScalingGroup-1PMXWABH1NKSG

Group size

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity
1

Minimum capacity
0

Maximum capacity
5

Cancel | Update

EC2 > Auto Scaling groups > Infra-ECS-Cluster-0cc5f695-1b06-49a1-9d5f-f1c495ba4dc8-ECSAutoScalingGroup-1PMXWABH1NKSG

Details | Activity | Automatic scaling | Instance management | Monitoring | Instance refresh

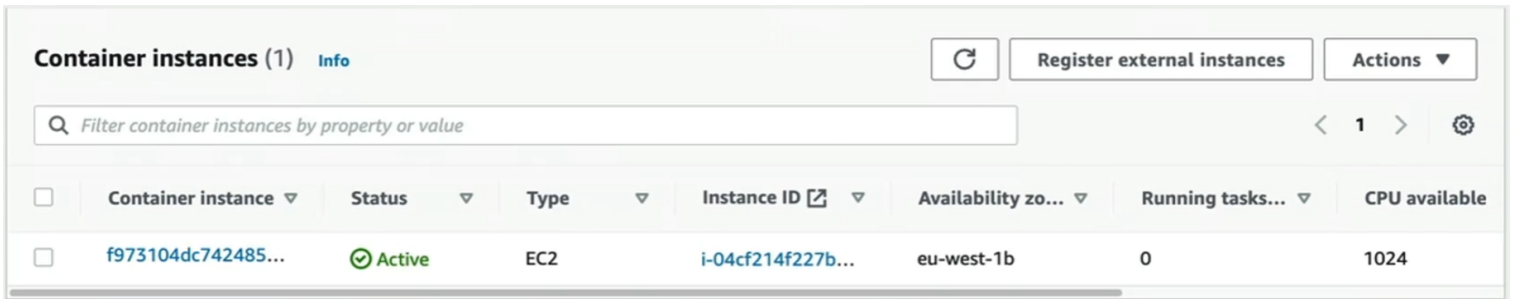
Instances (1)

Actions

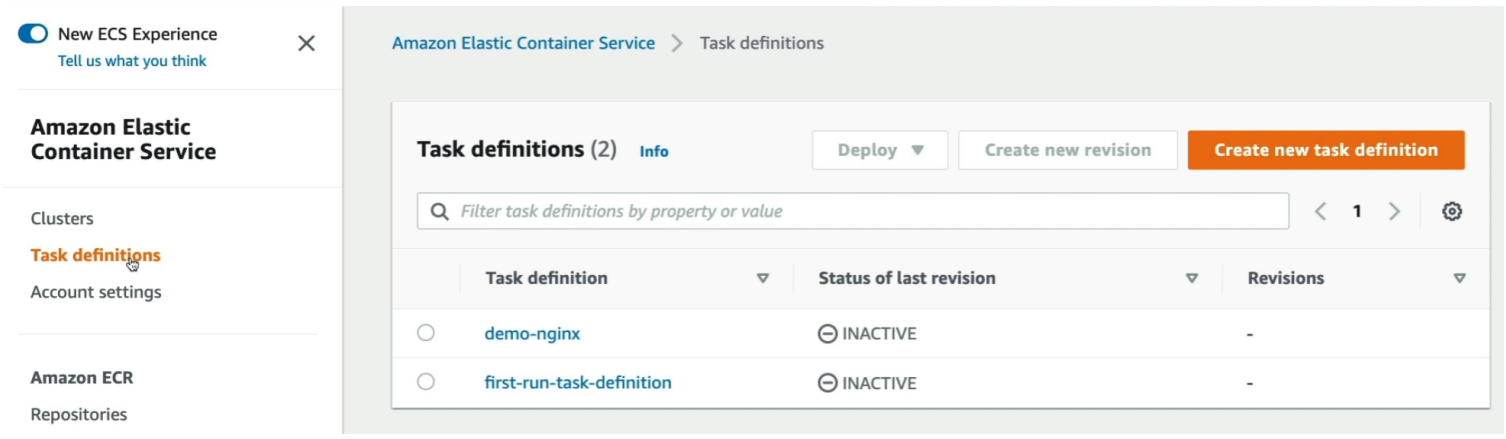
Filter instances

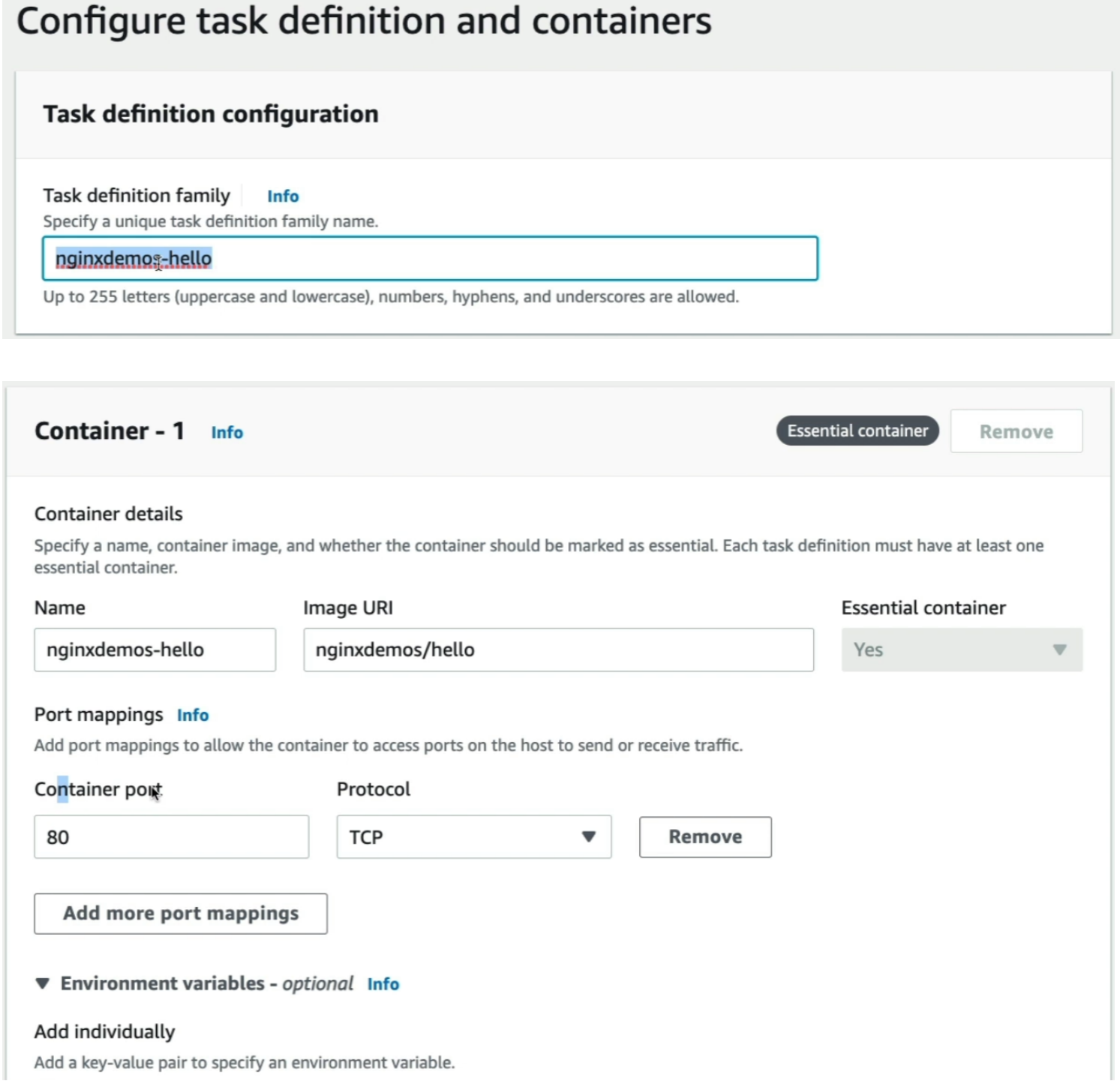
1

Instance ID	Lifecycle	Instance ty...	Weighted capacity	Launch template/config
i-04cf214f227bc1e1e	InService	t2.micro	-	ECSLaunchTemplate_egb



Configure task definition and containers





Environment

Specify the infrastructure requirements for the task definition.

App environment Info

Specify the infrastructure for the task definition.

Add an option

AWS Fargate (serverless)

Operating system/Architecture Info

Linux/X86_64

Task size Info

Specify the amount of CPU and memory to reserve for your task.

CPU: 1 vCPU

Memory: 3 GB

▶ **Container size** - *optional* Info

▼ **Task roles, network mode** - *conditional*

awslogs-region	Value	eu-west-1
awslogs-stream-prefix	Value	ecs
awslogs-create-group	Value	true

Remove

Add

☐ Use trace collection Info

Amazon ECS creates an AWS Distro for OpenTelemetry sidecar to route traces from your application to AWS X-Ray. See pricing information on AWS X-Ray.

☐ Use metric collection Info Preview

Amazon ECS creates an AWS Distro for OpenTelemetry sidecar to route custom container and application metrics to Amazon CloudWatch or Amazon Managed Service for Prometheus.

▶ **Tags** - *optional* Info

Tags help you to identify and organize your task definitions.

Cancel Previous Next

Task definition configuration

Task definition family

nginxdemos-hello

Container - 1

Essential container

Container details

Name	Image URI	Essential container
nginxdemos-hello	nginxdemos/hello	Yes

Port mappings

Host port:container port/protocol

-:80/tcp

▶ **Environment variables** - *optional*

▶ **Container size** - *optional*

Amazon CloudWatch

awslogs-group	Value	/ecs/nginxdemos-hello
awslogs-region	Value	eu-west-1
awslogs-stream-prefix	Value	ecs
awslogs-create-group	Value	true

Trace collection

Off

Metric collection

Off

Tags (1)

Key	Value
ecs:taskDefinition:createdFrom	ecs-console-v2

Cancel Previous Create

EC2 > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info

alb-ecs-sg

Name cannot be edited after creation.

Description Info

ALB for ECS SG

VPC Info

vpc-0a40b157dcd7c81f1

Inbound rules Info

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
Custom TCP	TCP	80	Anywh… 0.0.0.0/0		Delete
Custom TCP	TCP	80	Anywh… ::/0		Delete

Add rule

Outbound rules Info

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info	
All traffic	All	All	Custom 0.0.0.0/0		Delete

Add rule

Tags - *optional*

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag

You can add up to 50 more tag

Cancel Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info

nginx-demo-sg

Name cannot be edited after creation.

Description Info

Allows SSH access to developers

VPC Info

vpc-0a40b157dcd7c81f1

Inbound rules Info

Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
All TCP	TCP	0 - 65535	Custom sg-02ad0dadc206f9b37	Allow traffic from the ALB	Delete

Add rule

Outbound rules Info

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info	
All traffic	All	All	Custom 0.0.0.0/0		Delete

Add rule

Tags - *optional*

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag

You can add up to 50 more tag

Cancel | Create security group

Task definition successfully created

nginxdemos-hello has been successfully created. You can use this task definition to deploy a service or run a task.

Amazon Elastic Container Service > Clusters

All Clusters Info

Clusters (1) | Create cluster

Search clusters | 1

DemoCluster

No default found

Services	Tasks
No services deployed in cluster.	No tasks running in this cluster.

Services | Tasks | Infrastructure | Metrics | Tags

Services (0) | Edit | Delete | Deploy

Filter services by value | 1

Service name	ARN	Status	Deployments and tasks	Task definiti...	Revision	Laun...

No services

No services to display.

Deploy

Application type **Info**

Specify what type of application you want to run.

- (•) **Service**
  Launch a group of tasks handling a long-running computing work that can be stopped and restarted. For example, a web application.
- ( ) **Task**
  Launch a standalone task that runs and terminates. For example, a batch job.

Task definition

Select an existing task definition. To create a new task definition, go to Task definitions.

- [ ] Specify revision manually
  Manually input revision instead of choosing from the 100 most recent revisions for the selected task definition family.

Family: nginxdemos-hello

Revision: 1

Service name

Assign a unique name for this service.

Desired tasks

Specify the number of tasks to launch.

1

Load balancing - *optional*

Load balancer type **Info**

Configure a load balancer to distribute incoming traffic across the tasks running in your service.

Application Load Balancer

Application Load Balancer

Specify whether to create a new load balancer or choose an existing one.

- (•) Create a new load balancer
- ( ) Use an existing load balancer

Load balancer name

Assign a unique name for the load balancer.

DemoALBForECS

Listener **Info**

Specify the port and protocol that the load balancer will listen for connection requests on.

Port: 80

Protocol: HTTP

Target group **Info**

Create a target group that the load balancer will use to route requests to the tasks in your service.

Target group name: nginx-ecs

Protocol: HTTP

Health check path **Info**

/

Health check grace period **Info**

20 seconds

Subnets

Choose the subnets within the VPC that the task scheduler should consider for placement.

Choose subnets

subnet-01a4793dc9adf20bc
eu-west-1c

subnet-09b25e9baf4dae1d7
eu-west-1a

subnet-069b03c0b1a519b91
eu-west-1b

Security group Info

Choose an existing security group or create a new security group.

- (•) Use an existing security group
- ( ) Create a new security group

Security group name

Choose an existing security group.

sg-0284edb2b9ef37ffe
nginx-demo-sg

Public IP Info

Choose whether to auto-assign a public IP to the task's elastic network interface (ENI).

Enabled

Create Load Balancer | Actions

Filter by tags and attributes or search by keyword

1 to 1 of 1

Name	DNS name	State	VPC ID	Availability Zones
DemoALBForECS	DemoALBForECS-1739327…	Active	vpc-0a40b157dcd7c81f1	eu-west-1c, eu-west-1b…

IPv4 address: Assigned by AWS

Edit subnets

Hosted zone Z32O12XQLNTSW2

Creation time April 8, 2022 at 12:17:09 AM UTC+1

Security

Security groups sg-02ad0dadc206f9b37 , **alb-ecs-sg**
• ALB for ECS SG

Edit security groups

Attributes

Load balancer: DemoALBForECS

Description | **Listeners** | Monitoring | Integrated services | Tags

Listeners listen for connection requests using their protocol and port. You can add, remove, or update listeners and listener rules.

To view and edit listener attributes, select the listener and choose Edit.

Add listener | Edit | Delete

Listener ID	Security policy	SSL Certificate	Rules
HTTP : 80 arn...e570b9a7cf819ca7	N/A	N/A	Default: forwarding to nginx-ecs View/edit rules

demoalbforecs-1739327507.eu-west-1.elb.amazonaws.com

NGINX

Server address: 172.31.38.1:80

Server name: ip-172-31-38-1.eu-west-1.compute.internal

Date: 07/Apr/2022:23:34:40 +0000

URI: /

Deployment configuration

Task definition

Select an existing task definition. To create a new task definition, go to Task definitions.

Specify revision manually

Manually input revision instead of choosing from the 100 most recent revisions for the selected task definition family.

Family: nginxdemos-hello

Revision: 1

Desired tasks

Specify the number of tasks to launch.

4

Deployment options

Tags - *optional*

Tags help you to identify and organize your clusters.

Cancel Update

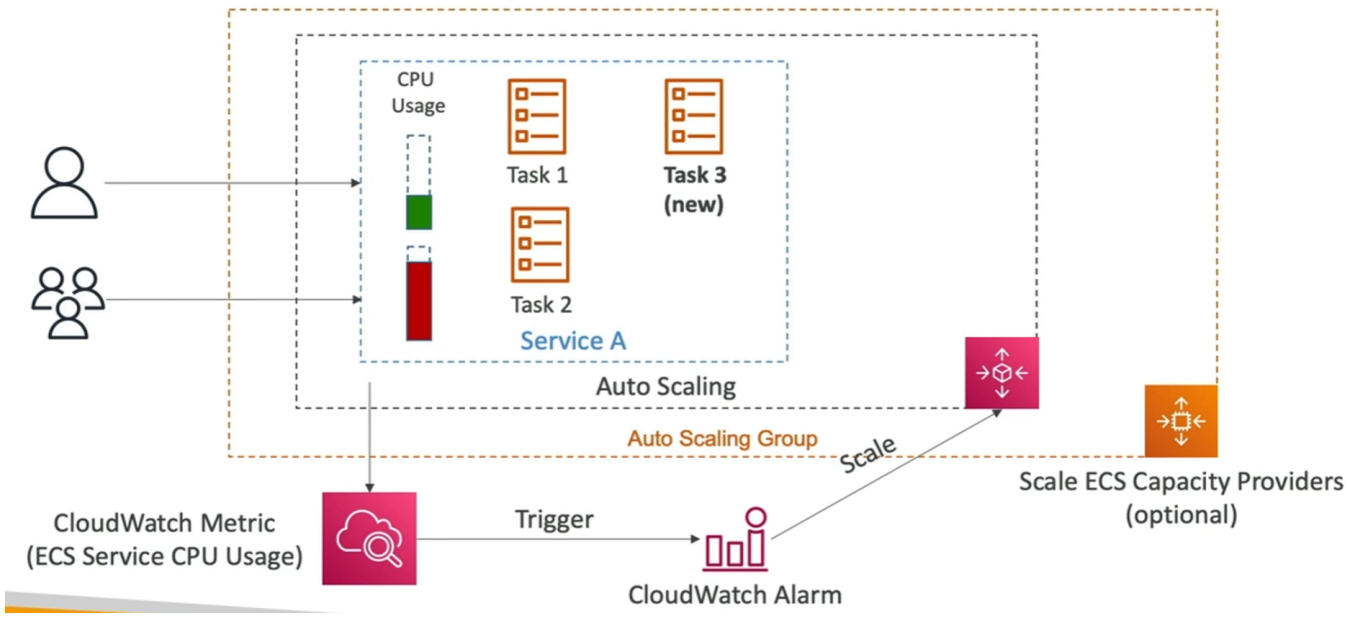
Tasks (4)

Filter tasks by property or value

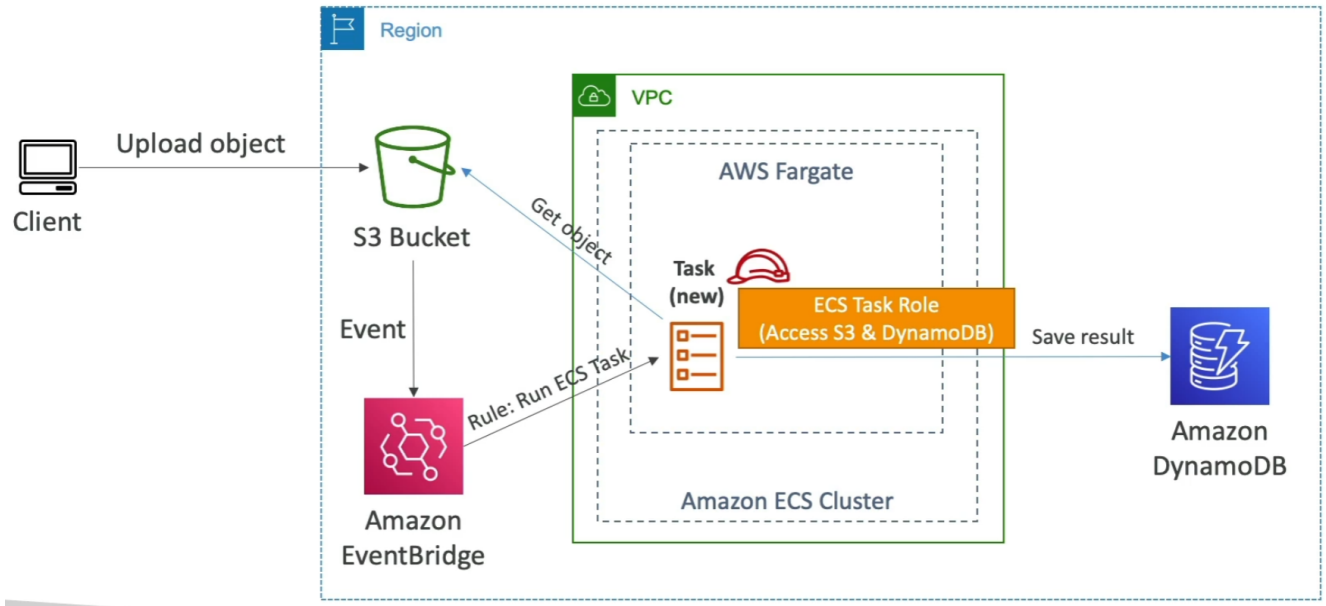
1

Task	Last status	Task definiti...	Revisi...	Health stat...	Started...	Container instan...	Launch type
6db6b...	Running	nginxdemos-hello	1	Unknown	15 minute...	-	FARGATE
6eb49...	Provisioning	nginxdemos-hello	1	Unknown	-	-	FARGATE
72963...	Pending	nginxdemos-hello	1	Unknown	-	-	FARGATE
759f92...	Pending	nginxdemos-hello	1	Unknown	-	-	FARGATE

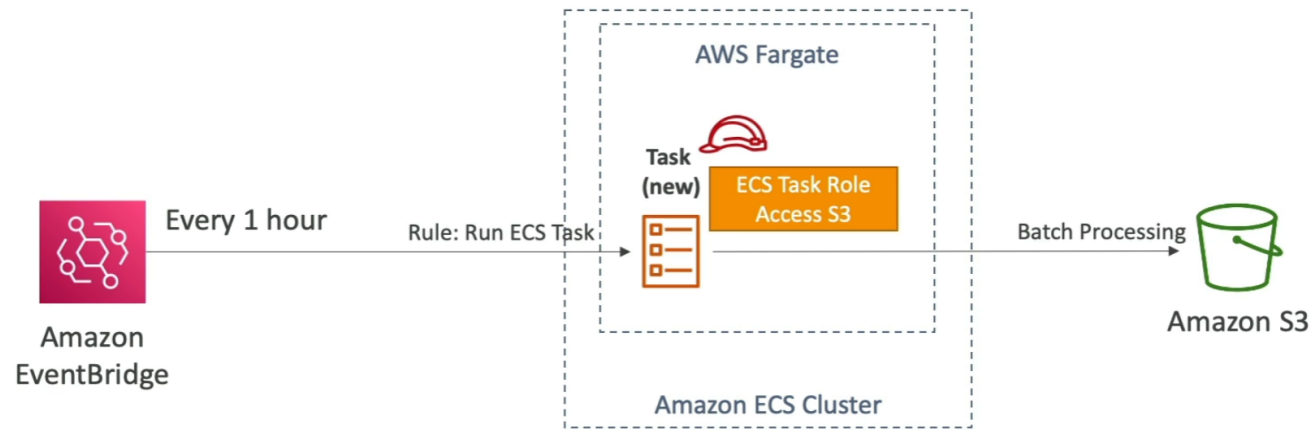
ECS Scaling - Service CPU Usage Example



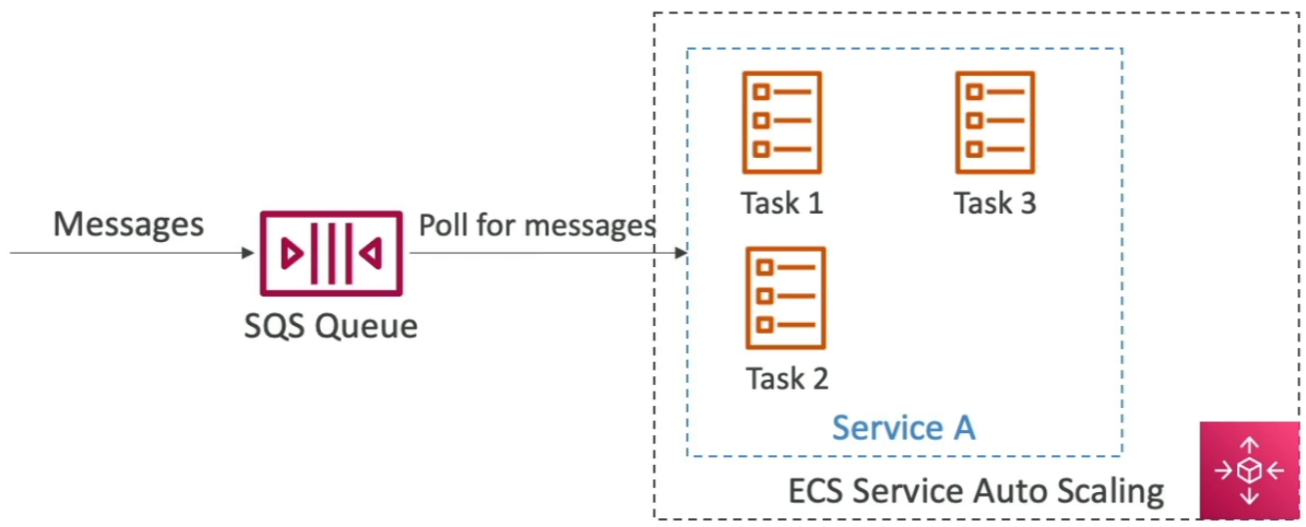
ECS tasks invoked by Event Bridge



ECS tasks invoked by Event Bridge Schedule

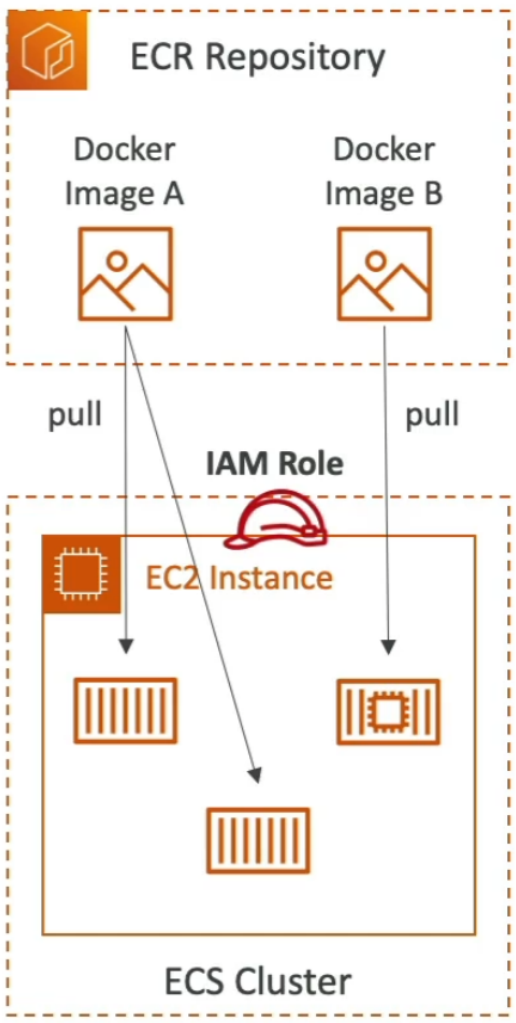


ECS - SQS Queue Example



Amazon ECR

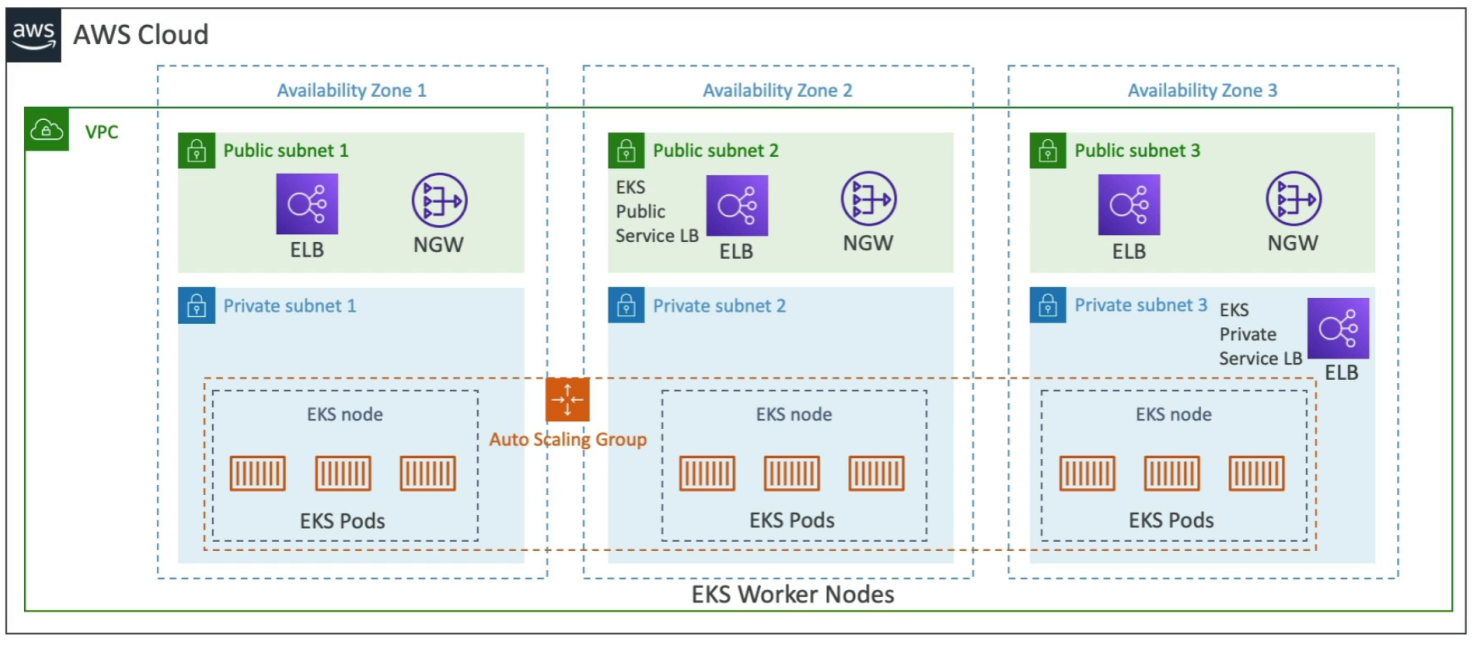
- ECR = Elastic Container Registry
- Store and manage Docker images on AWS
- Private and Public repository (Amazon ECR Public Gallery)
- Fully integrated with ECS,backed by Amazon S3
- Access is controlled through IAM (permission errors => policy)
- Supports image vulnerability scanning, versioning, image tags,image lifecycle,...



Amazon EKS Overview

- Amazon EKS = Amazon Elastic **Kubernetes** Service
- It is a way to launch **managed Kubernetes clusters on AWS**
- Kubernetes is an open-source system for automatic deployment, scaling and management of containerized (usually Docker) application
- It's an alternative to ECS, similar goal but different API
- EKS supports **EC2** if you want to deploy worker nodes or **Fargate** to deploy serverless containers
- **Use case**: if your company is already using Kubernetes on-premises or in another cloud, and wants to migrate to AWS using Kubernetes
- Kubernetes is cloud-agnostic (can be used in any cloud - Azure, GCP..)

Amazon EKS - Diagram



Amazon EKS - Node Types

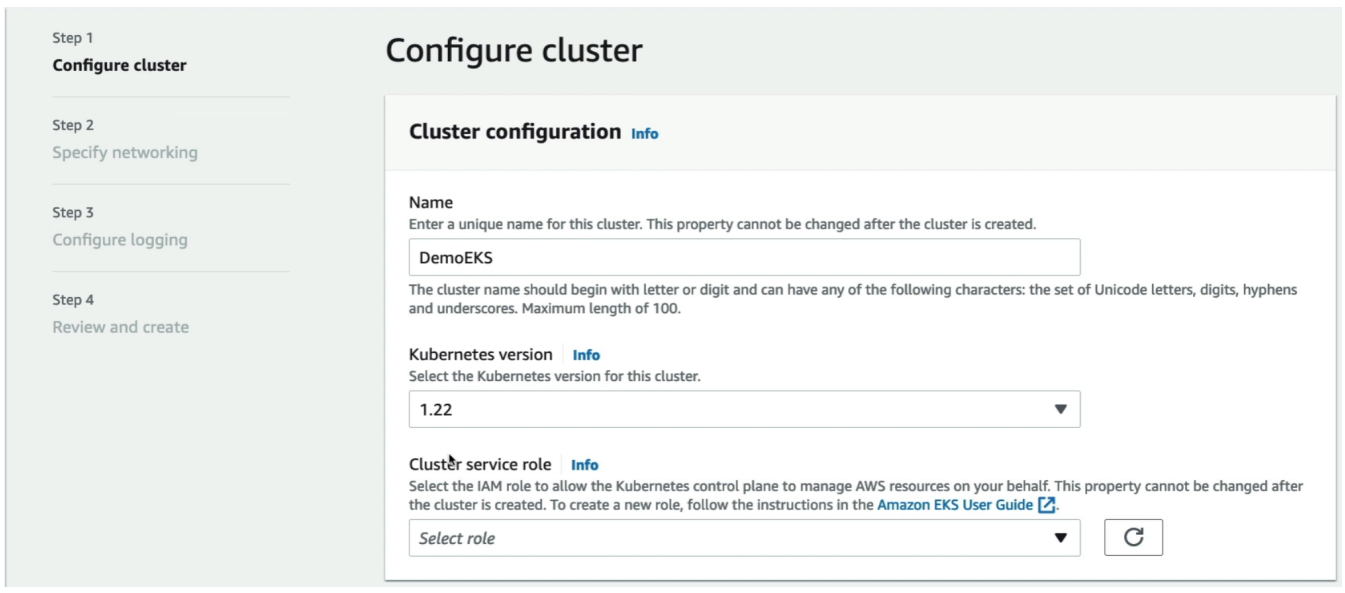
- **Managed Node Groups**
  - Create and manages Nodes (ECs instances) for you
  - Nodes are part of an ASG managed by EKS
  - Supports On-Demand or Spot Instances

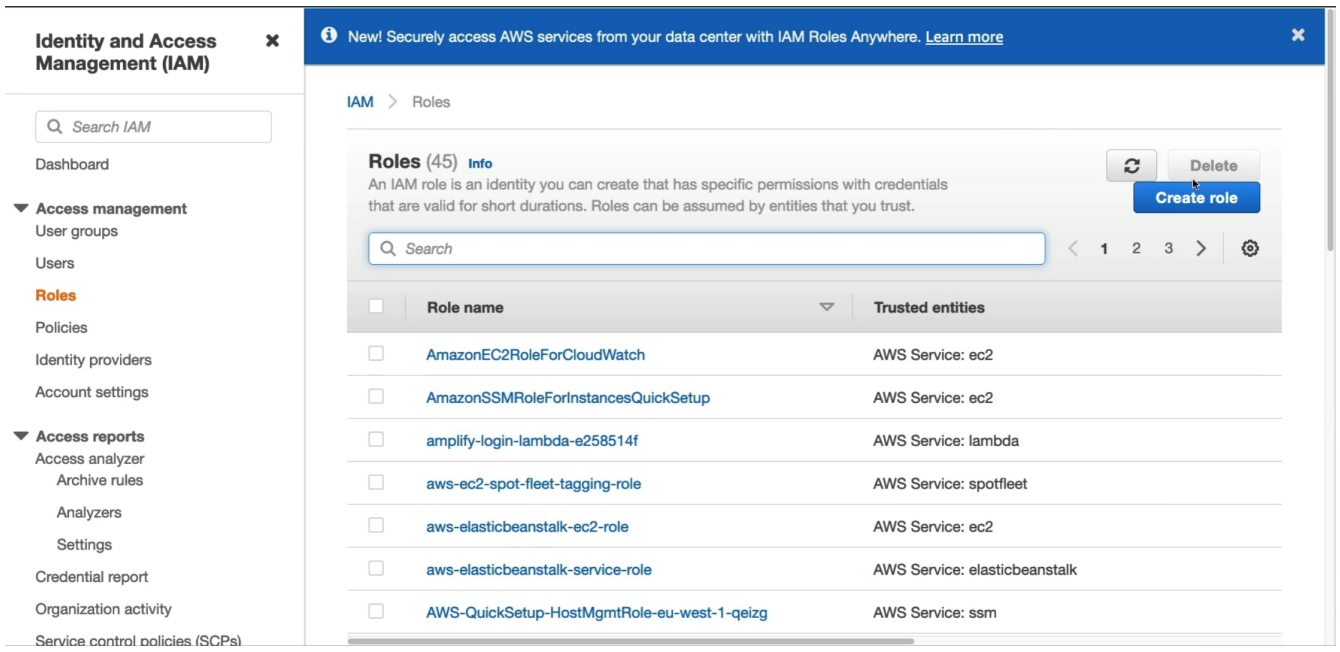
- **Self-Managed Nodes**
  - Nodes created by you and registered to the EKS cluster and managed by an ASG
  - You can use prebuilt AMI - Amazon EKS Optimized AMI
  - Supports On-Demand or Spot Instances
- **AWS Fargate**
  - No maintenance required; no nodes managed

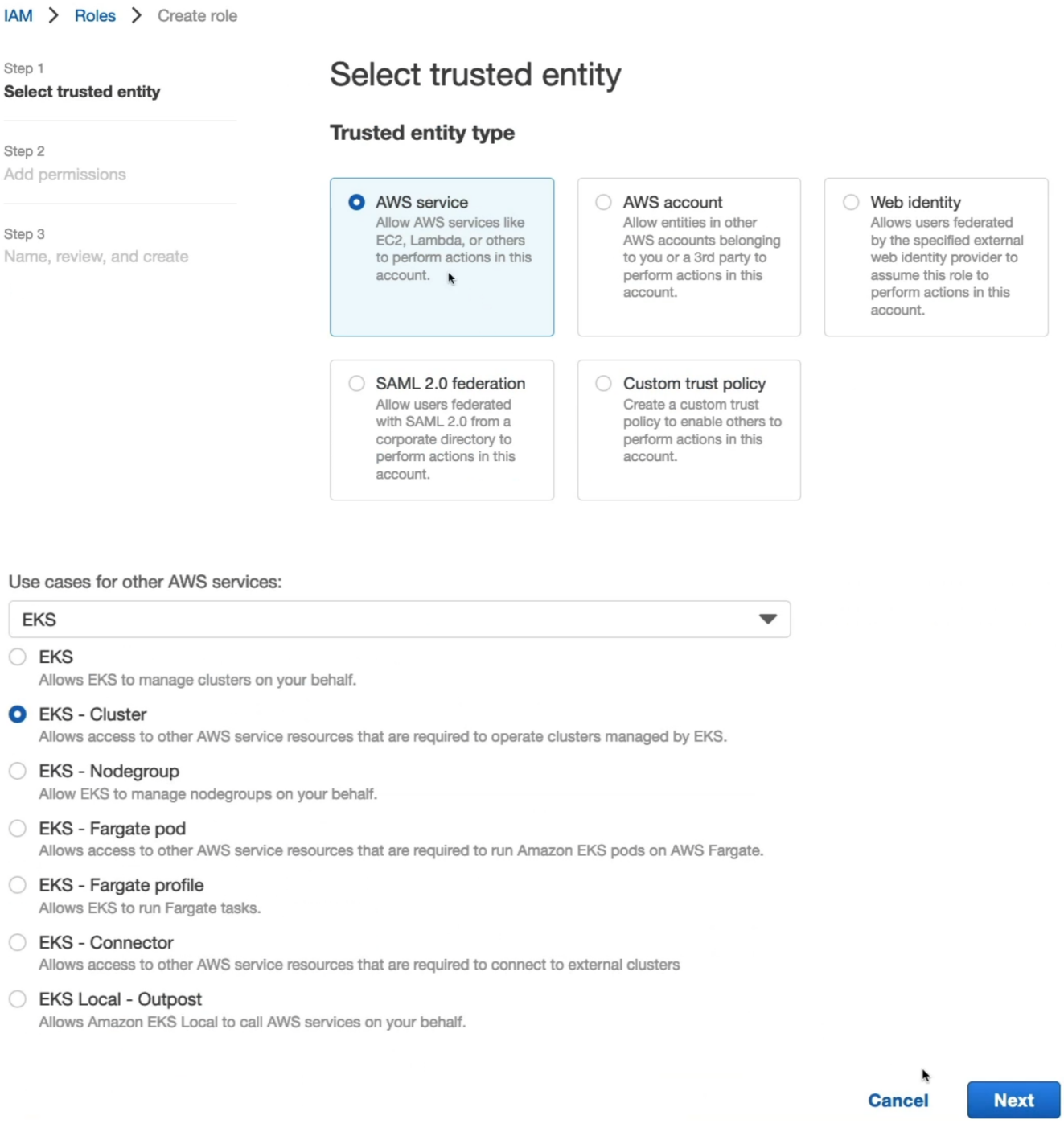
Amazon EKS - Data Volumes

- Need to specify **StorageClass** manifest on your EKS cluster
- Leverages a **Container Storage Interface (CSI)** compliant driver
- **Support for ...**
  - Amazon EBS
  - Amazon EFS (work with Fargate)
  - Amazon FSx for Lustre
  - Amazon FSx for NetApp ONTAP

Hands on







Create IAM Role

Step 2
Add permissions

Step 3
Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

eksClusterRole

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.

Allows access to other AWS service resources that are required to operate clusters managed by EKS.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Step 1: Select trusted entities

Edit

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "eks.amazonaws.com"
                ]
            },
```

Step 2: Add permissions

Edit

Permissions policy summary

Policy name	Type	Attached as
AmazonEKSClusterPolicy	AWS managed	Permissions policy

Tags

Add tags (Optional)
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel | Previous | Create role

Select created IAM Role

DemoEKS

The cluster name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 100.

Kubernetes version Info
Select the Kubernetes version for this cluster.

1.22

Cluster service role Info
Select the IAM role to allow the Kubernetes control plane to manage AWS resources on your behalf. This property cannot be changed after the cluster is created. To create a new role, follow the instructions in the Amazon EKS User Guide.

eksClusterRole

Networking Setting

Networking Info

These properties cannot be changed after the cluster is created.

VPC Info

Select a VPC to use for your EKS cluster resources.To create a new VPC, go to the VPC console.

vpc-0a40b157dcd7c81f1 | Default

Subnets Info

Choose the subnets in your VPC where the control plane may place elastic network interfaces (ENIs) to facilitate communication with your cluster. To create a new subnet, go to the corresponding page in the VPC console.

Select subnets

subnet-01a4793dc9adf20bc ✕ subnet-09b25e9baf4dae1d7 ✕

subnet-069b03c0b1a519b91 ✕

Security groups Info

Choose the security groups to apply to the EKS-managed Elastic Network Interfaces that are created in your worker node subnets. To create a new security group, go to the corresponding page in the VPC console.

Select security groups

Choose cluster IP address family Info

Specify the IP address type for pods and services in your cluster.

- (•) IPv4
- ( ) IPv6

Networking add-ons

Configure add-ons that provide advanced networking functionalities on the cluster.

Amazon VPC CNI Info

Enable pod networking within your cluster.

Version

Select the version for this add-on.

v1.10.1-eksbuild.1

ⓘ This add-on will use the IAM role of the node where it runs. You can change this add-on to use IAM roles for service accounts after cluster creation.

CoreDNS Info

Enable service discovery within your cluster.

Version

Select the version for this add-on.

v1.8.7-eksbuild.1

kube-proxy Info

Enable service networking within your cluster.

Version

Configure logging

Control plane logging Info

Send audit and diagnostic logs from the Amazon EKS control plane to CloudWatch Logs.

- API server
  Logs pertaining to API requests to the cluster.
- Audit
  Logs pertaining to cluster access via the Kubernetes API.
- Authenticator
  Logs pertaining to authentication requests into the cluster.
- Controller manager
  Logs pertaining to state of cluster controllers.
- Scheduler
  Logs pertaining to scheduling decisions.

Cancel | Previous | Next

Step 3: Logging — Edit

Control plane logging

API server: off	Audit: off	Authenticator: off
Controller manager: off	Scheduler: off	

Cancel | Previous | Create

Configure node group

Configure node group Info

A node group is a group of EC2 instances that supply compute capacity to your Amazon EKS cluster. You can add multiple node groups to your cluster.

Node group configuration

These properties cannot be changed after the node group is created.

Name

Assign a unique name for this node group.

DemoNodeGroup

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 63.

Node IAM role Info

Select the IAM role that will be used by the nodes. To create a new role, go to the IAM console.

Select role

The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

Learn more

Use case

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Common use cases

- (•) **EC2**
  Allows EC2 instances to call AWS services on your behalf.
- ( ) **Lambda**
  Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:

Choose a service to view use case

Cancel Next

Add permissions

Permissions policies (Selected 1/777)
Choose one or more policies to attach to your new role.

Create policy

Filter policies by property or policy name and press enter 7 matches 1

"eks" Clear filters

	Policy name	Type	Description
☐	AmazonEKSClusterPolicy	AWS m…	This policy provi
☑	AmazonEKSWorkerNodePolicy	AWS m…	This policy allow
☐	AmazonEKSServicePolicy	AWS m…	This policy allow
☐	AmazonEKS_CNI_Policy	AWS m…	This policy provi
☐	AmazonEKSFargatePodExecutionRolePolicy	AWS m…	Provides access
☐	AmazonEKSLocalOutpostClusterPolicy	AWS m…	This policy provi

Add permissions

Permissions policies (Selected 2/777)
Choose one or more policies to attach to your new role.

Create policy

Filter policies by property or policy name and press enter 1 match 1

"AmazonEC2ContainerRegistryReadOnly" Clear filters

	Policy name	Type	Description
☑	AmazonEC2ContainerRegistryReadOnly	AWS m…	Provides read-only acce

▸ **Set permissions boundary - *optional***

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Step 2: Add permissions

Edit

Permissions policy summary

Policy name	Type	Attached as
AmazonEC2ContainerRegistryReadOnly	AWS managed	Permissions policy
AmazonEKSWorkerNodePolicy	AWS managed	Permissions policy

Tags

Add tags (Optional)

Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel Previous Create role

Node group configuration

These properties cannot be changed after the node group is created.

Name

Assign a unique name for this node group.

DemoNodeGroup

The node group name should begin with letter or digit and can have any of the following characters: the set of Unicode letters, digits, hyphens and underscores. Maximum length of 63.

Node IAM role Info

Select the IAM role that will be used by the nodes. To create a new role, go to the IAM console.

AmazonEKSNodeRole

The selected role must not be used by a self-managed node group as this could lead to a service interruption upon managed node group deletion.

Learn more

Node group compute configuration

These properties cannot be changed after the node group is created.

AMI type Info

Select the EKS-optimized Amazon Machine Image for nodes.

Amazon Linux 2 (AL2_x86_64)

Capacity type

Select the capacity purchase option for this node group.

On-Demand

Instance types Info

Select instance types you prefer for this node group.

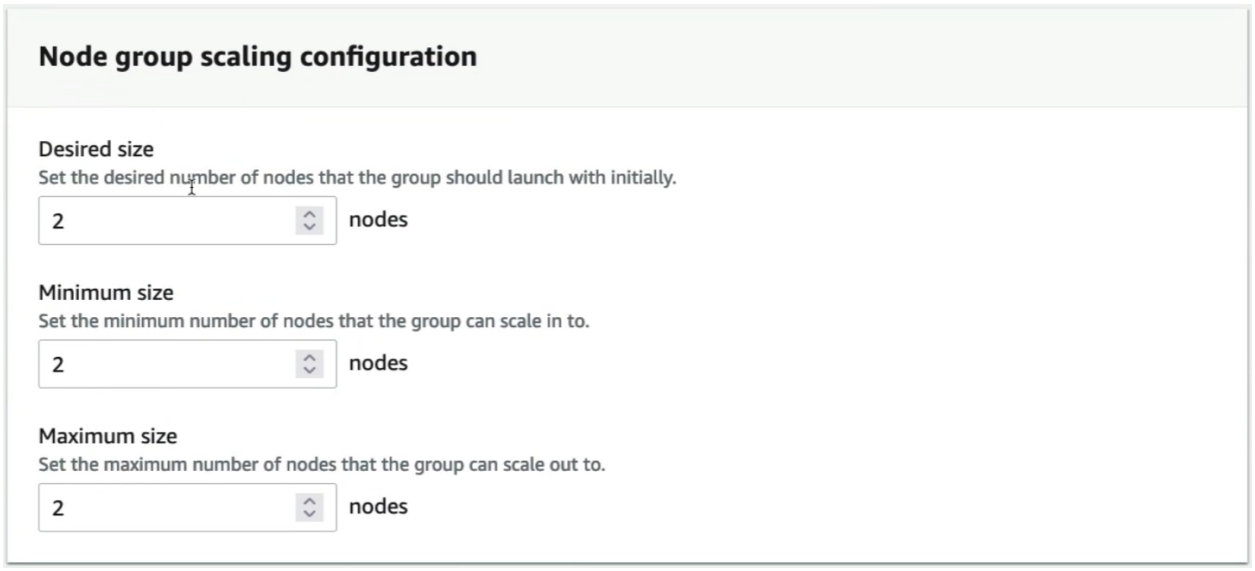
Select

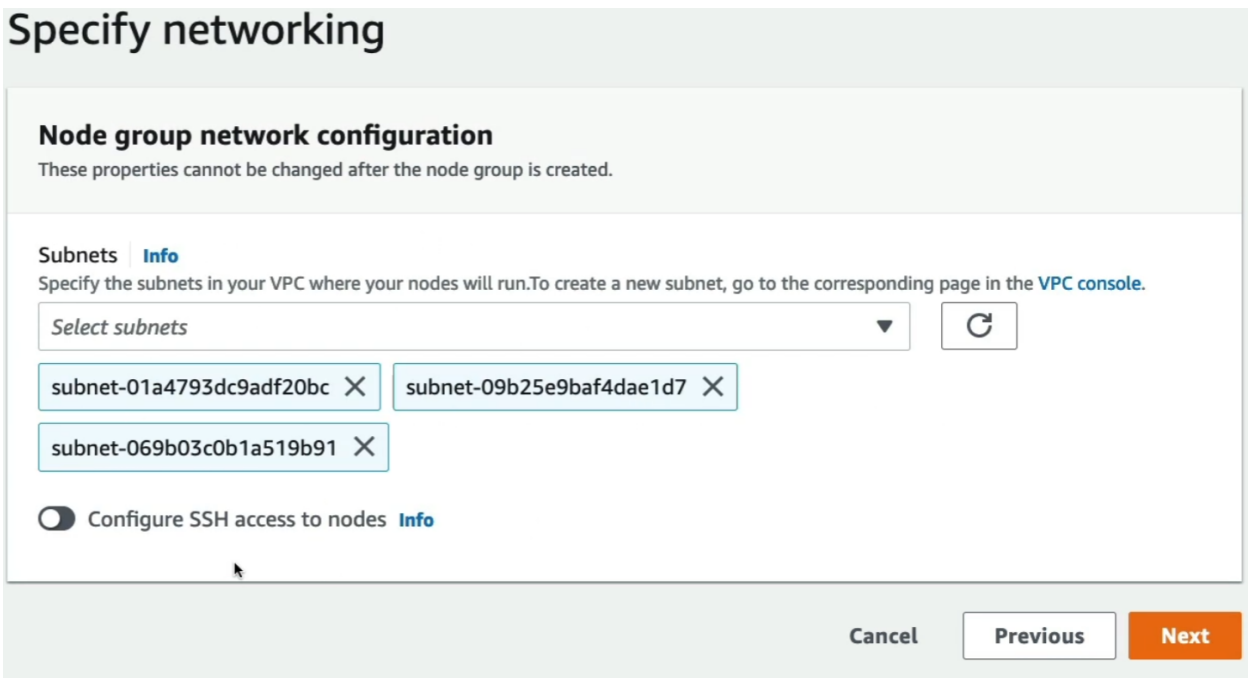
t3.medium
vCPU: Up to 2 vCPUs memory: 4.0 GiB

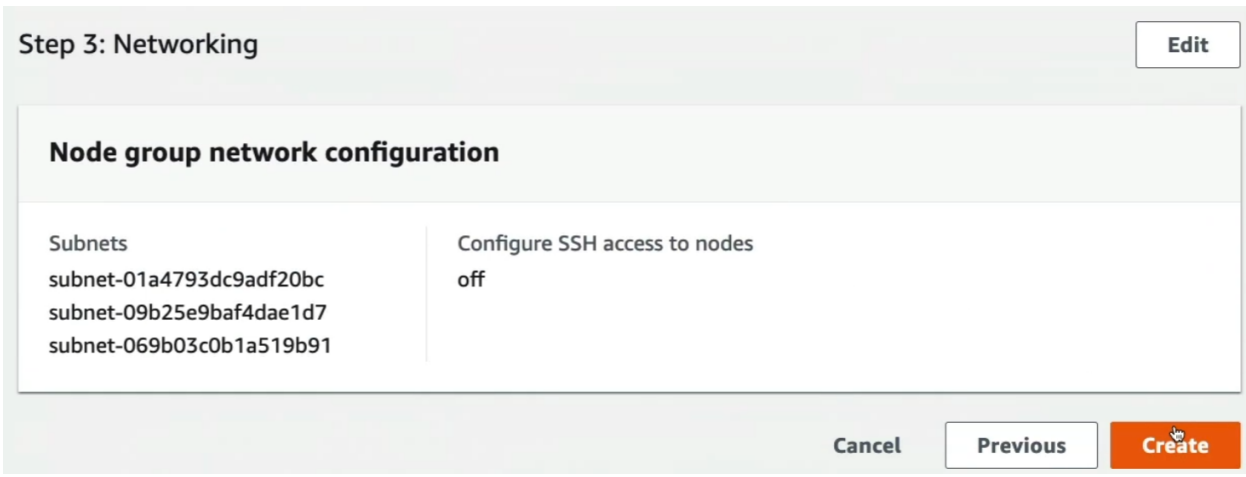
Disk size

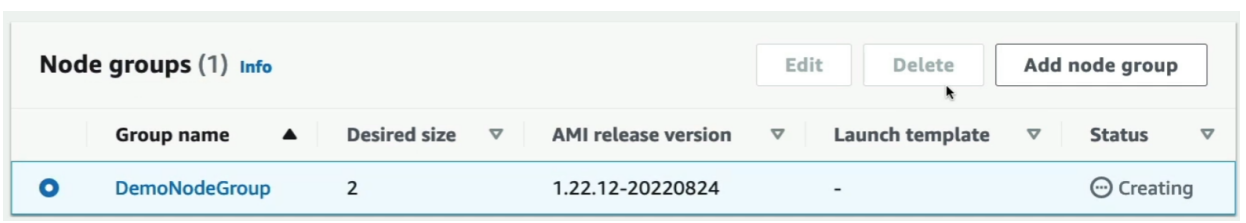
Select the size of the attached EBS volume for each node.

20 GiB





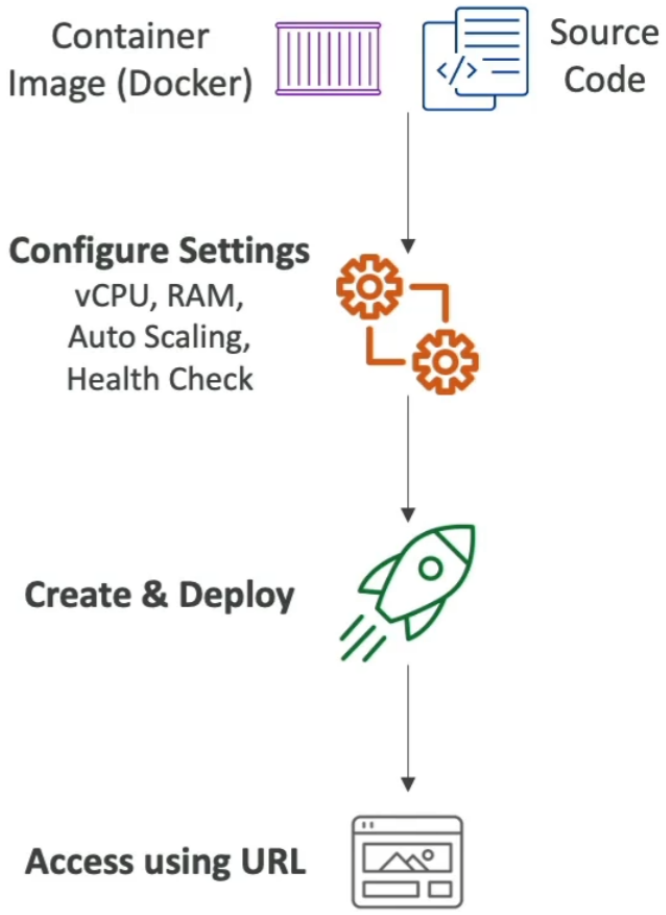




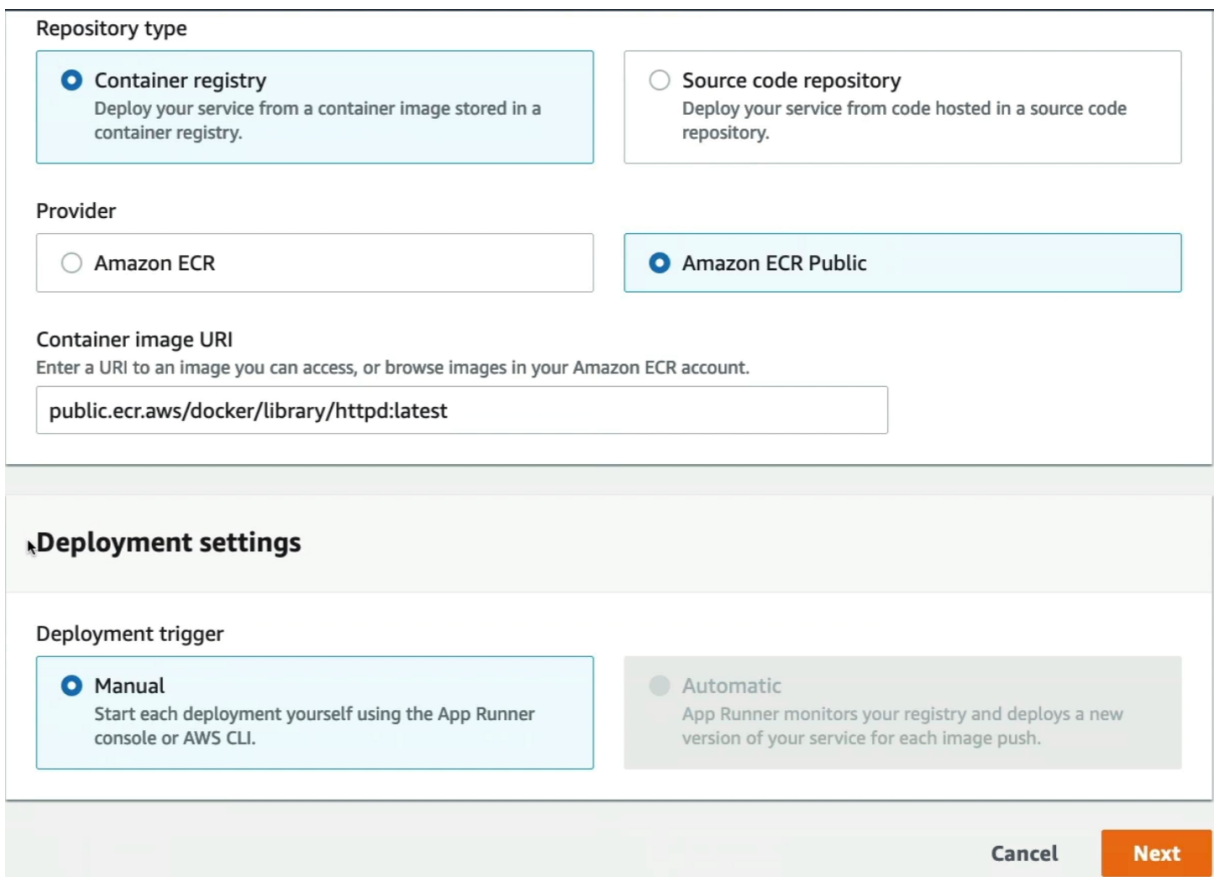
AWS App Runner

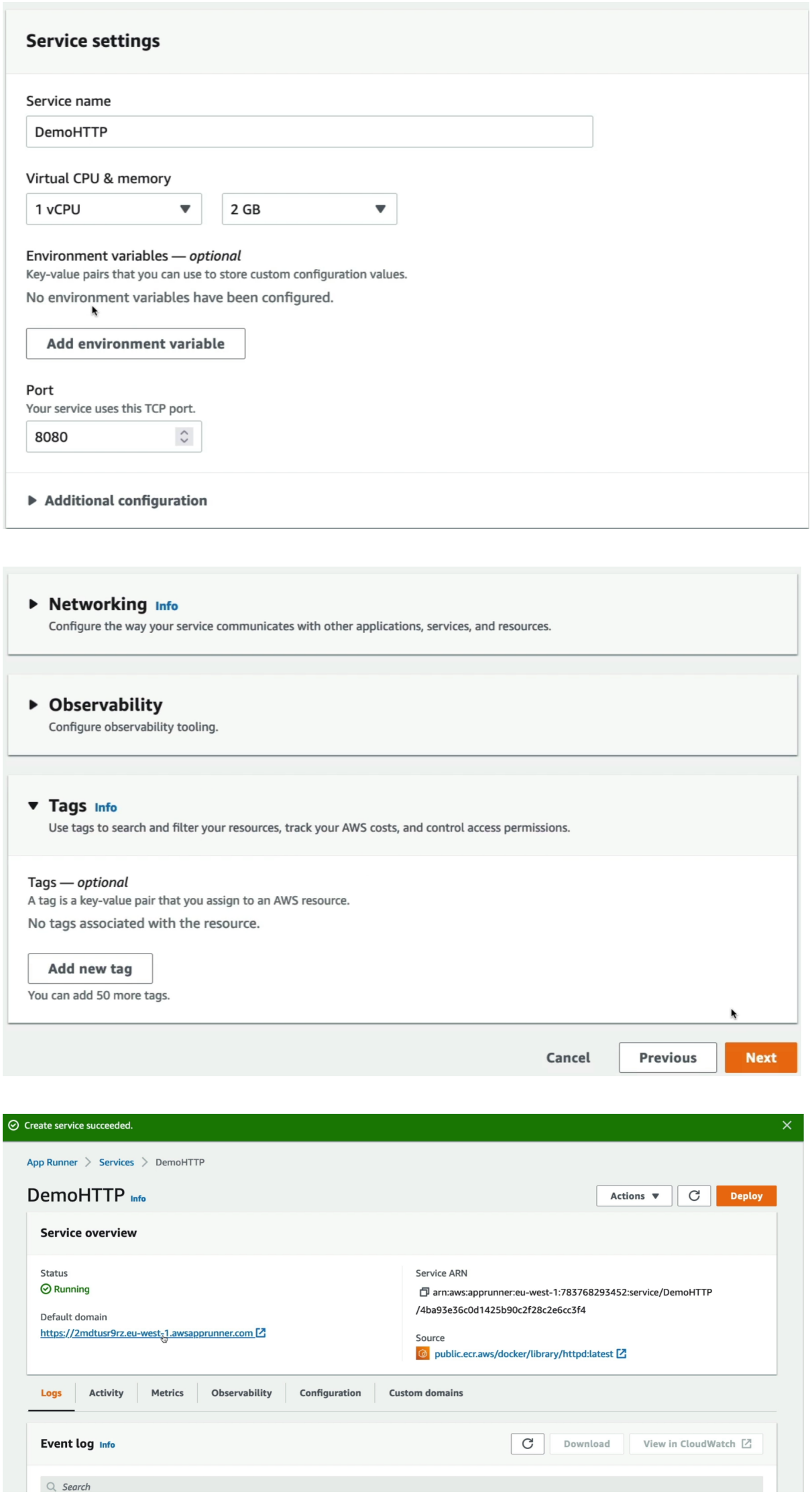
- Fully managed service that makes it easy to deploy web applications and APIs at scale
- No infrastructure experience required
- Start with your source code or container image
- Automatically builds and deploy the web app
- Automatic scaling,high available, load balancer, encryption
- VPC access support
- Connect to database, cache,and message queue services

- Use cases: web apps, APIs, microservices， rapid production deployments



Hands on

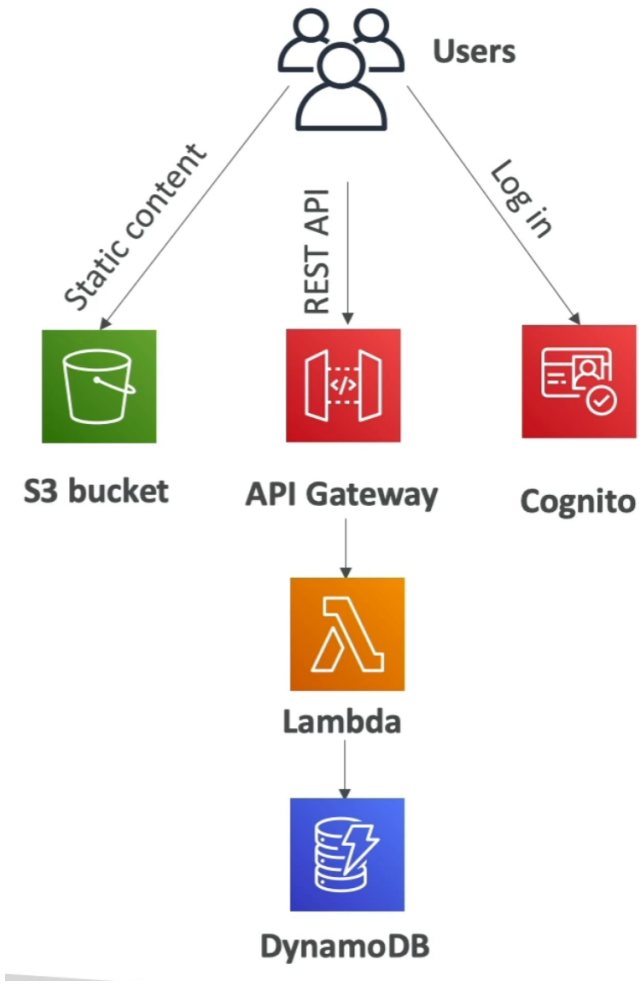




What's serverless?

- 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

Serverless in AWS



- AWS Lambda
- DynamoDB
- AWS Cognito
- AWS API Gateway
- Amazon S3
- AWS SNS & SQS
- AWS Kinesis Data Firehose
- Aurora Serverless
- Step Functions
- Fargate

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

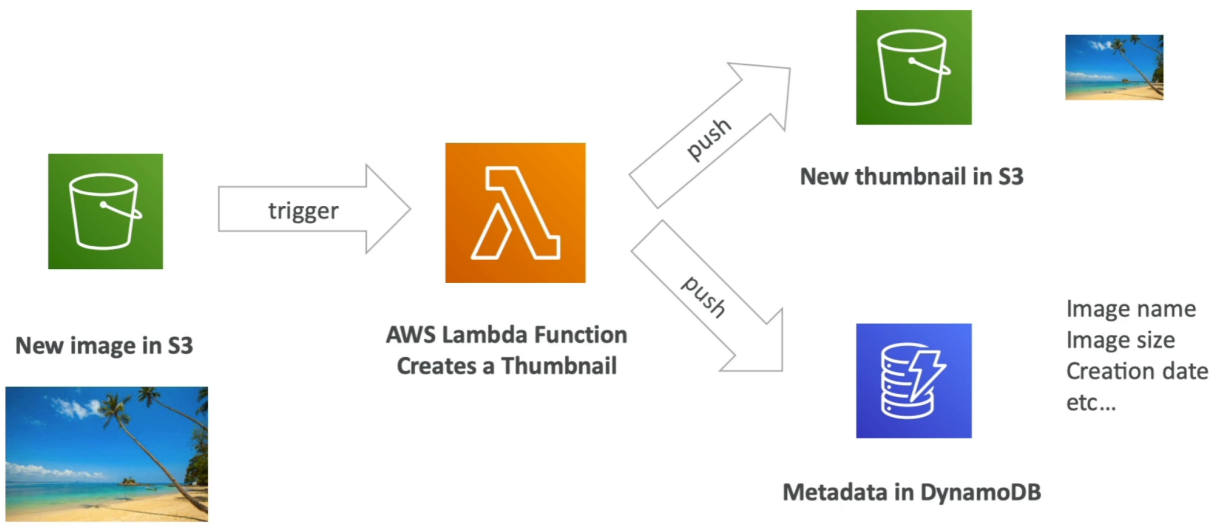
Benefits of AWS Lambda

- 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
- 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

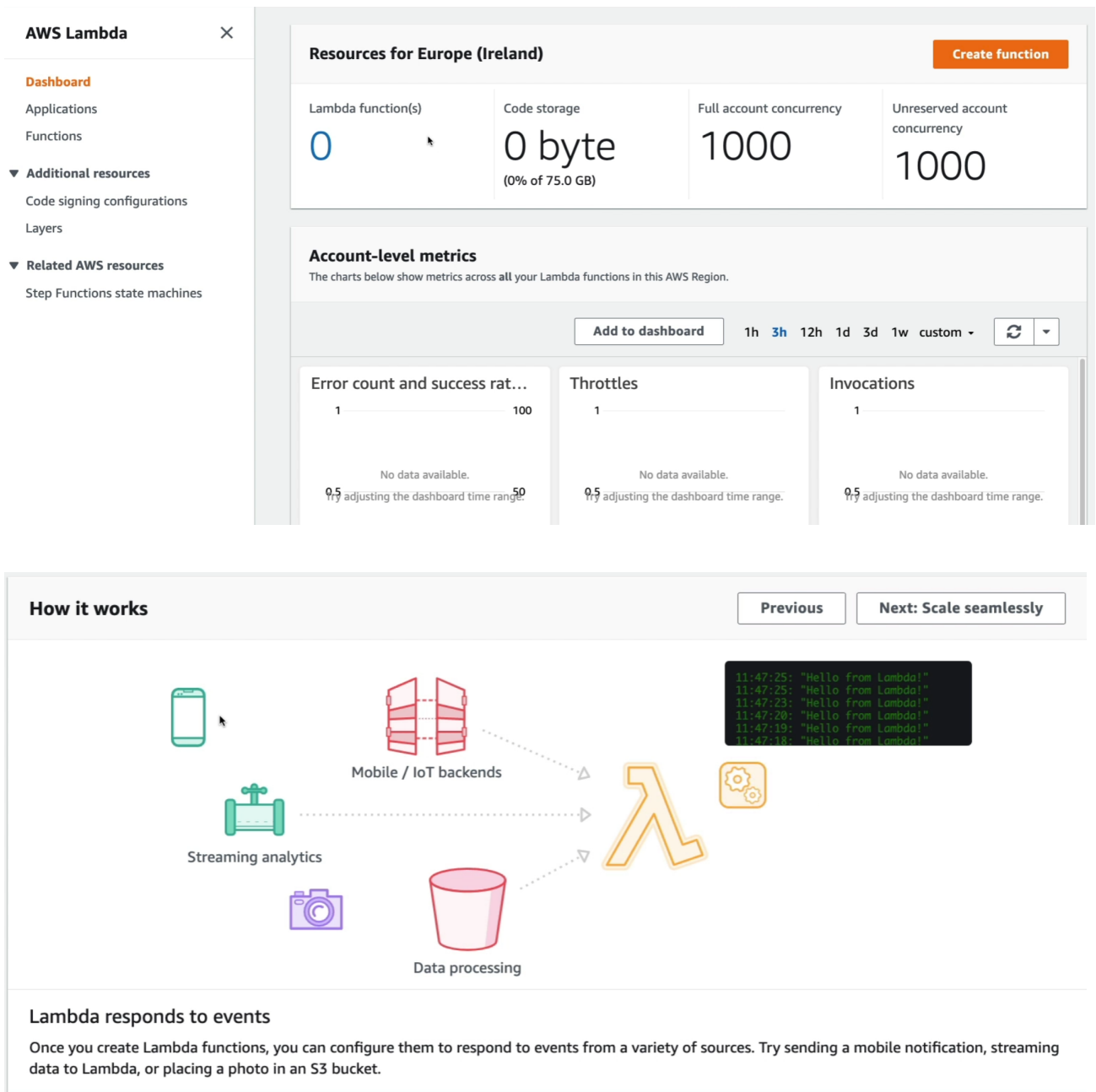
AWS Lambda Integrations Main ones

- Examole: Serverless Thumbnail creation



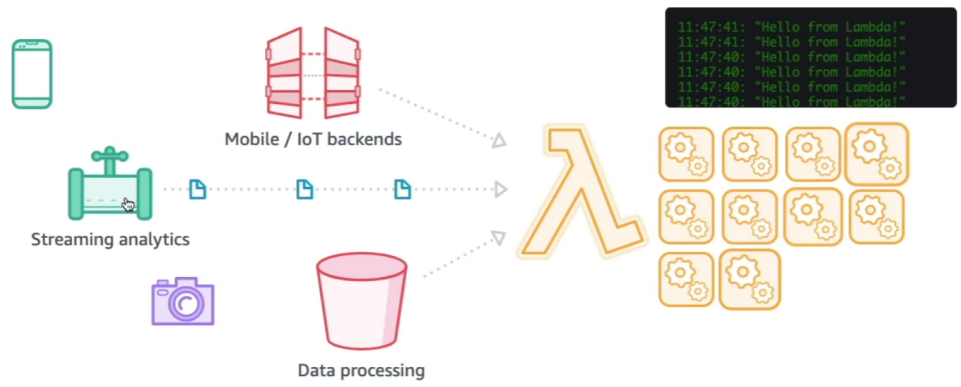
- Example: Serverless CRON Job

Hands on



How it works

Previous | Next: Scale seamlessly



Lambda responds to events

Once you create Lambda functions, you can configure them to respond to events from a variety of sources. Try sending a mobile notification, streaming data to Lambda, or placing a photo in an S3 bucket.

Lambda > Functions > Create function

Create function Info

Choose one of the following options to create your function.

Author from scratch
Start with a simple Hello World example.

Use a blueprint (selected)
Build a Lambda application from sample code and configuration presets for common use cases.

Container image
Select a container image to deploy for your function.

Browse serverless app repository
Deploy a sample Lambda application from the AWS Serverless Application Repository.

Blueprints Info

Export

Filter by tags and attributes or search by keyword — 1 match — 1

Blueprint name: hello-world-python ✕ | Clear filters

Lambda > Functions > Create function > Configure blueprint hello-world-python

Basic information Info

Function name

demo-lambda

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the **IAM console**.

- (•) Create a new role with basic Lambda permissions
- ( ) Use an existing role
- ( ) Create a new role from AWS policy templates

ⓘ Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named demo-lambda-role-eaek950w, with permission to upload logs to Amazon CloudWatch Logs.

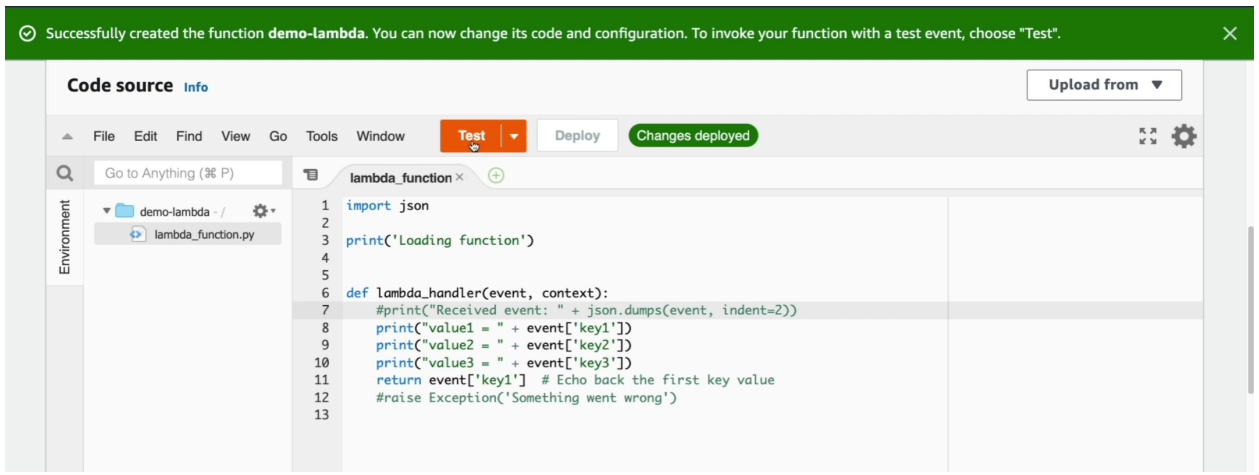
Lambda function code

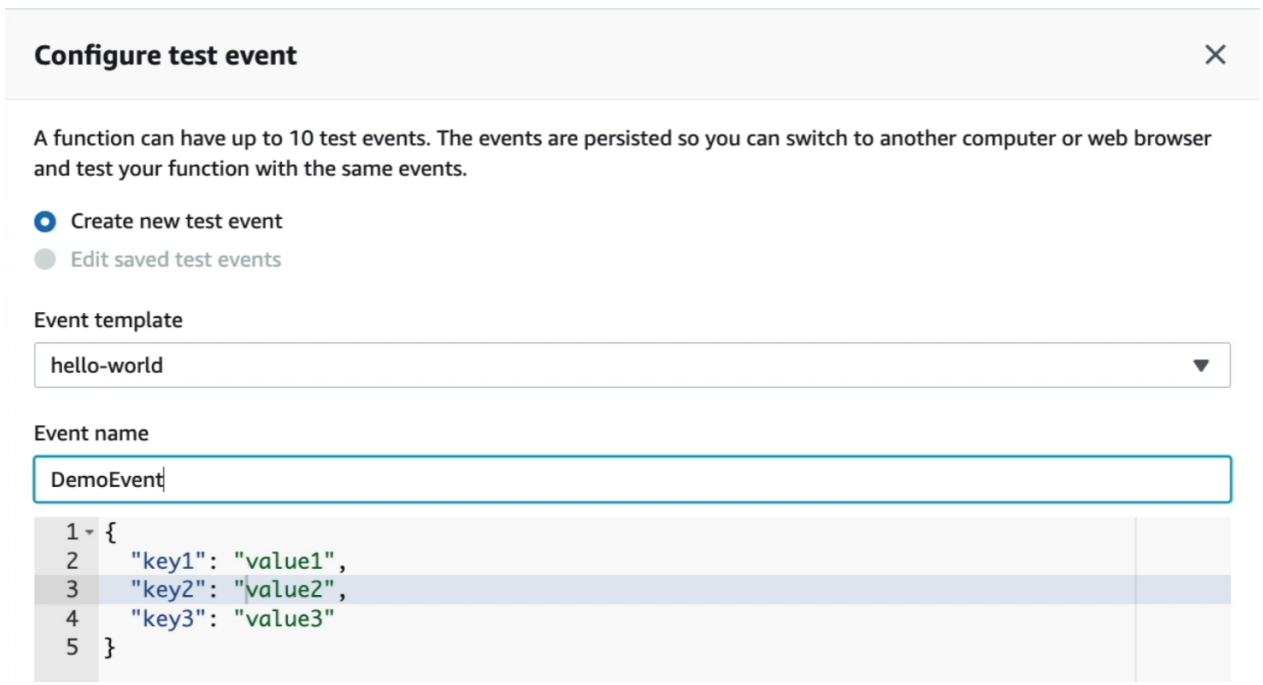
Code is preconfigured by the chosen blueprint. You can configure it after you create the function. **Learn more** about deploying Lambda functions.

Runtime

Python 3.7

```
1  import json
2
3  print('Loading function')
4
5
6  def lambda_handler(event, context):
7      #print("Received event: " + json.dumps(event, indent=2))
8      print("value1 = " + event['key1'])
9      print("value2 = " + event['key2'])
10     print("value3 = " + event['key3'])
11     return event['key1']  # Echo back the first key value
12     #raise Exception('Something went wrong')
13
```





AWS Lambda Limits to Know - per region

- Execution:
  - Memory allocation: 128MB - 10GB (1 MB increments)
  - Maximum execution time: 900 seconds (15 minutes)
  - Environment variables (4KB)
  - Disk capacity in the "function container" (in /tmp): 512MB to 10GB
  - Concurrency executions: 1000 (can be increased)
- Deployment:
  - Lambda function deployment size (compressed .zip): 50 MB
  - Size of uncompressed deployment (code + dependencies): 250MB
  - Can use the /tmp directory to load other files at startup
  - Size of environment variables: 4 KB

Customization At The Edge

- Many modern applications execute some form of the logic at the edge
- **Edge Function:**
  - A code that you write and attach to CloudFront distributions
  - Runs close to your users to minimize latency
- CloudFront provides two types: **CloudFront Functions** & **Lambda@Edge**
- You don't have to manage any servers, deployed globally

- Use case:customize the CDN content
- Pay only for what you use
- Fully serverless

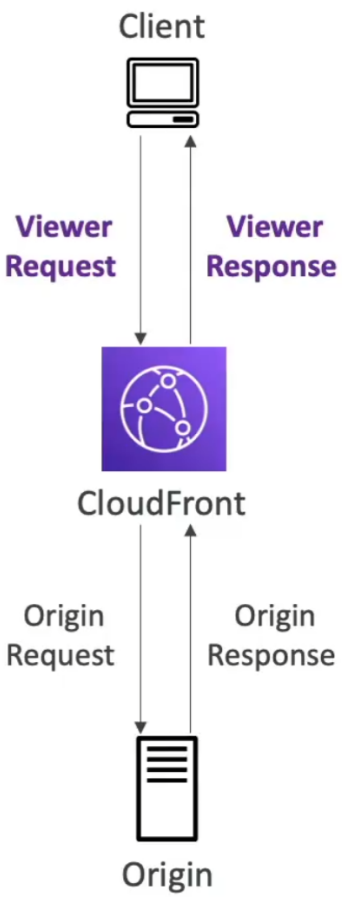
Use Cases

- Website Security and Privacy
- Dynamic Web Application at the Edge
- Search Engine Optimization (SEO)
- Intelligently Route Across Origins and Data Centers
- Bot Mitigation at the Edge

- Real-time Image Transformation
- A/B Testing
- User Authentication and Authorization
- User Prioritization
- User Tracking and Analytics

CloudFront Functions

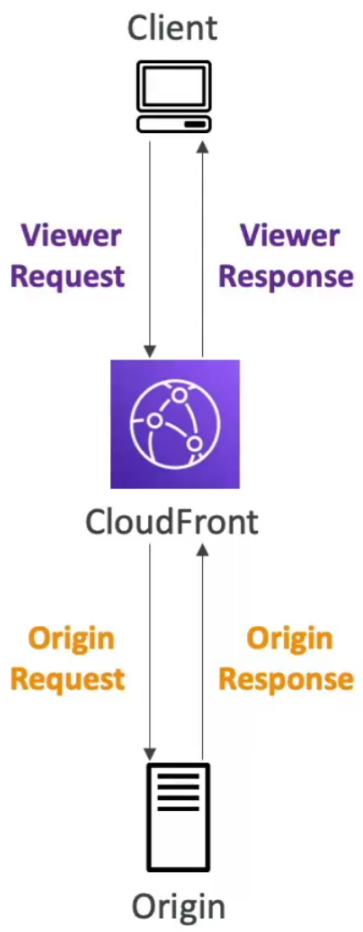
- Lightweight functions written in JavaScript
- For high-scale, latency-sensitive CDN customizations
- Sub-ms startup times, millions of requests/second
- Used to change Viewer requests and responses:
  - **Viewer Request**: after CloudFront receives a request from a viewer
  - **Viewer Response**:before CloudFront forwards the response to the viewer
- Native feature of CloudFront (manage code entirely within CloudFront)



Lambda@Edge

- Lambda functions written in NodeJS or Python
- Scales to 1000s of requests/second
- Used to change CloudFront requests and responses:
  - **Viewer Request** - after CloudFront receives a request from a viewer
  - **Origin Request** - before CloudFront forwards the request to the origin
  - **Origin Response** - after CloudFront receives the response from the origin
  - **Viewer Response** - before CoudFront forwards the response to the viewer

- Author your functions in one AWS Region (us-east-1), then CloudFront replicates to its locations



CloudFront Functions vs. Lambda@Edge - Use Cases

CloudFront Functions

- Cache key normalization
  - Transform request attributes (headers,cookies,query strings, URL) to create an optimal Cache Key
- Header manipulation
  - Insert/modify/delete HTTP headers in the request or response
- URL rewrites or redirects
- Request authentication & authorization
  - Create and validate user-generated tokens (e.g., JWT) to allow/deny requests

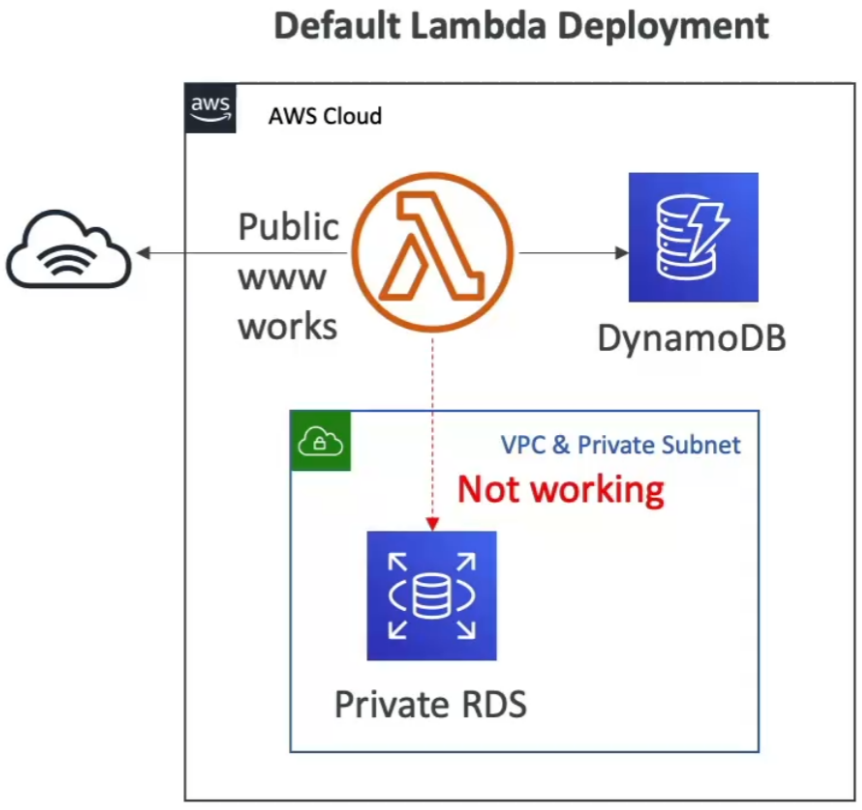
Lambda@Edge

- Longer execution time (several ms)
- Adjustable CPU or memory
- Your code depends on a 3rd libraries (e.g.,AWS SDK to access other AWS services)
- Network access to use external services for processing
- File system access or access to the body of HTTP requests

Lambda by default

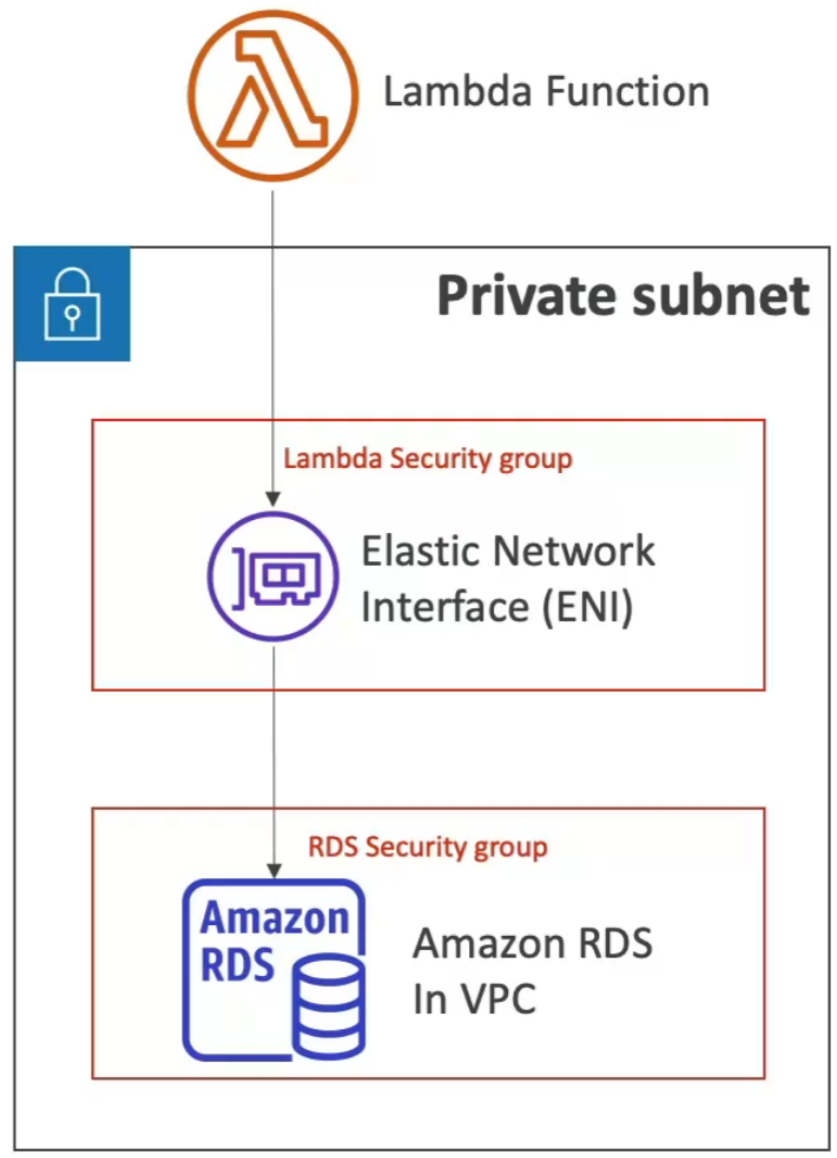
- By default, your Lambda function is launched outside your own VPC (in an AWS-owned VPC)

- Therefore, it cannot access resources in your VPC (RDS,ElastiCache,internal ELB...)



Lambda in VPC

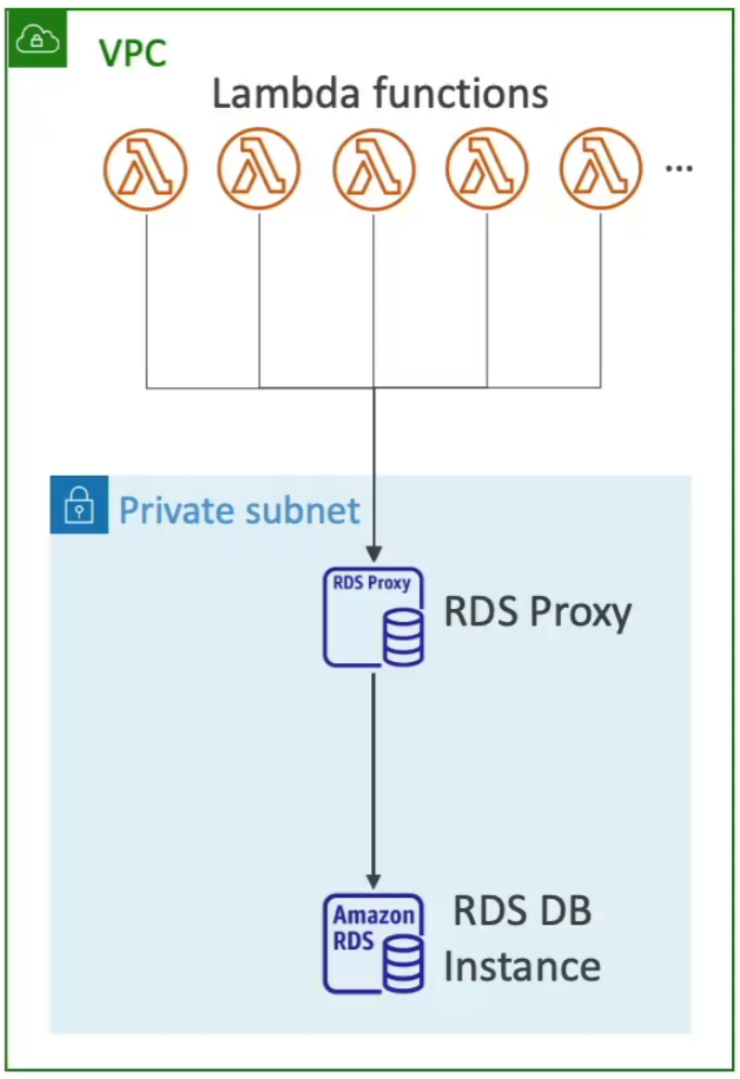
- You must define the VPC ID, the Subnets and the Security Groups
- Lambda will create an ENI (Elastic Network Interface) in your subnets



Lambda with RDS Proxy

- If Lambda functions directly access your database, they may open too many connections under high load
- RDS Proxy
  - Improve scalability by pooling and sharing DB connections
  - Improve availability by reducing by 66% the failover time and preserving connections
  - Improve security by enforcing IAM authentication and storing credentials in Secrets Manager

- **The Lambda function must be deployed in your VPC, because RDS Proxy is never publicly accessible**

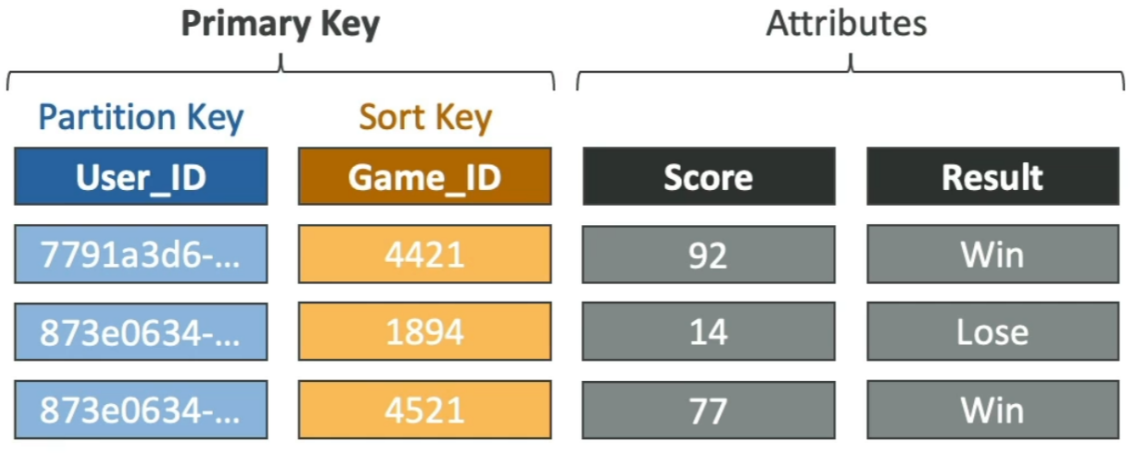


Amazon DynamoDB

- Fully managed,highly available with replication across multiple AZs
- NoSQL database - not a relational database - with tansaction support
- Scales to massive workloads, distributed database
- Millions of requests per seconds, trillions of row, 100s of TB of storage
- Fast and consistent in performance (single-digit millisecond)
- Integrated with IAM for security, authorization and administration
- Low cost and auto-scaling capabilities
- No maintenance or patching, always available
- Standard & Infrequent Access (IA) Table Class

DynamoDB - Basics

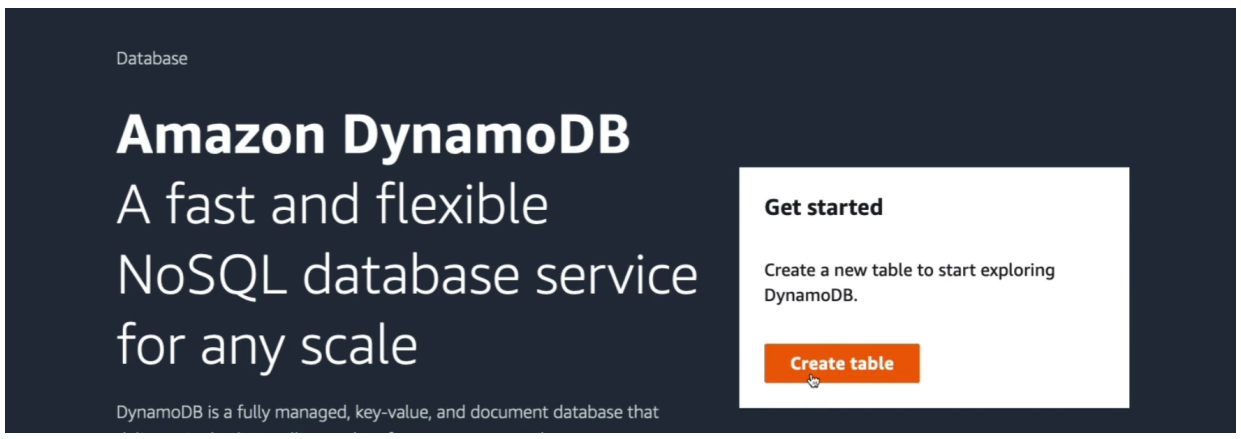
- DynamoDB is made of **Tables**
- Each table has a **Primary Key** (must be decided at creation time)
- Each table can have an infinite number of items (= rows)
- Each item has **attributes** (can be added over time - can be null)
- Maximum size of an item is **400KB**
- Data types supported are:
  - **Scalar Types** - String, Number,Binary,Boolean,Null
  - **DocumentTypes** - List, Map
  - **Set Types** - String Set, Number Set, Binary Set
- **Therefore, in DynamoDB you can rapidly evolve schemas**

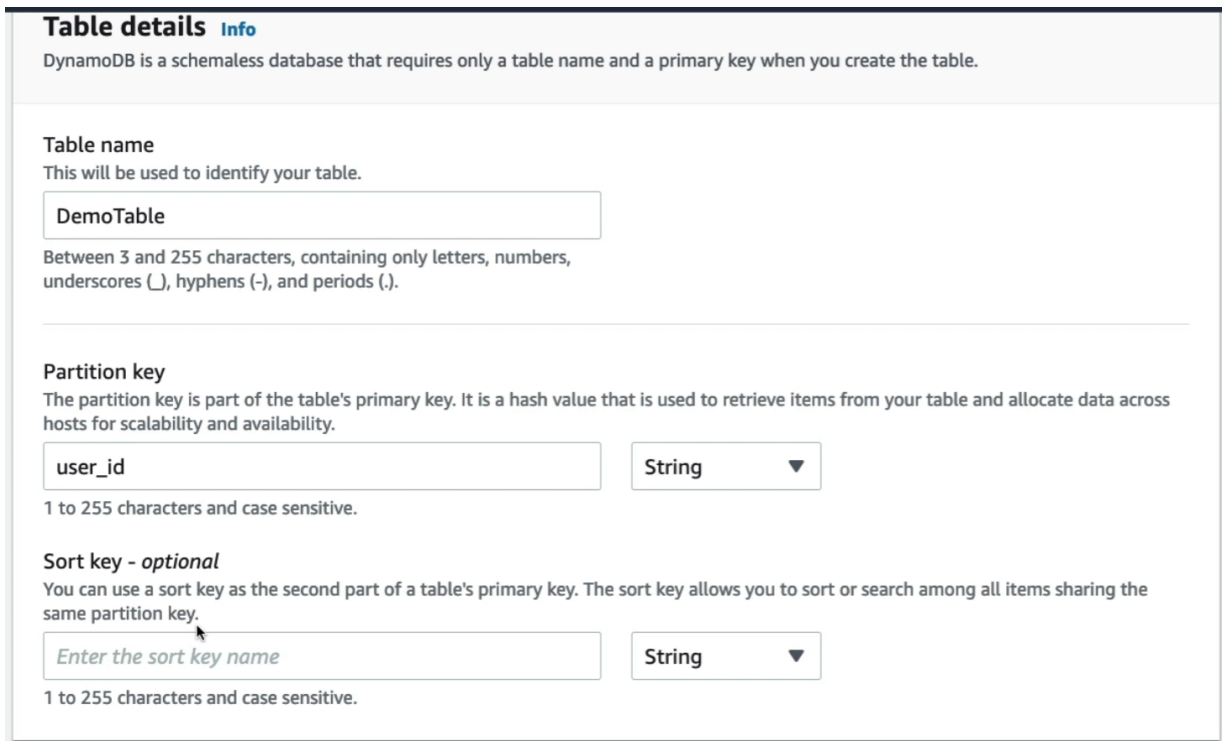


DynamoDB - Read/Write Capacity Modes

- Control how you manage your table's capacity (read/write throughput)
- **Provisioned Mode (default)**
  - You specify the number of reads/writes per second
  - You need to plan capacity beforehand
  - Pay for provisioned Read Capacity Units (RCU) & Write Capacity Units (WCU)
  - Possibility to add auto-scaling mode for RCU & WCU
- On-Demand Mode
  - Read/write automatically scale up/down with your workloads
  - No capacity planning needed
  - Pay for what you use, more expensive($$$)
  - Great for unpredictable workloads, steep sudden spikes

Hands on





On-demand
Simplify billing by paying for the actual reads and writes your application performs.

Provisioned
Manage and optimize your costs by allocating read/write capacity in advance.

Read capacity

Auto scaling **Info**
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

On
Off

Provisioned capacity units

1

Write capacity

Auto scaling **Info**
Dynamically adjusts provisioned throughput capacity on your behalf in response to actual traffic patterns.

On
Off

Provisioned capacity units

1

are not charged any fee for using it.

Encryption key management

Owned by Amazon DynamoDB Learn more
The key is owned and managed by DynamoDB. You are not charged an additional fee for using this customer master key (CMK).

AWS managed CMK Learn more
The key is stored in your account and is managed by AWS Key Management Service (AWS KMS). AWS KMS charges apply.

Stored in your account, and owned and managed by you Learn more
The key is stored in your account and is owned and managed by you. AWS KMS charges apply.

Tags

Tags are pairs of keys and optional values, that you can assign to AWS resources. You can use tags to control access to your resources or track your AWS spending.

No tags are associated with the resource.

Add new tag

You can add 50 more tags.

Cancel Create table

The DemoTable table was created successfully.

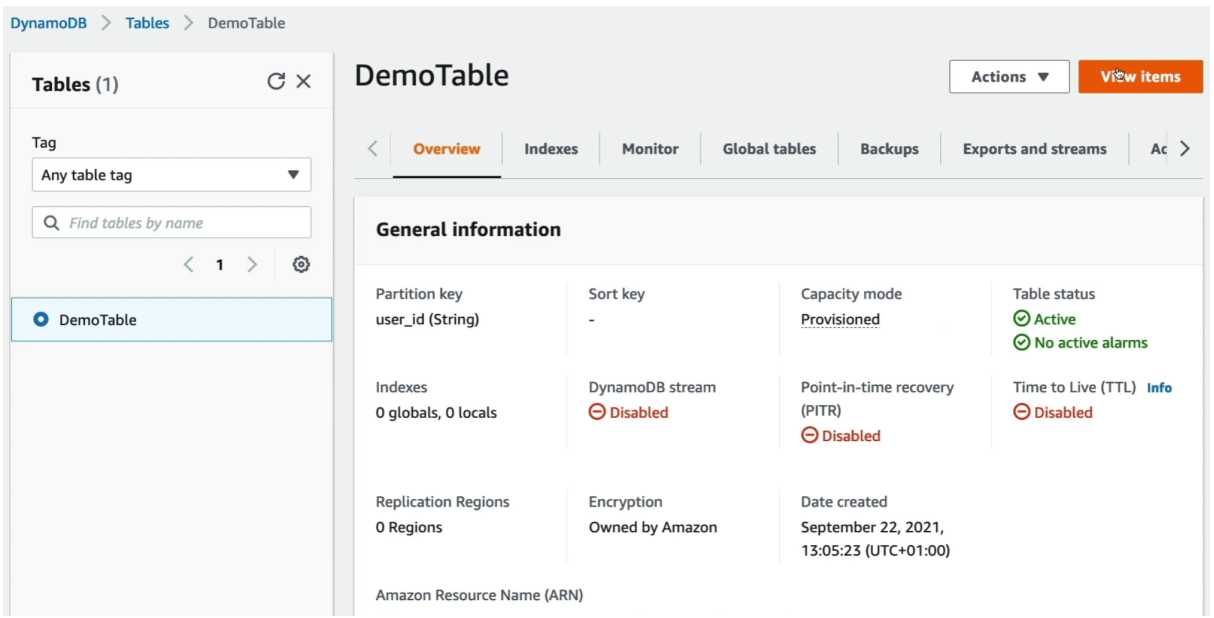
DynamoDB > Tables

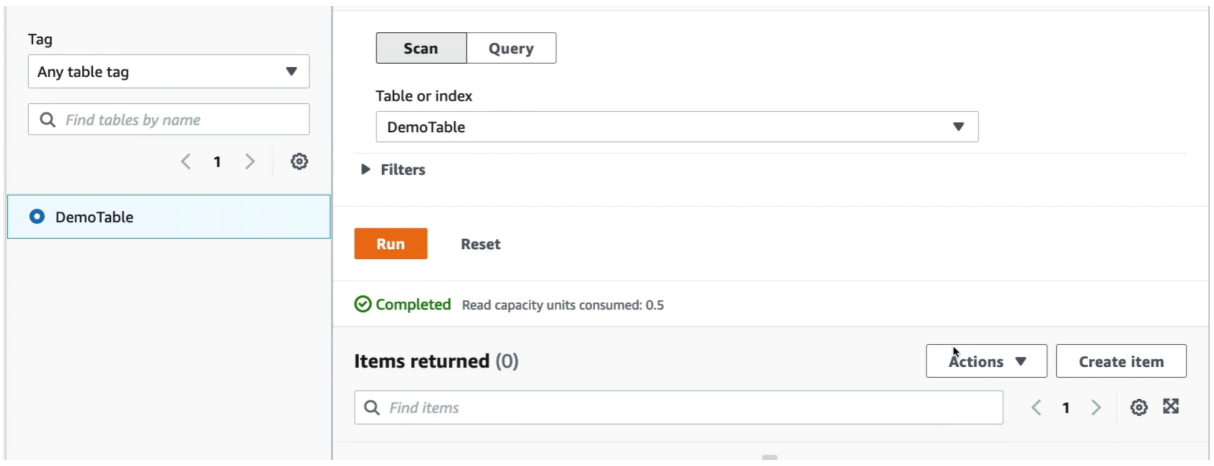
Tables (1) Info

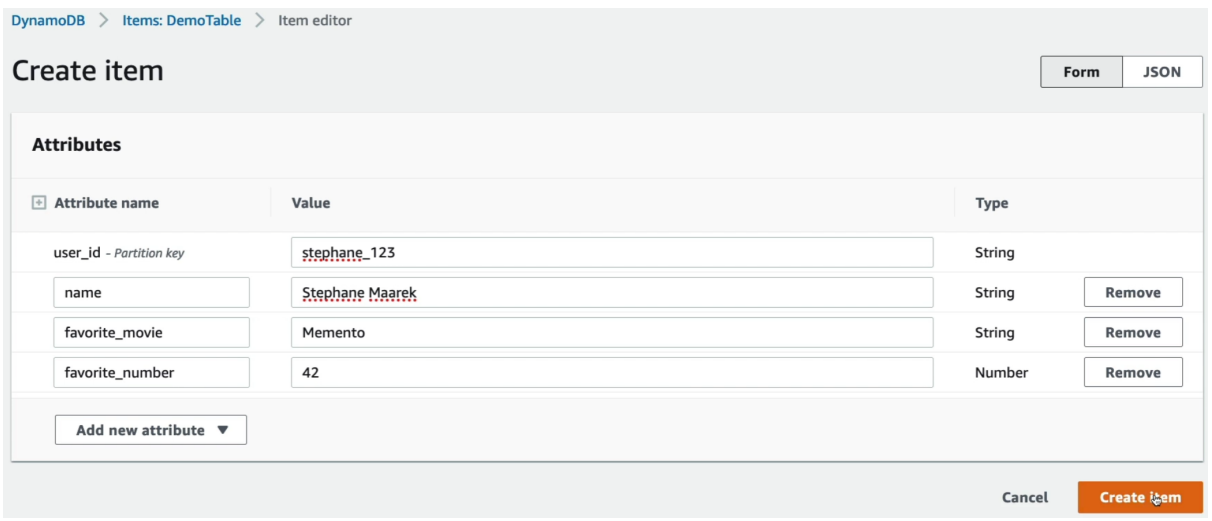
Actions Delete Create table

Find tables by table nam Any table tag 1

Name	Status	Partition key	Sort key	Indexes	Read capacity mode	Write capacit
DemoTable	Active	user_id (String)	-	0	Provisioned (1)	Provisioned (1



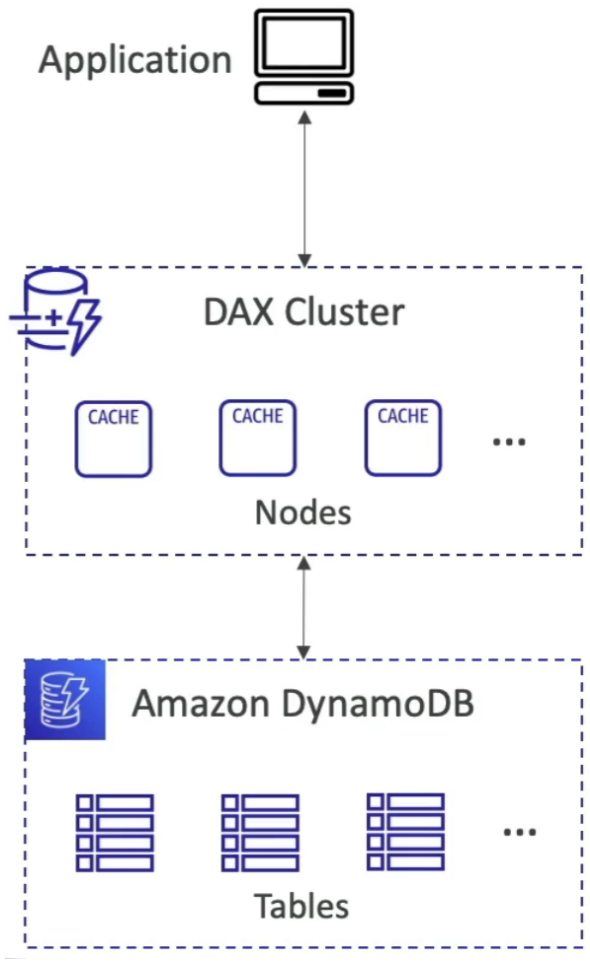




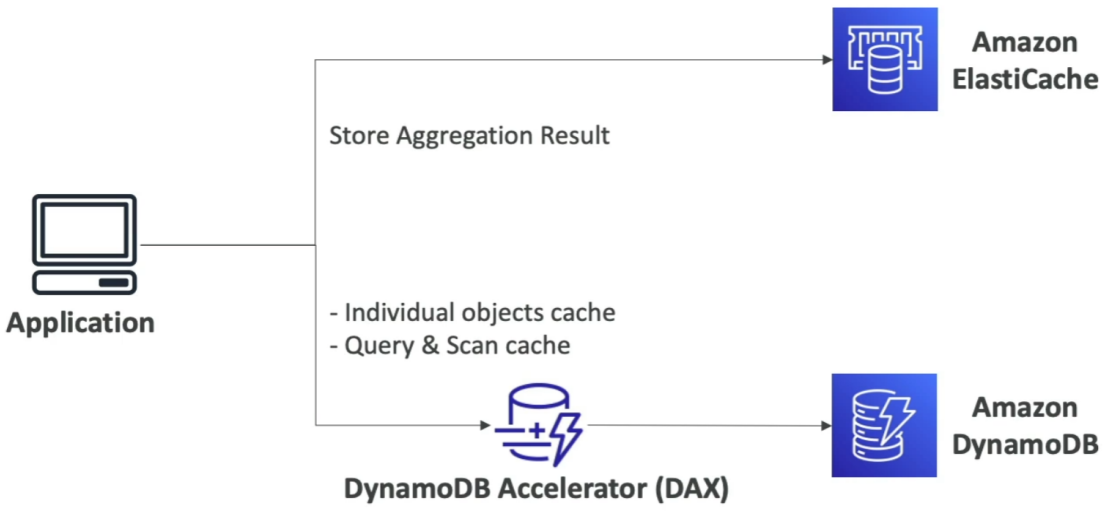
DynamoDB Accelerator (DAX)

- Fully-managed, highly available, seamless in-memory cache for DynamoDB
- **Help solve read congestion by caching**
- **Microseconds latency for cached data**
- Doesn't require application logic modification (compatible with existing DynamoDB APIs)

- 5 minutes TTL for cache (default)



DynamoDB Accelerator (DAX) vs. ElastiCache

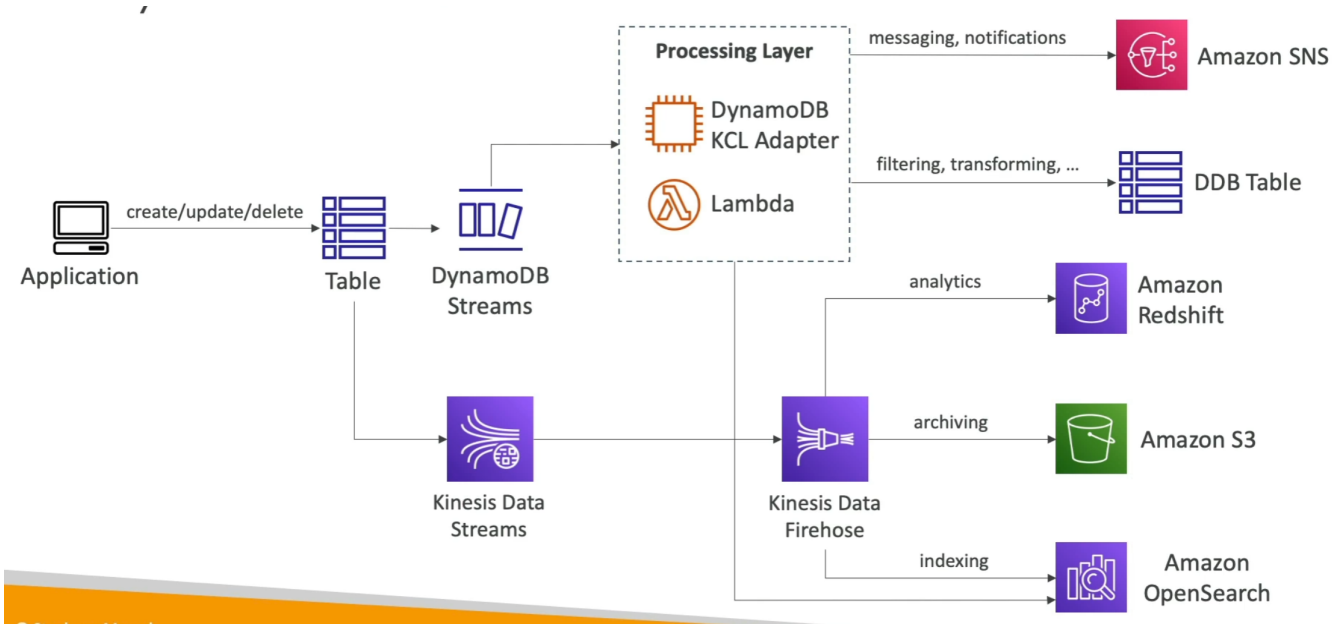


DynamoDB - Stream Processing

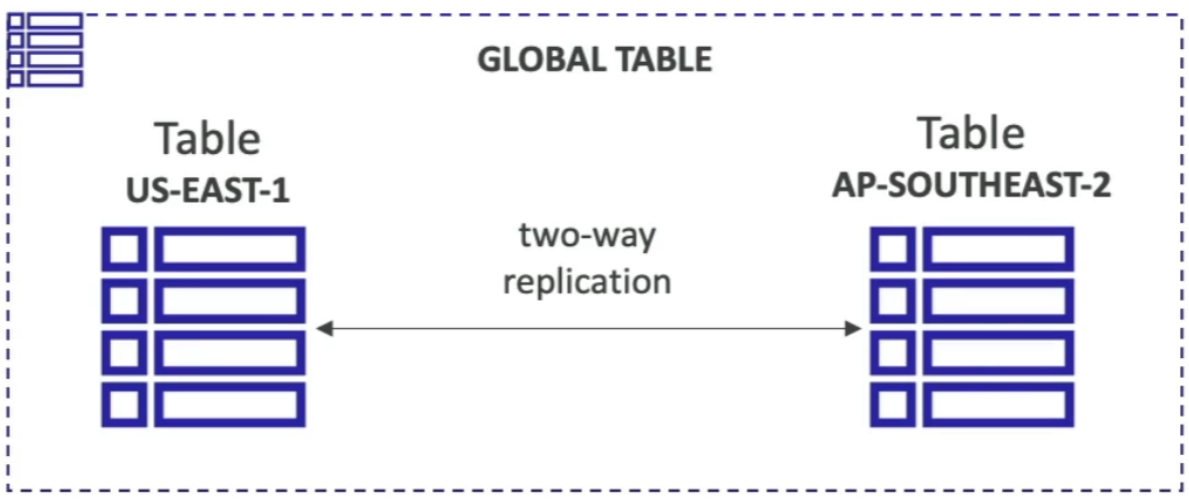
- Ordered stream of item-level modifications (create/update/delete) in a table
- Use cases:
  - React to changes in real-time (welcome email to users)
  - Real-time usage analytics
  - Insert into derivative tables
  - Implement cross-region replication
  - Invoke AWS Lambda on changes to your DynamoDB table

DynamoDB Streams	Kinesis Data Streams (newer)
24 hours retention	1 year retention
Limited # of consumers	High # of consumers
Process using AWS Lambda Triggers, or DynamoDB Stream Kinesis adapter	Process using AWS Lambda, Kinesis Data Analytics, Kineis Data Firehose, AWS Glue Streaming ETL ...

DynamoDB Streams



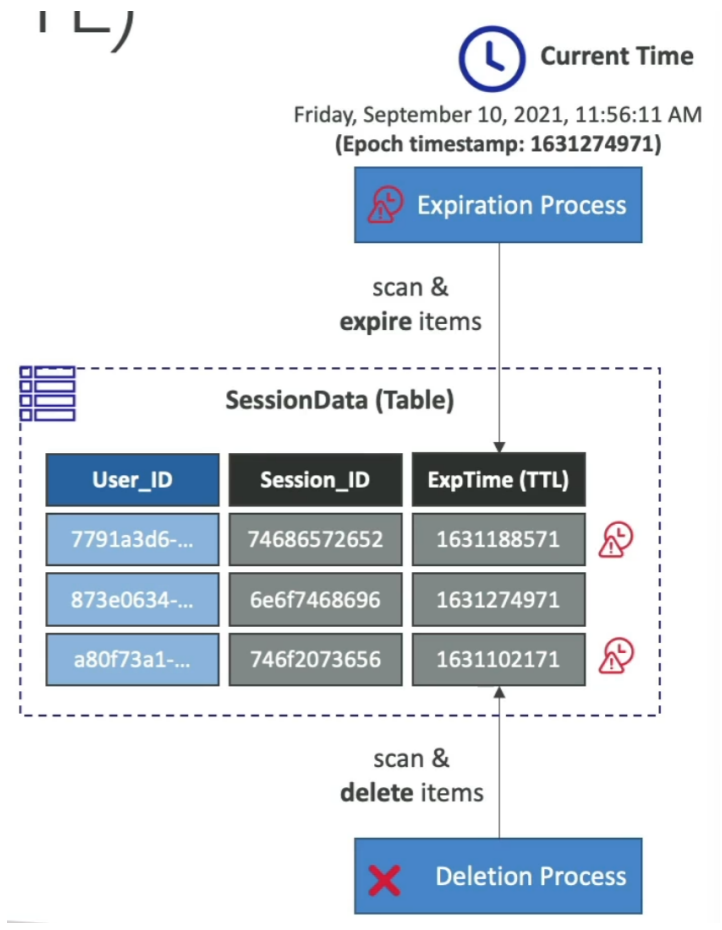
DynamoDB Global Tables



- Make a DynamoDB table accessible with **low latency** in multiple-regions
- Active-Active replication
- Applications can READ and WRITE to the table in any region
- Must enable DynamoDB Streams as a pre-requisite

DynamoDB - Time To Live (TTL)

- Automatically delete items after an expiry timestamp
- Use cases: reduce stored data by keeping only current items, adhere to regulatory obligations, web session handing ...



DynamoDB - Backups for disaster recovery

- Continuous backups using point-in-time recovery (PITR)
  - Optionally enabled for the last 35 days
  - Point-in-time recovery to any time within the backup window
  - The recovery process creates a new table
- On - demand backups
  - Full backups for long-term retention, util explicitely deleted
  - Doesn't affect performance or latency
  - Can be configured and managed in AWS Backup (enables cross-region copy)
  - The recovery process creates a new table

DynamoDB - Integration with Amazon S3

- Export to S3 (must enable PITR)
  - Works for any point of time in the last 35 days
  - Doesn't affect the read capacity of your table
  - Perform data analysis on top of DynamoDB
  - Retain snapshots for auditing
  - ETL on top of S3 data before importing back into DynamoDB
  - Export in DynamoDB JSON or ION format

- Import from S3
  - Import CSV, DynamoDB JSON or ION format
  - Doesn't consume any write capacity
  - Creates a new table
  - Import errors are logged in CloudWatch Logs