AWS re:Invent

API301

Securing data in serverless applications and messaging services

Otavio Ferreira

Sr. Software Development Manager

AWS Serverless / AWS Messaging

AWS re:Invent

aws

Agenda

Designing a serverless architecture

Securing a serverless architecture

- Authentication
- Authorization
- Message encryption
- Message privacy
- Auditing

Designing a serverless architecture

AWS re:Invent

aws

Use case

Electronic medical record (EMR) system

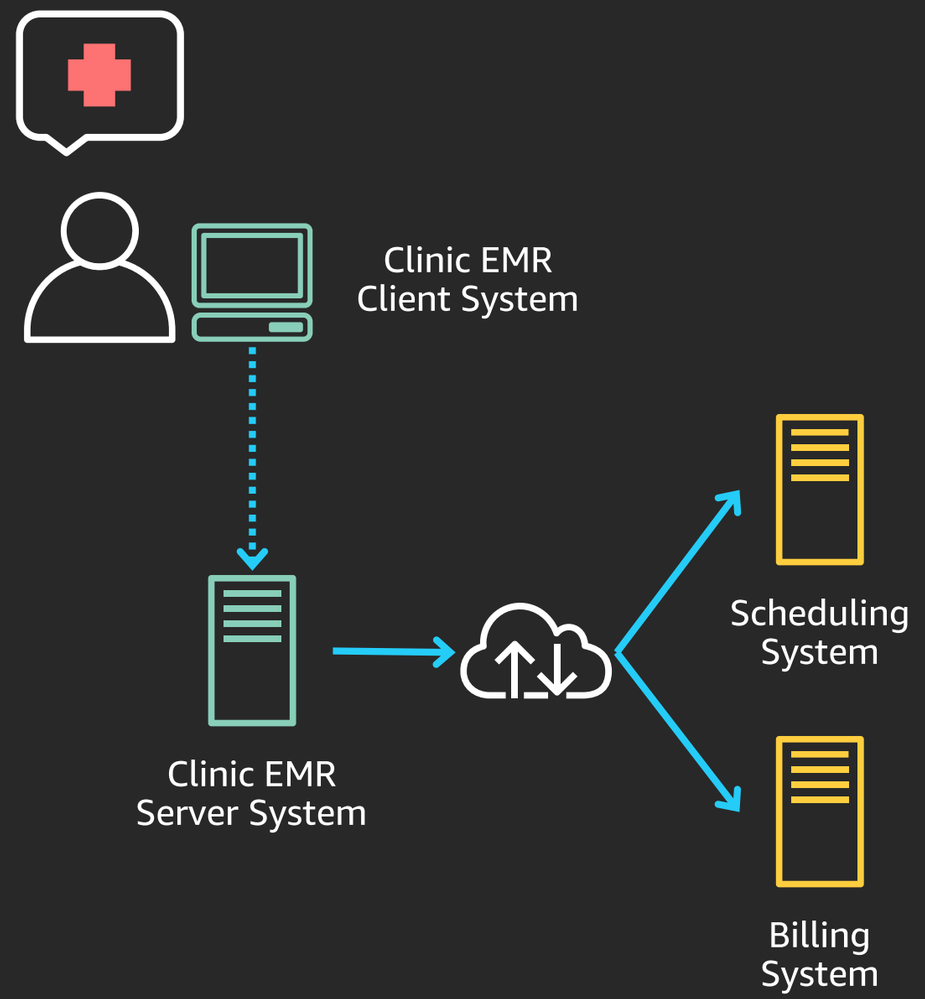
- Personally identifiable information (PII)
- Protected health information (PHI)

Third party legacy back-office systems

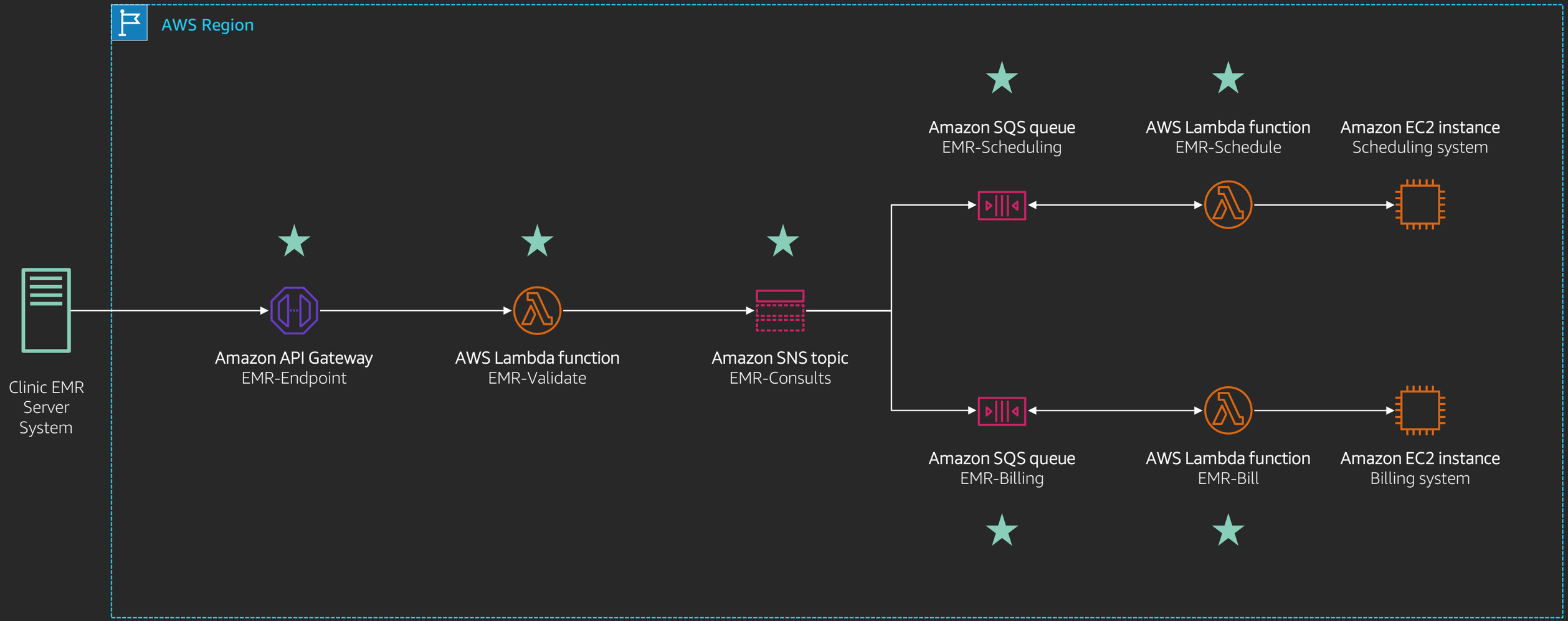
- Scheduling of clinic appointments and lab exams
- Billing of clinic appointments and prescriptions

Strict requirements

- Identity management
- Auditable access control
- Data encryption, in transit and at rest
- Data privacy, no exposure to public internet
- Hands-off capacity management



Designing a serverless architecture



Quick recap

Serverless building blocks

- Amazon API Gateway for handling the HTTP API requests
- Amazon SNS for fanning out the medical consults to multiple back-office systems
- Amazon SQS for queueing the consults and feeding the integration with back-office systems
- AWS Lambda for validating and loading the medical consults into back-office systems

Fully managed platform

- No provisioning of servers
- No patching of operating systems
- No capacity management for peaks
- No software to install or operate
- No upfront costs

Our progress so far…

	Security scope	Addressed yet?
Authentication	Defines identity for each user	-
Authorization	Defines resource access permissions for each user	-
Message encryption	Prevents data from being read in transit and at rest	-
Message privacy	Prevents data from traversing the public internet	-
Auditing	Keeps track of resource access over time	-

Securing a serverless architecture
Authentication

AWS re:Invent

aws

Securing the architecture | Authentication

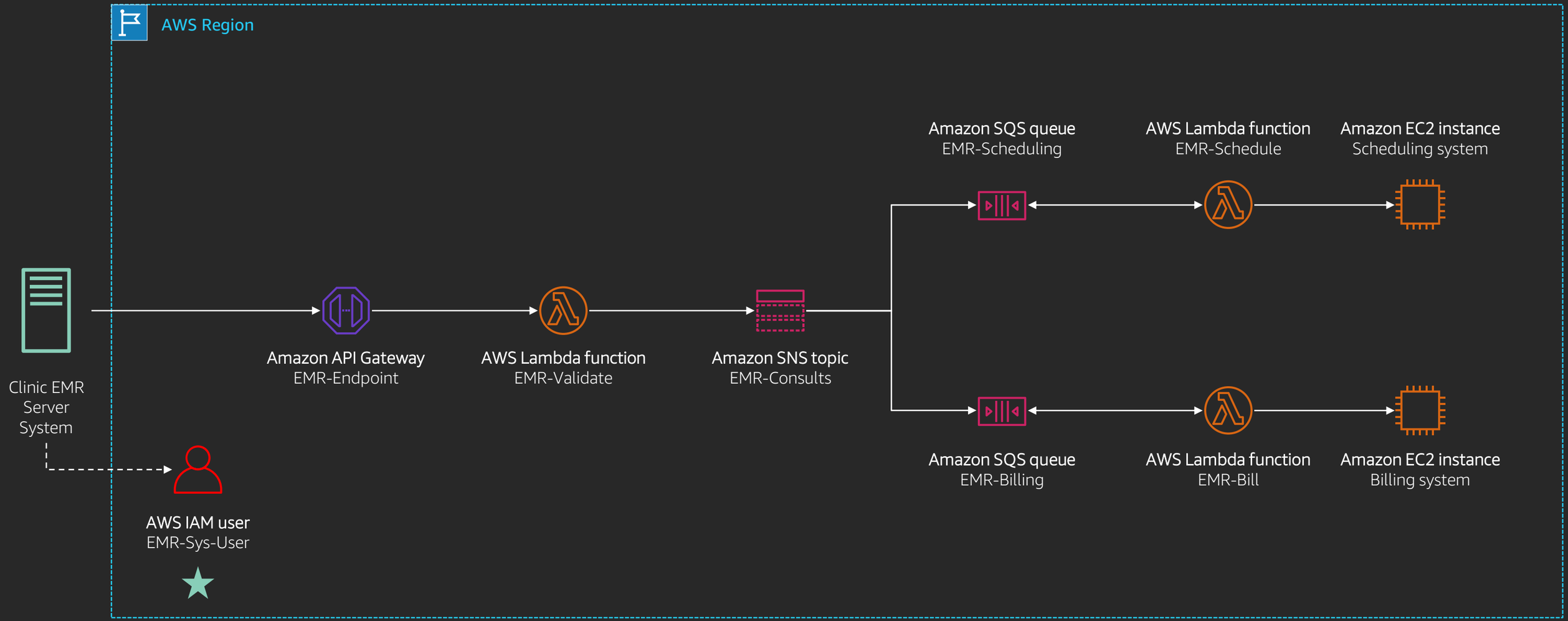
User

- Consists of a name, an AWS Management Console password, and AWS API access keys
- Identifies a person or an application

Role

- Similar to IAM user, but without credentials (e.g. password, access keys)
- Assumed by an IAM user to temporarily take on different permissions

Securing the architecture | Authentication

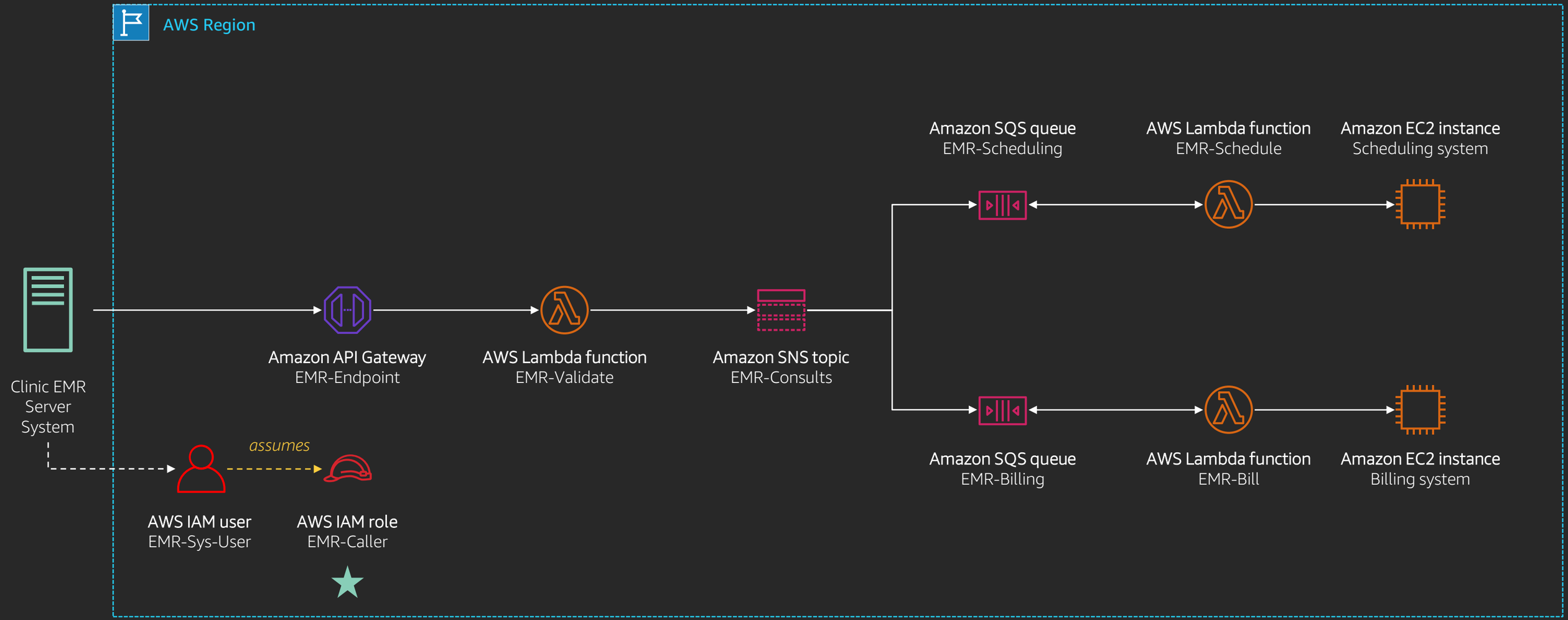


Securing the architecture | Authentication

```
$ aws iam create-user --user-name EMR-Sys-User

$ aws configure --profile EMR-Sys-User
  AWS Access Key ID [****************ABC1]:
  AWS Secret Access Key [****************abcdf:
  Default region name [us-east-1]:
  Default output format [json]:
```

Securing the architecture | Authentication



Securing the architecture | Authentication

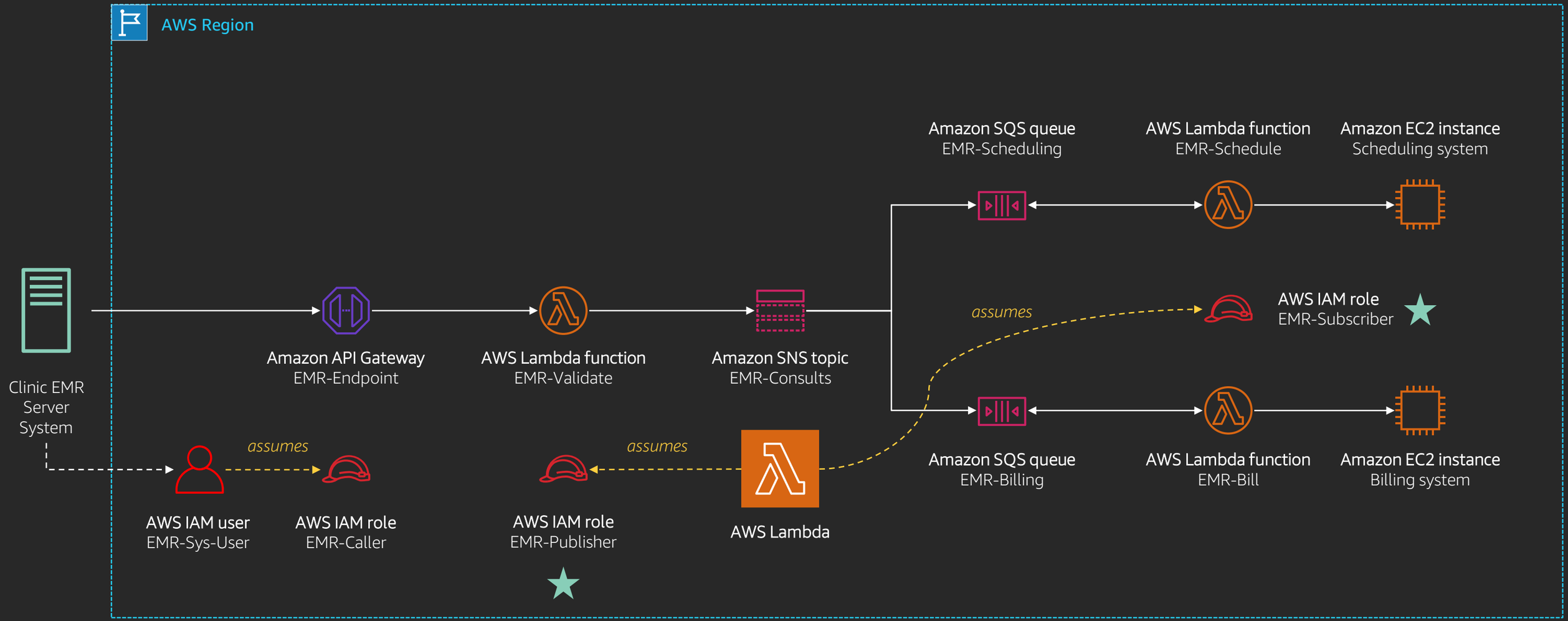
```
$ aws iam create-role
    --role-name EMR-Caller
    --assume-role-policy-document
      '{
         "Version": "2012-10-17",
         "Statement": [
           {
             "Effect": "Allow",
             "Principal": { "AWS": "arn:aws:iam::123456789012:user/EMR-Sys-User" },
             "Action": "sts:AssumeRole"
           }
         ]
      }'
```

Securing the architecture | Authentication

```
$ aws sts assume-role
    --role-arn arn:aws:iam::123456789012:role/EMR-Caller
    --role-session-name my-caller-session
    --profile EMR-Sys-User

$ aws configure --profile EMR-Caller
  AWS Access Key ID [****************ABC1]:
  AWS Secret Access Key [****************abcdf:
  Default region name [us-east-1]:
  Default output format [json]:
```

Securing the architecture | Authentication



Securing the architecture | Authentication

```
$ aws iam create-role
    --role-name EMR-Publisher
    --assume-role-policy-document
      '{
         "Version": "2012-10-17",
         "Statement": [
           {
             "Effect": "Allow",
             "Principal": { "Service": "lambda.amazonaws.com" },
             "Action": "sts:AssumeRole"
           }
         ]
      }'
```

Securing the architecture | Authentication

```
$ aws iam create-role
    --role-name EMR-Subscriber
    --assume-role-policy-document
      '{
         "Version": "2012-10-17",
         "Statement": [
           {
             "Effect": "Allow",
             "Principal": { "Service": "lambda.amazonaws.com" },
             "Action": "sts:AssumeRole"
           }
         ]
      }'
```

Our progress so far…

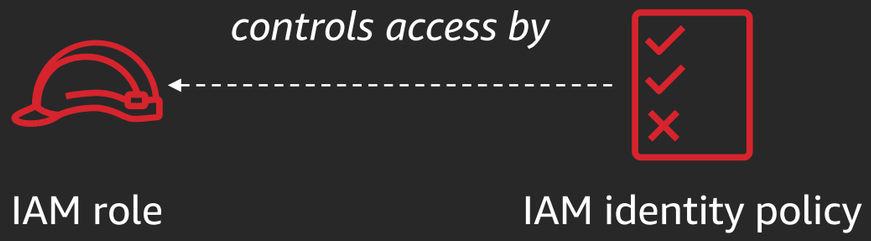
	Security scope	Addressed yet?
Authentication	Defines identity for each user	Yes
Authorization	Defines resource access permissions for each user	-
Message encryption	Prevents data from being read in transit and at rest	-
Message privacy	Prevents data from traversing the public internet	-
Auditing	Keeps track of resource access over time	-

Securing a serverless architecture
Authorization (identity policies)

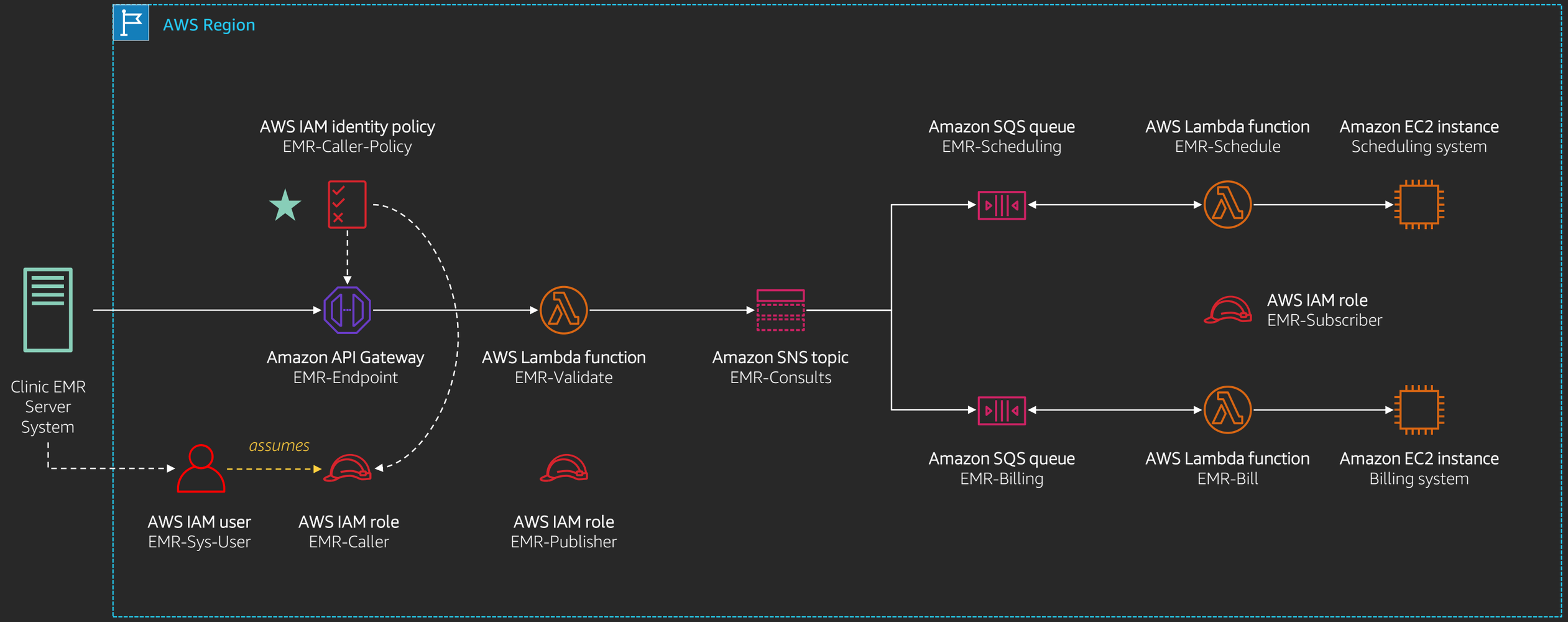
Securing the architecture | Authorization (identity policies)

Identity-based policy

- Attached to an IAM user or IAM role
- Defines their access permissions in terms of resources and actions



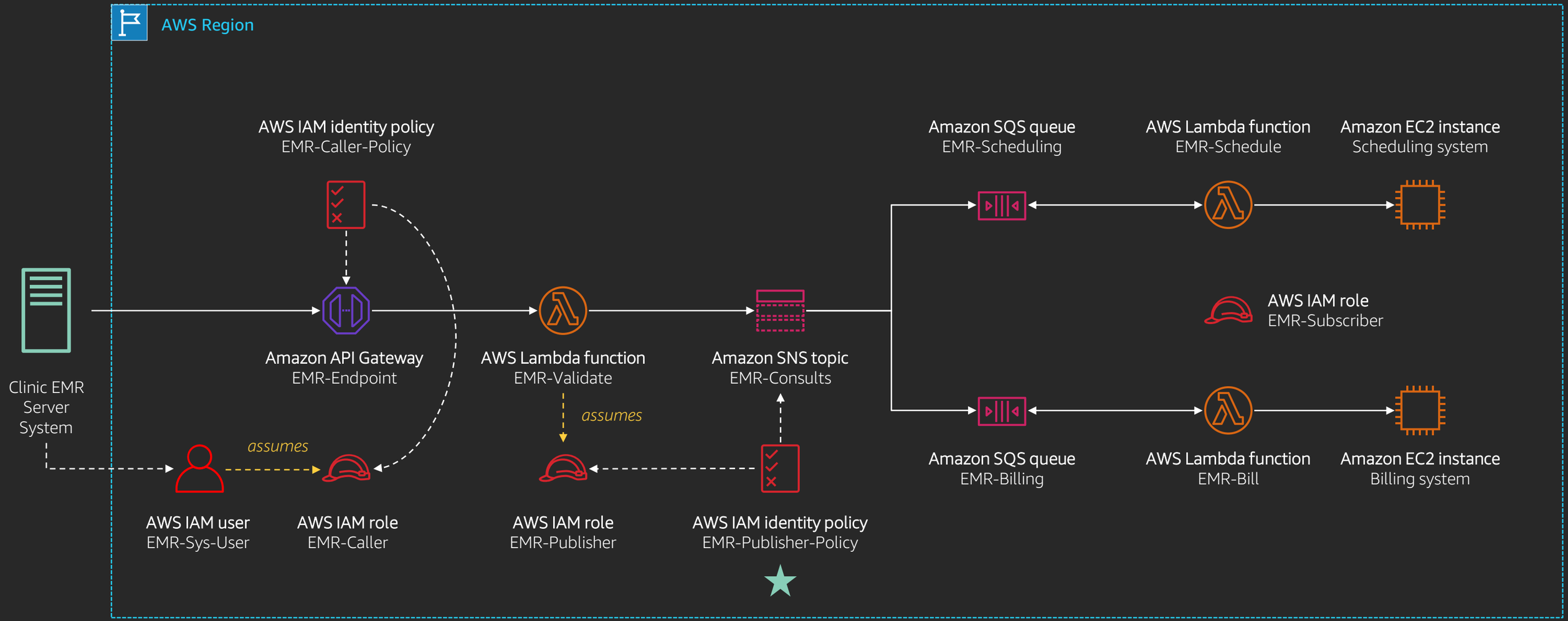
Securing the architecture | Authorization (identity policies)



Securing the architecture | Authorization (identity policies)

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Caller-Policy",
  "Statement": [
    {
      "Sid": "ApiGatewayInvoke",
      "Effect": "Allow",
      "Action": [ "execute-api:Invoke", "execute-api:ManageConnections", "apigateway:POST" ],
      "Resource": [
        "arn:aws:execute-api:us-east-1:123456789012:b1w3v2cyx3/prod/POST/consult",
        "arn:aws:apigateway:us-east-1:123456789012:b1w3v2cyx3/prod/POST/consult"
      ]
    }
  ]
}
```

Securing the architecture | Authorization (identity policies)



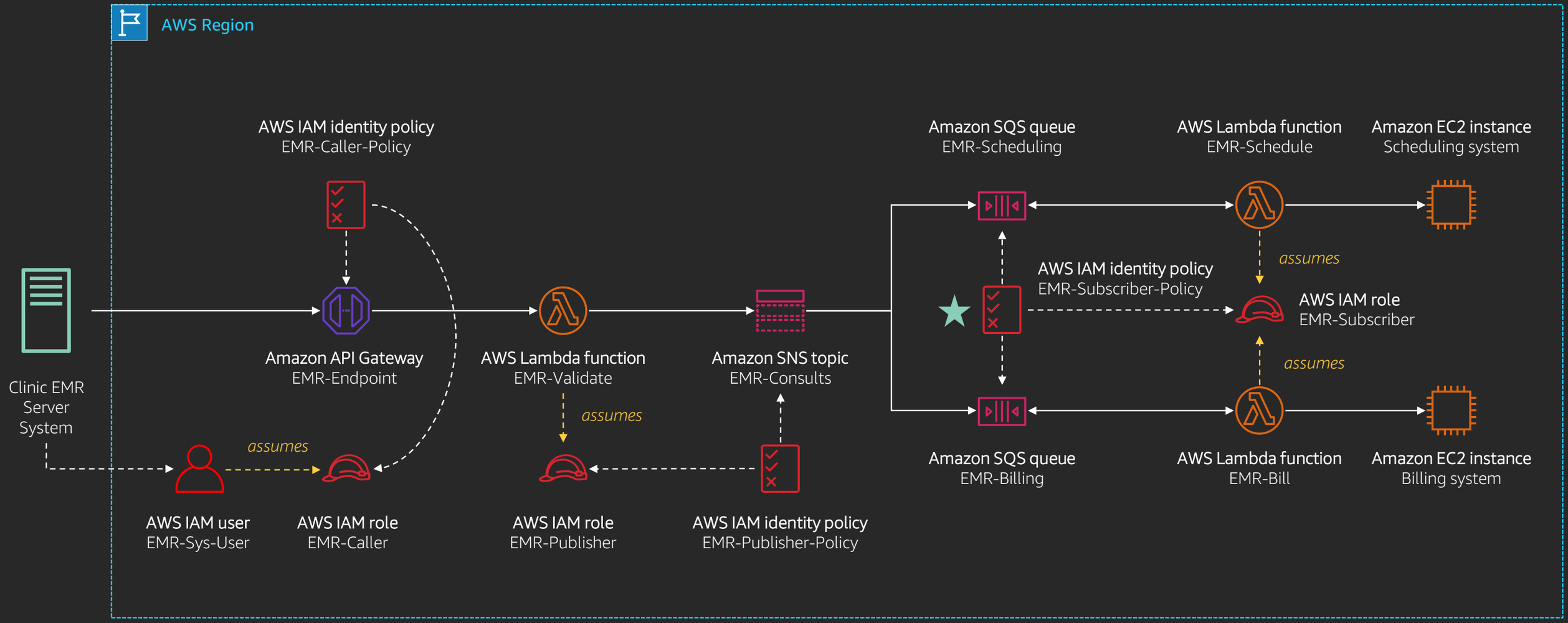
Securing the architecture | Authorization (identity policies)

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Publisher-Policy",
  "Statement": [
    {
      "Sid": "SnsPublish",
      "Effect": "Allow",
      "Action": "sns:Publish",
      "Resource": "arn:aws:sns:us-east-1:123456789012:EMR-Consults"
    },
    { "Sid": "CloudWatchCreateLogGroup", … },
    { "Sid": "CloudWatchCreateLogStream", … }
  ]
}
```

Securing the architecture | Authorization (identity policies)

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Publisher-Policy",
  "Statement": [
    { "Sid": "SnsPublish", … },
    {
      "Sid": "CloudWatchCreateLogGroup",
      "Effect": "Allow",
      "Action": "logs:CreateLogGroup",
      "Resource": "arn:aws:logs:us-east-1:123456789012:*"
    }, {
      "Sid": "CloudWatchCreateLogStream",
      "Effect": "Allow",
      "Action": [ "logs:CreateLogStream", "logs:PutLogEvents" ],
      "Resource": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/lambda/EMR-Publisher:*"
    }
  ]
}
```

Securing the architecture | Authorization (identity policies)



Securing the architecture | Authorization (identity policies)

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Subscriber-Policy",
  "Statement": [
    {
      "Sid": "SqsReceiveMessage",
      "Effect": "Allow",
      "Action": [ "sqs:ReceiveMessage", "sqs:DeleteMessage", "sqs:GetQueueAttributes" ],
      "Resource": [
        "arn:aws:sqs:us-east-1:123456789012:EMR-Scheduling”,
        "arn:aws:sqs:us-east-1:123456789012:EMR-Billing”
      ]
    },
    { "Sid": "CloudWatchCreateLogGroup", … },
    { "Sid": "CloudWatchCreateLogStream", … }
  ]
}
```

Securing the architecture | Authorization (identity policies)

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Subscriber-Policy",
  "Statement": [
    { "Sid": "SqsReceiveMessage", … },
    {
      "Sid": "CloudWatchCreateLogGroup",
      "Effect": "Allow",
      "Action": "logs:CreateLogGroup",
      "Resource": "arn:aws:logs:us-east-1:123456789012:*"
    }, {
      "Sid": "CloudWatchCreateLogStream",
      "Effect": "Allow",
      "Action": [ "logs:CreateLogStream", "logs:PutLogEvents" ],
      "Resource": "arn:aws:logs:us-east-1:123456789012:log-group:/aws/lambda/EMR-Subscriber:*"
    }
  ]
}
```

Our progress so far…

	Security scope	Addressed yet?
Authentication	Defines identity for each user	Yes
Authorization	Defines resource access permissions for each user	Partially
Message encryption	Prevents data from being read in transit and at rest	-
Message privacy	Prevents data from traversing the public internet	-
Auditing	Keeps track of resource access over time	-

Securing a serverless architecture
Authorization (resource policies)

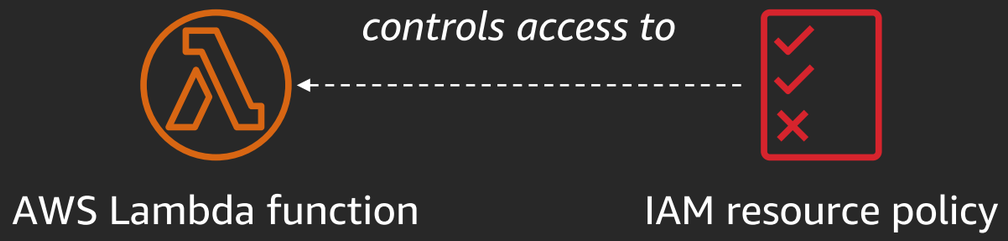
AWS re:Invent

aws

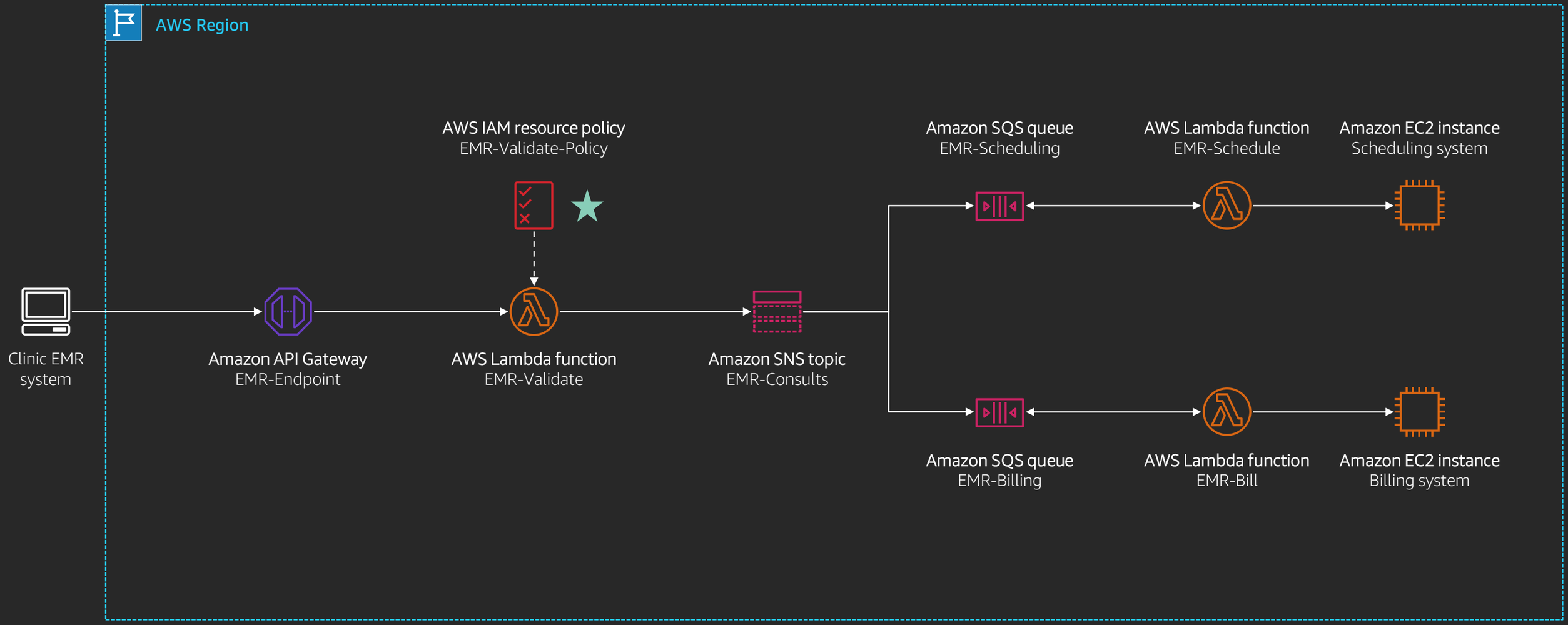
Securing the architecture | Authorization (resource policies)

Resource-based policy

- Attached to an AWS resource (e.g. SNS topic, SQS queue, Lambda function)
- Defines who has access to the resource, and what actions they can perform on it
- Required when you have no access to the resource caller (e.g. AWS service principal, another AWS account)



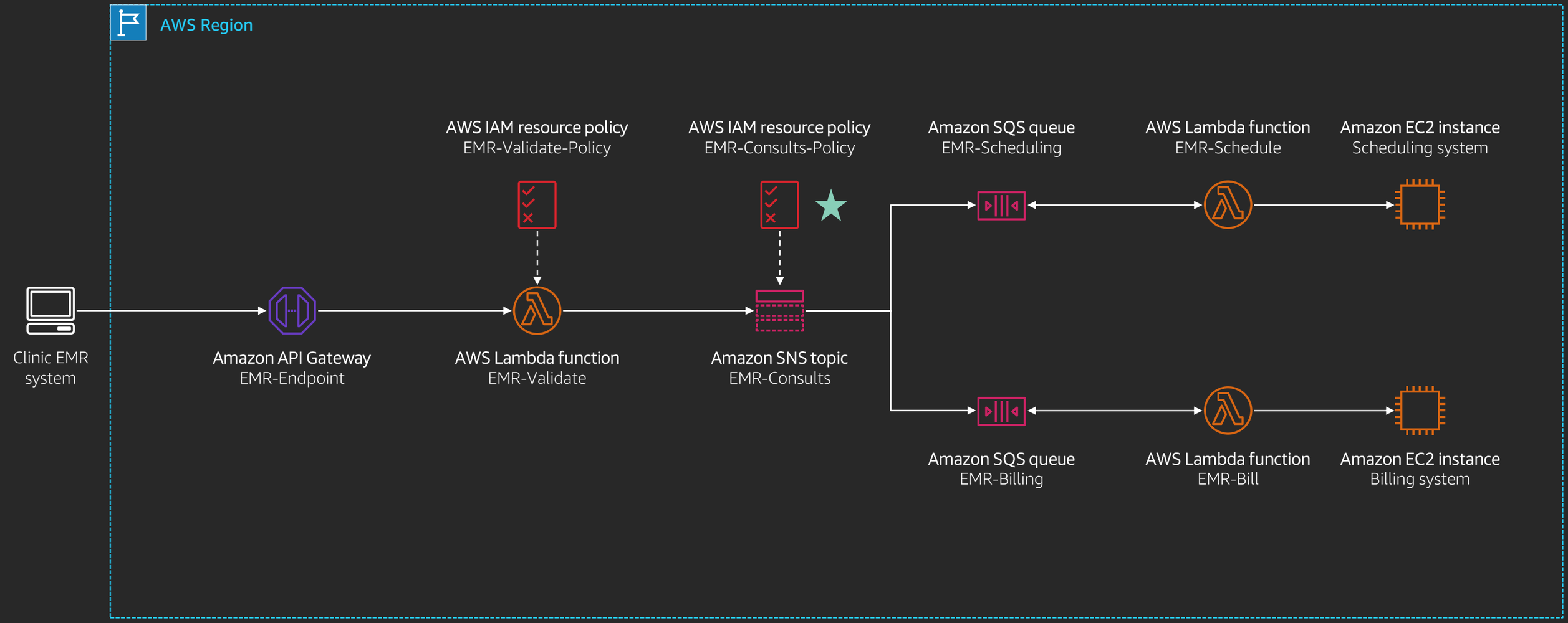
Securing the architecture | Authorization (resource policies)



Securing the architecture | Authorization (resource policies)

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Validate-Policy",
  "Statement": [
    {
      "Sid": "LambdaInvoke",
      "Effect": "Allow",
      "Principal": {
        "Service": "apigateway.amazonaws.com"
      },
      "Action": "lambda:invokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:123456789012:function:EMR-Validate",
      "Condition": {
        "ArnEquals": {
        "AWS:SourceArn": "arn:aws:execute-api:us-east-1:123456789012:b1w3v2cyx3/prod/POST/consult"
        }
      }
    }
  ]
}
```

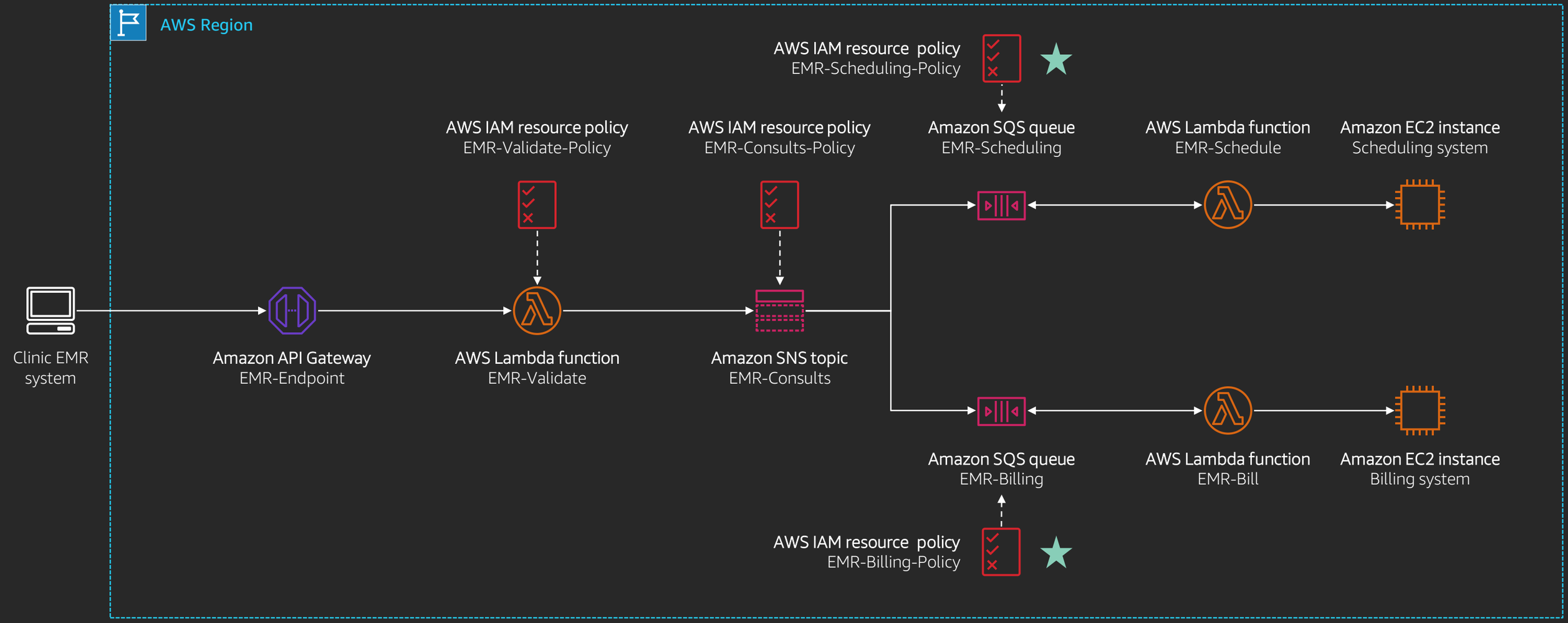
Securing the architecture | Authorization (resource policies)



Securing the architecture | Authorization (resource policies)

```
{
  "Version": "2008-10-17",
  "Id": "EMR-Consults-Policy",
  "Statement": [
    {
      "Sid": "SnsSubscribe",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "SNS:Subscribe",
      "Resource": "arn:aws:sns:us-east-1:123456789012:EMR-Consults",
      "Condition": {
        "StringEquals": {
          "sns:Protocol": "sqs"
        }
      }
    }
  ]
}
```

Securing the architecture | Authorization (resource policies)



Securing the architecture | Authorization (resource policies)

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Scheduling-Policy",
  "Statement": [
    {
      "Sid": "SqsSendMessage",
      "Effect": "Allow",
      "Principal": {
        "Service": "sns.amazonaws.com"
      },
      "Action": "SQS:SendMessage",
      "Resource": "arn:aws:sqs:eu-north-1:123456789012:EMR-Scheduling",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:sns:eu-north-1:123456789012:EMR-Consults"
        }
      }
    }
  ]
}
```

Securing the architecture | Authorization (resource policies)

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Billing-Policy",
  "Statement": [
    {
      "Sid": "SqsSendMessage",
      "Effect": "Allow",
      "Principal": {
        "Service": "sns.amazonaws.com"
      },
      "Action": "SQS:SendMessage",
      "Resource": "arn:aws:sqs:eu-north-1:123456789012:EMR-Billing",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": "arn:aws:sns:eu-north-1:123456789012:EMR-Consults"
        }
      }
    }
  ]
}
```

Our progress so far…

	Security scope	Addressed yet?
Authentication	Defines identity for each user	Yes
Authorization	Defines resource access permissions for each user	Yes
Message encryption	Prevents data from being read in transit and at rest	-
Message privacy	Prevents data from traversing the public internet	-
Auditing	Keeps track of resource access over time	-

Securing a serverless architecture
Message encryption in transit

AWS re:Invent

aws

Securing the architecture | Message encryption in transit

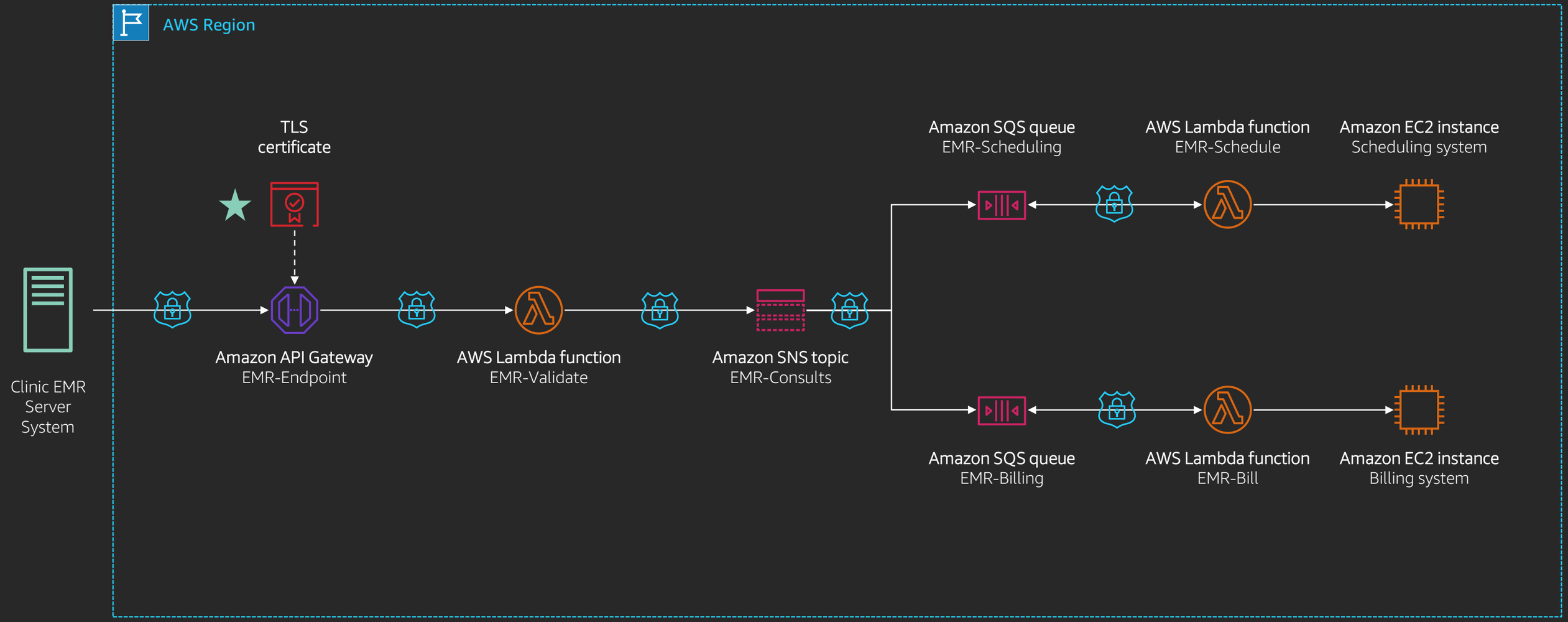
Data encryption in transit

API Gateway endpoints, served from your custom domains, can be secured by TLS certificates managed by AWS Certificate Manager (ACM)

Amazon SNS, Amazon SQS, and Lambda provide HTTPS APIs that encrypt your messages in transit too

HTTPS APIs are secured by TLS 1.2 certificates, issued by Amazon Trust Services (ATS) as Certificate Authority (CA), using a 256-bit SHA algorithm and 2048-bit RSA keys

Securing the architecture | Message encryption in transit



Securing the architecture | Message encryption in transit

```
$ aws apigateway create-domain-name
    --domain-name 'emr.mydomain.com'
    --endpoint-configuration types=REGIONAL
    --regional-certificate-arn
    'arn:aws:acm:us-east-1:123456789012:certificate/c29332f0-3be6-467f-a244-e03a423084e7'
```

Our progress so far…

	Security scope	Addressed yet?
Authentication	Defines identity for each user	Yes
Authorization	Defines resource access permissions for each user	Yes
Message encryption	Prevents data from being read in transit and at rest	Partially
Message privacy	Prevents data from traversing the public internet	-
Auditing	Keeps track of resource access over time	-

Securing a serverless architecture
Message encryption at rest

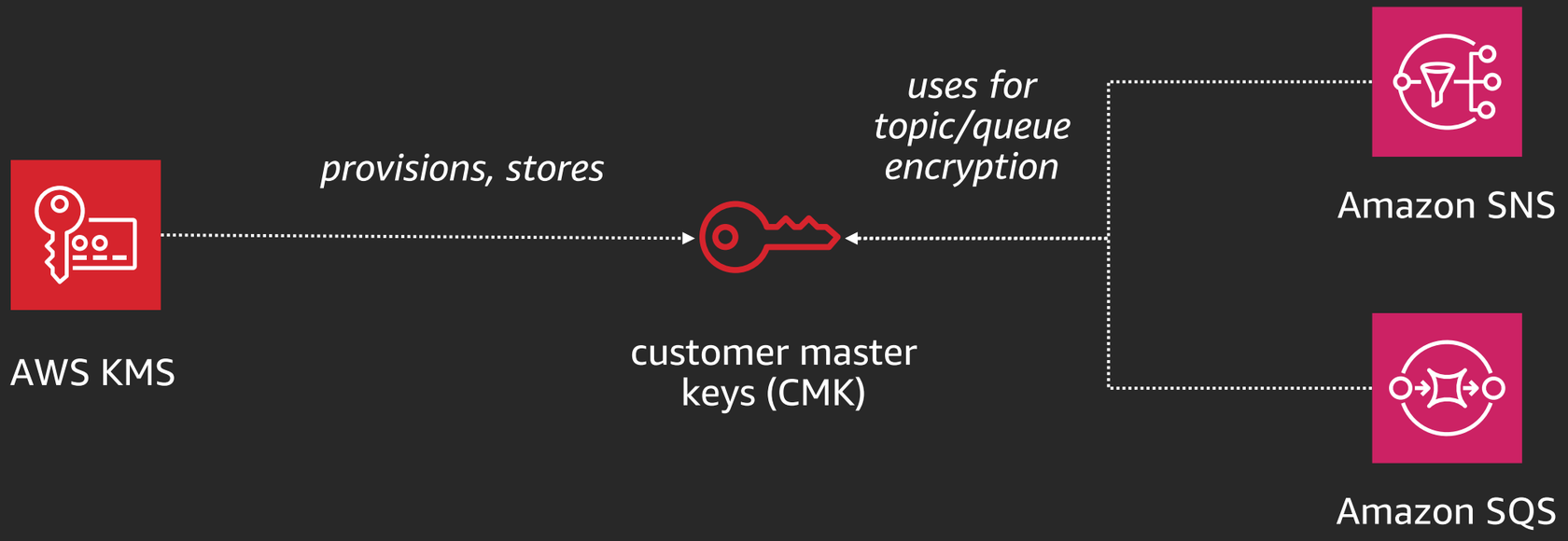
AWS re:Invent

aws

Securing the architecture | Message encryption at rest

Server-side encryption (SSE)

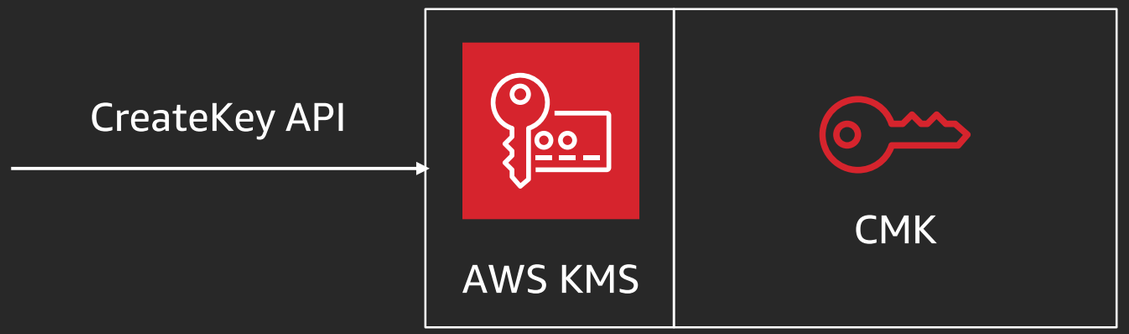
- Amazon SNS encrypted topics and Amazon SQS encrypted queues are powered by AWS Key Management Service (AWS KMS)
- Messages are encrypted using a 256-bit AES-GCM algorithm
- Messages are stored in encrypted form, in multiple Availability Zones (AZ)



Securing the architecture | Message encryption at rest

Customer master key (CMK)

- Created in AWS KMS; never leaves AWS KMS
- Used for generating, encrypting, and decrypting your data encryption keys (DEK)



Securing the architecture | Message encryption at rest

Customer master key (CMK)

Amazon SNS and Amazon SQS support both customer managed and AWS managed CMK

	Can view	Can set a key policy	Used only in your account
Customer managed CMK	Yes	Yes	Yes
AWS managed CMK	Yes	No	Yes
AWS-owned CMK	No	No	No

Securing the architecture | Message encryption at rest

Data encryption key (DEK)

- Managed and used only outside of AWS KMS
- Used for encrypting and decrypting your application data

Securing the architecture | Message encryption at rest

Envelope encryption

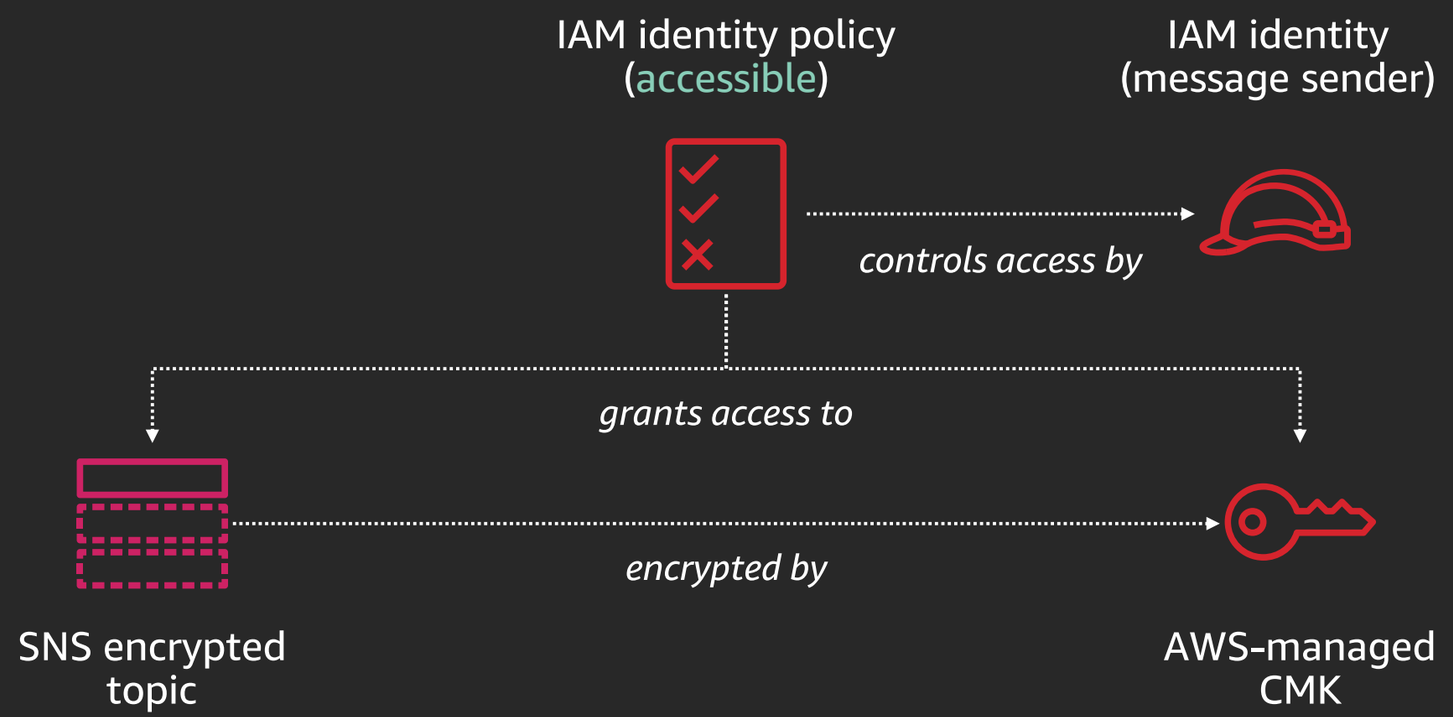
The practice of encrypting plaintext data with a data key, and then encrypting the data key under another key

Amazon SNS and Amazon SQS rotate your DEK every 5 minutes, by default

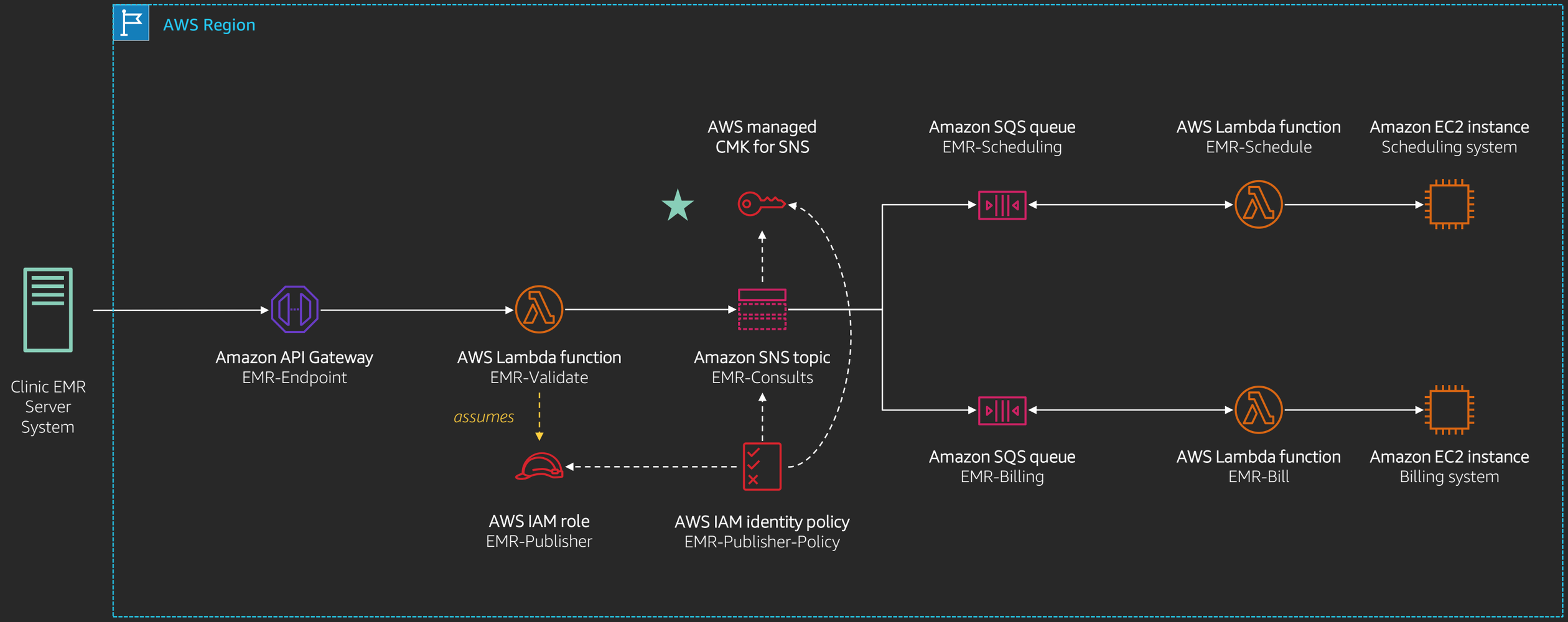
Securing the architecture | Message encryption at rest

AWS managed CMK

- Can be used when you have access to the message sender's identity
- The key policy cannot be changed, but the sender’s identity policy can



Securing the architecture | Message encryption at rest



Securing the architecture | Message encryption at rest

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Publisher-Policy",
  "Statement": [
    {
      "Sid": "KmsGenerateDataKey",
      "Effect": "Allow",
      "Action": [ "kms:GenerateDataKey", "kms:Decrypt" ]
      "Resource": "arn:aws:kms:us-east-1:123456789012:alias/aws/sns"
    },
    {"Sid": "SnsPublish", … },
    {"Sid": "CloudWatchCreateLogGroup", … },
    {"Sid": "CloudWatchCreateLogStream", … }
  ]
}
```

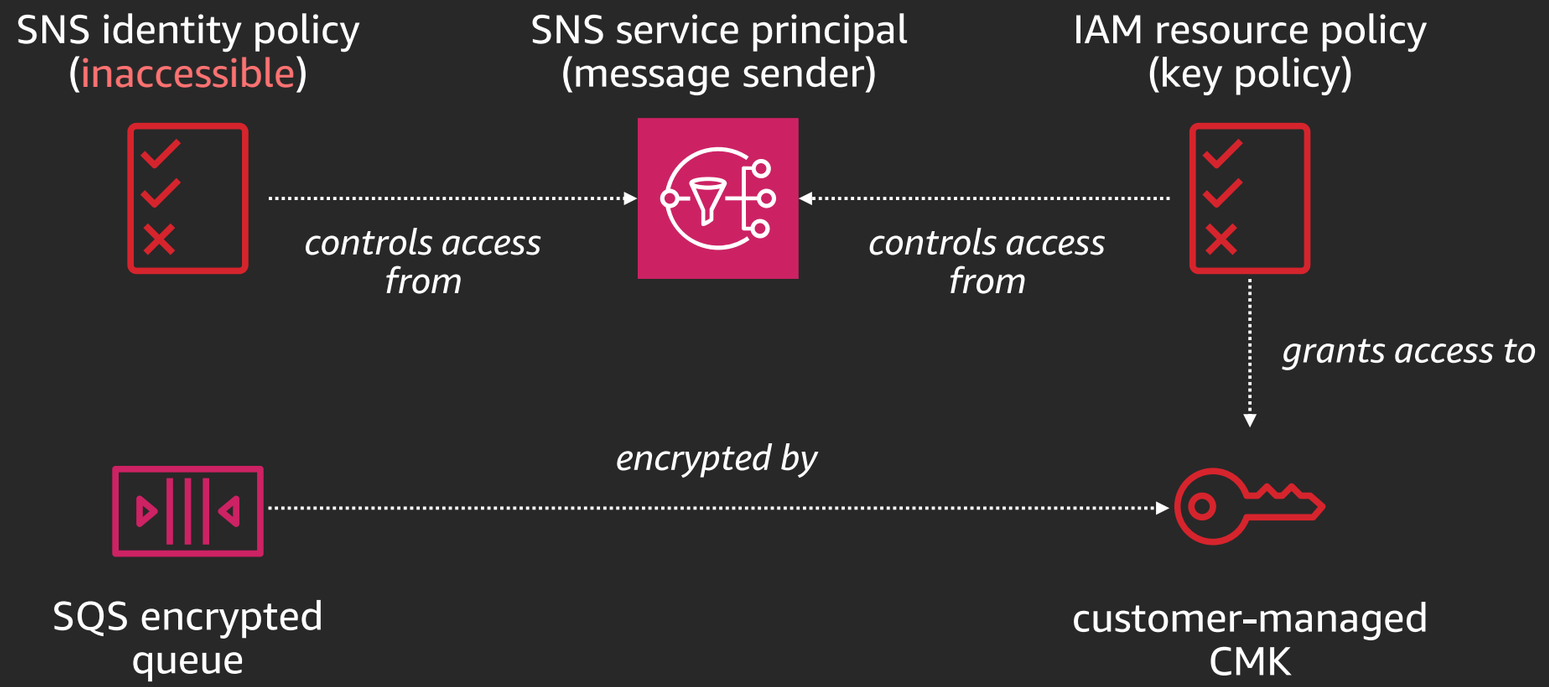
Securing the architecture | Message encryption at rest

```
$ aws sns set-topic-attributes
    --topic-arn arn:aws:sns:us-east-1:123456789012:EMR-Consults
    --attribute-name KmsMasterKeyId
    --attribute-value arn:aws:kms:us-east-1:123456789012:alias/aws/sns
```

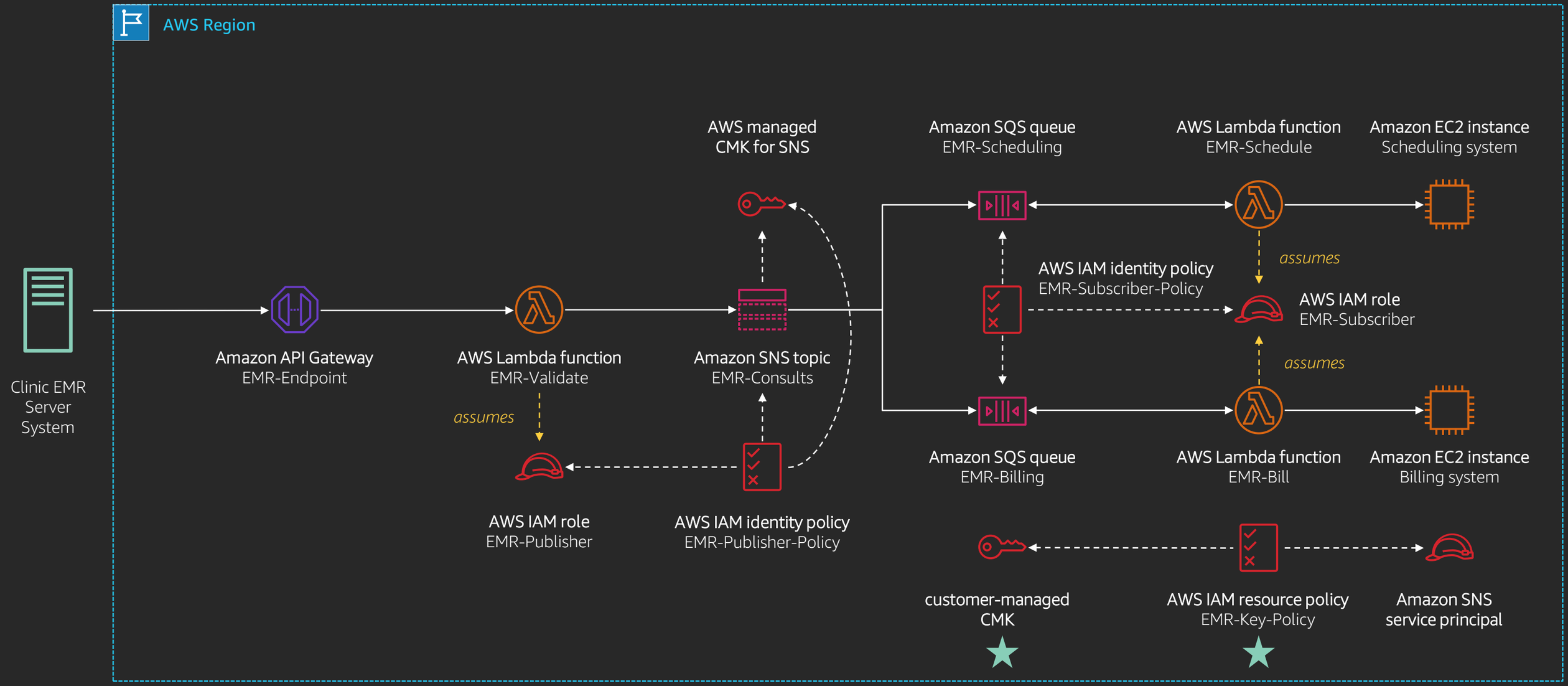
Securing the architecture | Message encryption at rest

Customer managed CMK

- Should be used to address compliance, or when you have no access to the sender's identity
- A custom key policy allows or denies access to the CMK, for either an IAM identity (user, role) or an AWS service principal



Securing the architecture | Message encryption at rest



Securing the architecture | Message encryption at rest

```
$ aws kms create-key
```

Securing the architecture | Message encryption at rest

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Key-Policy",
  "Statement": [
    {
      "Sid": "AllowServicePrincipal",
      "Effect": "Allow",
      "Principal": { "Service": "sns.amazonaws.com" },
      "Action": [ "kms:GenerateDataKey", "kms:Decrypt" ],
      "Resource": "arn:aws:kms:us-east-1:123456789012:key/e40ad9d9-b1e8-45b6-8779-e3fc8dfc634c"
    },
    { "Sid": "AllowRootUser", … },
    { "Sid": "AllowKeyAdministrator", … }
  ]
}
```

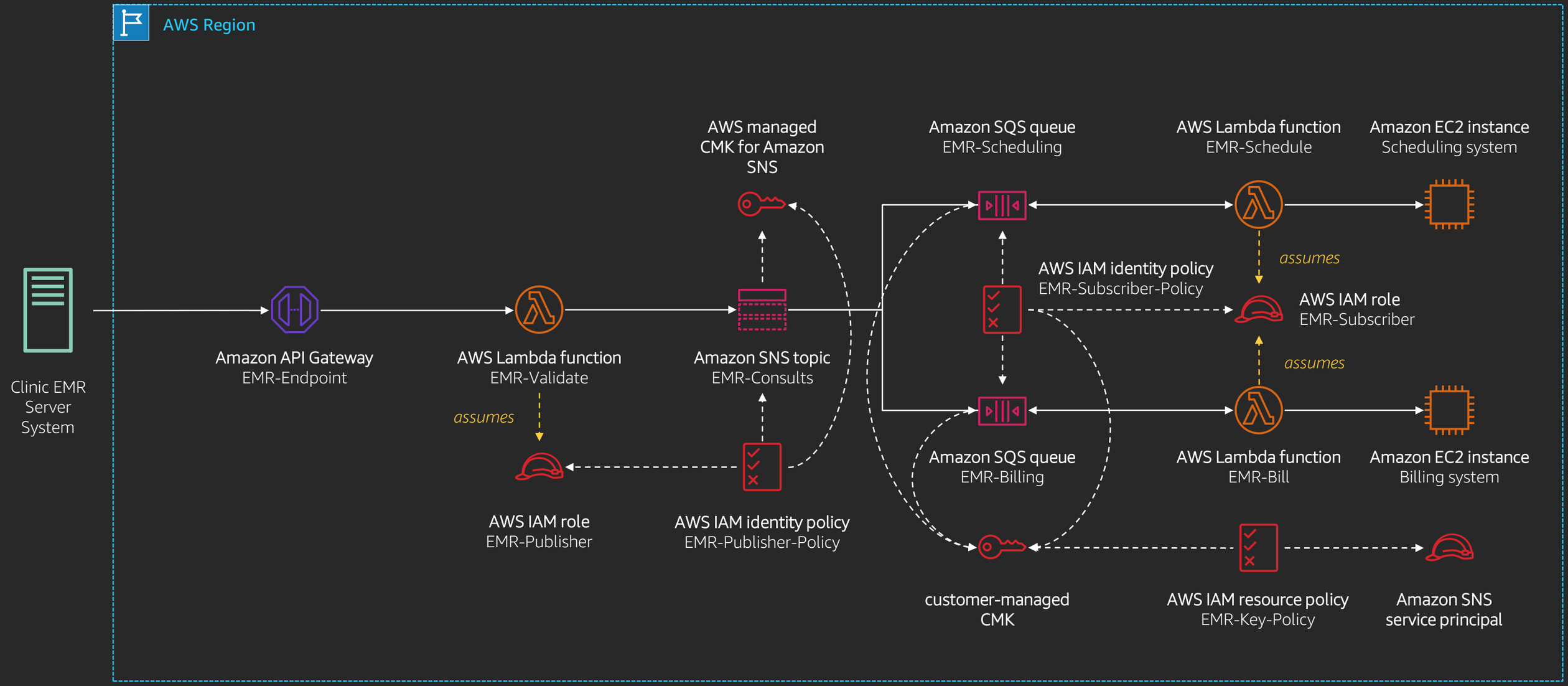
Securing the architecture | Message encryption at rest

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Key-Policy",
  "Statement": [
    { "Sid": "AllowServicePrincipal", … },
    {
      "Sid": "AllowRootUser",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::123456789012:root" },
      "Action": "kms:*",
      "Resource": "*"
    },
    { "Sid": "AllowKeyAdministrator", … }
  ]
}
```

Securing the architecture | Message encryption at rest

```
{
  "Version": "2012-10-17",
  "Id": "EMR-Key-Policy",
  "Statement": [
    { "Sid": "AllowServicePrincipal", … },
    { "Sid": "AllowRootUser", … },
    {
      "Sid": "AllowKeyAdministrator",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::123456789012:user/EMR-Key-Admin" },
      "Action": [
        "kms:Create*", "kms:Describe*", "kms:Enable*", "kms:List*", "kms:Put*", "kms:Update*",
        "kms:Revoke*", "kms:Disable*", "kms:Get*","kms:Delete*", "kms:Schedule*", "kms:Cancel*"
      ],
      "Resource": "arn:aws:kms:us-east-1:123456789012:key/e40ad9d9-b1e8-45b6-8779-e3fc8dfc634c"
    }
  ]
}
```

Securing the architecture | Message encryption at rest



Securing the architecture | Message encryption at rest

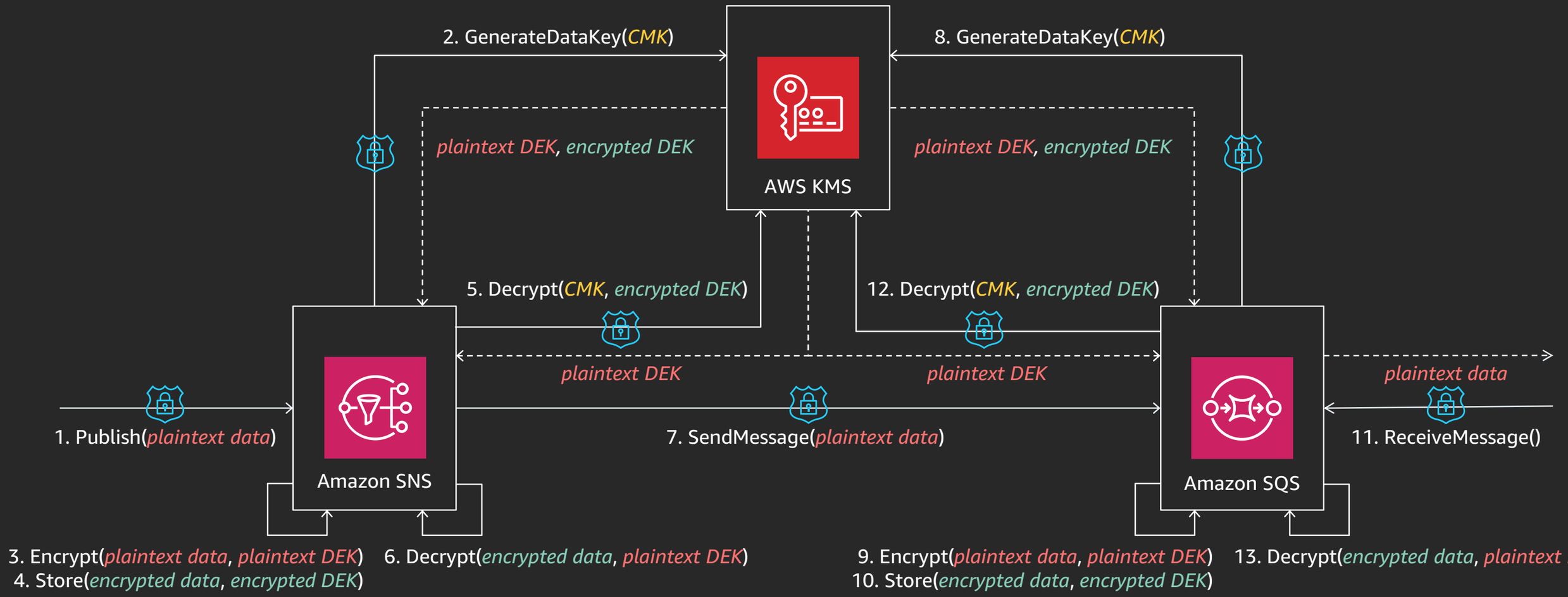
```
{
  "Version": "2012-10-17",
  "Id": "EMR-Subscriber-Policy",
  "Statement": [
    {
      "Sid": "KmsGenerateDataKey",
      "Effect": "Allow",
      "Action": [ "kms:GenerateDataKey", "kms:Decrypt" ],
      "Resource": "arn:aws:kms:us-east-1:123456789012:key/e40ad9d9-b1e8-45b6-8779-e3fc8dfc634c"
    },
    { "Sid": "SqsReceiveMessage", … },
    { "Sid": "CloudWatchCreateLogGroup", … },
    { "Sid": "CloudWatchCreateLogStream", … }
  ]
}
```

Securing the architecture | Message encryption at rest

```
$ aws sqs set-queue-attributes
    --queue-url https://sqs.us-east-1.amazonaws.com/123456789012/EMR-Scheduling
    --attributes
'{"KmsMasterkeyId":"arn:aws:kms:us-east-1:123456789012:key/e40ad9d9-b1e8-45b6-8779-e3fc8dfc634c"}'


$ aws sqs set-queue-attributes
    --queue-url https://sqs.us-east-1.amazonaws.com/123456789012/EMR-Billing
    --attributes
'{"KmsMasterkeyId":"arn:aws:kms:us-east-1:123456789012:key/e40ad9d9-b1e8-45b6-8779-e3fc8dfc634c"}'
```

Securing the architecture | Message encryption at rest



Our progress so far…

	Security scope	Addressed yet?
Authentication	Defines identity for each user	Yes
Authorization	Defines resource access permissions for each user	Yes
Message encryption	Prevents data from being read in transit and at rest	Yes
Message privacy	Prevents data from traversing the public internet	-
Auditing	Keeps track of resource access over time	-

Securing a serverless architecture
Message privacy

AWS re:Invent

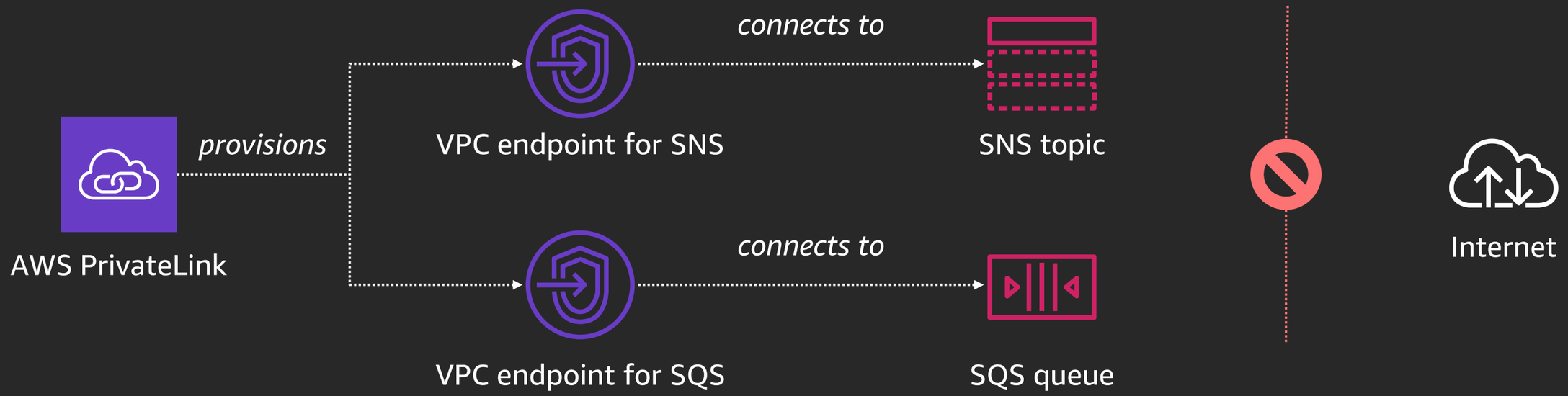
aws

Securing the architecture | Message privacy

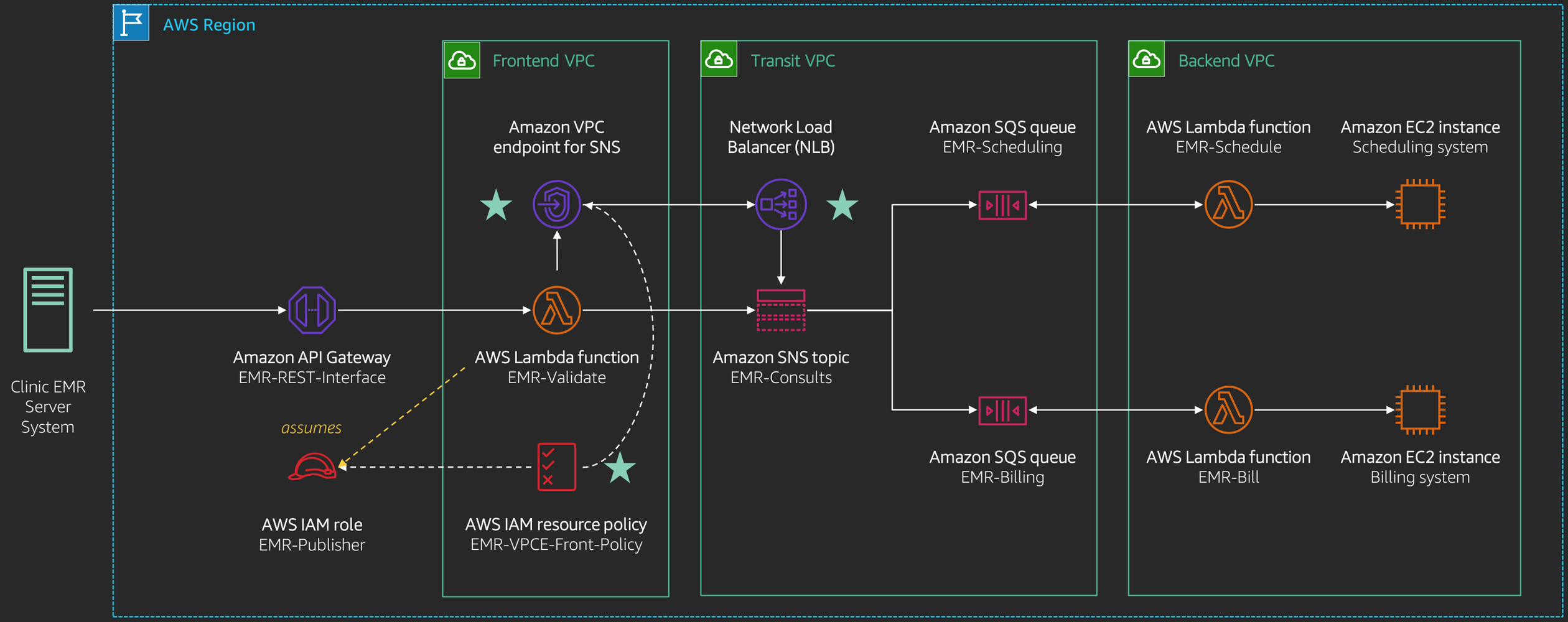
Virtual private cloud (VPC) endpoints

Messages are sent privately from a VPC subnet to AWS resources, including SNS topics and SQS queues, without traversing the public Internet

VPC endpoints powered by AWS PrivateLink prevent you from setting up an internet gateway, network address translation (NAT) device, or virtual private network (VPN) connection



Securing the architecture | Message privacy



Securing the architecture | Message privacy

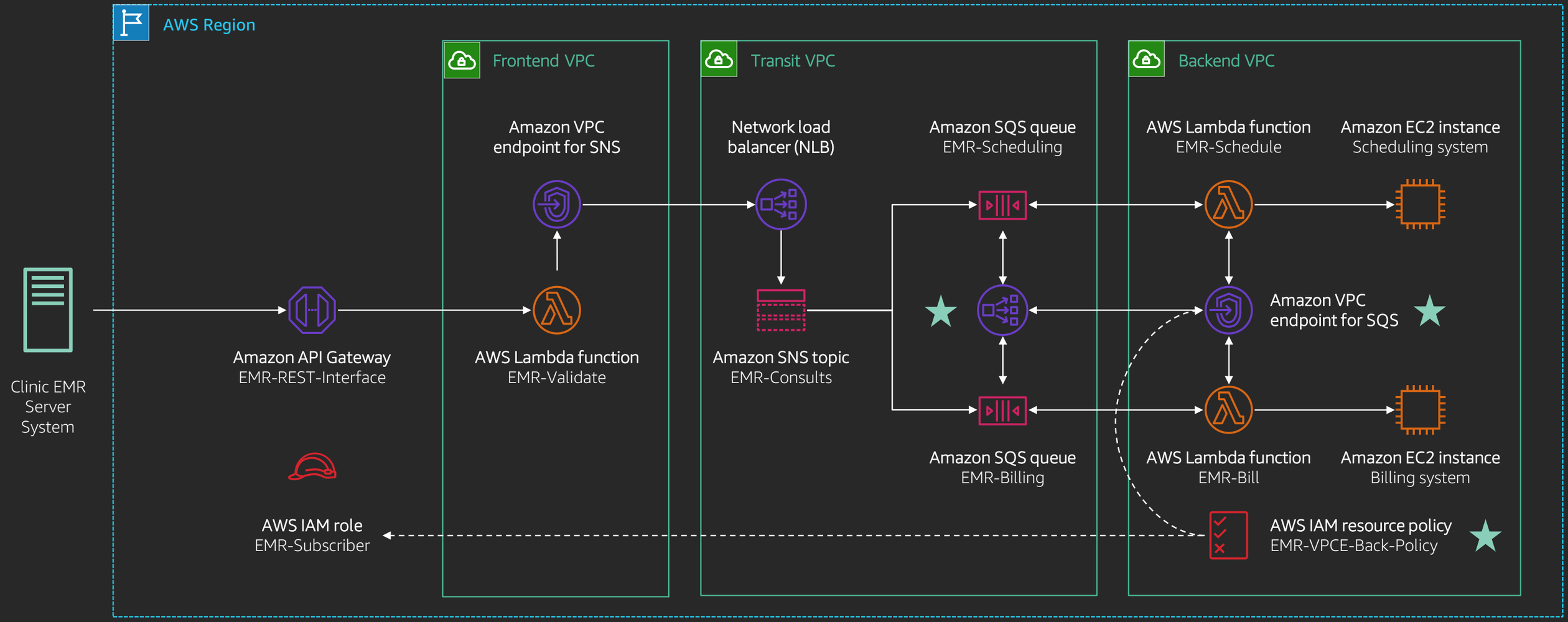
```
$ aws ec2 describe-vpc-endpoint-services

$ aws ec2 create-vpc-endpoint
    --service-name com.amazonaws.us-east-1.sns
    --vpc-endpoint-type Interface
    --vpc-id vpc-ec43eb81
    --subnet-id subnet-0931fc2fa5f1cbe45
    --security-group-id sg-1a2b3c4e
```

Securing the architecture | Message privacy

```
{
  "Version": "2012-10-17",
  "Id": "EMR-VPCE-Front-Policy",
  "Statement": [
    {
      "Sid": "SnsVpcePublish",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam:123456789012:role/EMR-Publisher"
      },
      "Action": "sns:Publish",
      "Resource": "arn:aws:sns:us-east-1:123456789012:EMR-Consults",
    }
  ]
}
```

Securing the architecture | Message privacy



Securing the architecture | Message privacy

```
$ aws ec2 describe-vpc-endpoint-services

$ aws ec2 create-vpc-endpoint
    --service-name com.amazonaws.us-east-1.sqs
    --vpc-endpoint-type Interface
    --vpc-id vpc-ec43eb82
    --subnet-id subnet-0931fc2fa5f1cbe47
    --security-group-id sg-1a2b3c4f
```

Securing the architecture | Message privacy

```
{
  "Version": "2012-10-17",
  "Id": "EMR-VPCE-Back-Policy",
  "Statement": [
    {
      "Sid": "SqsVpceReceiveMessage",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam:123456789012:role/EMR-Subscriber" },
      "Action": [ "sqs:ReceiveMessage", "sqs:DeleteMessage", "sqs:GetQueueAttributes" ],
      "Resource": [
         "arn:aws:sqs:us-east-1:123456789012:EMR-Scheduling",
         "arn:aws:sqs:us-east-1:123456789012:EMR-Billing"
      ],
    }
  ]
}
```

Our progress so far…

	Security scope	Addressed yet?
Authentication	Defines identity for each user	Yes
Authorization	Defines resource access permissions for each user	Yes
Message encryption	Prevents data from being read in transit and at rest	Yes
Message privacy	Prevents data from traversing the public internet	Yes
Auditing	Keeps track of resource access over time	-

Securing a serverless architecture
Auditing

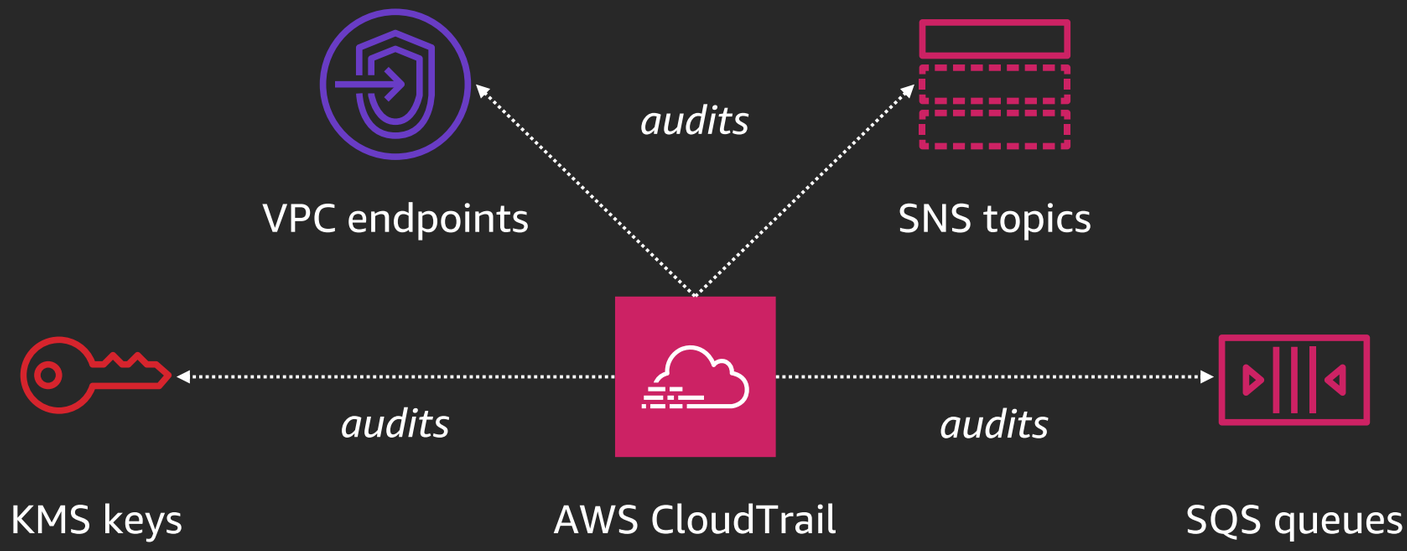
AWS re:Invent

aws

Securing the architecture | Auditing

API usage auditing

- API requests are logged by AWS CloudTrail
- Audit scope includes AWS KMS crypto operations executed, VPC endpoints accessed, and SNS/SQS resource policy changes



Securing the architecture | Auditing (AWS KMS operation)

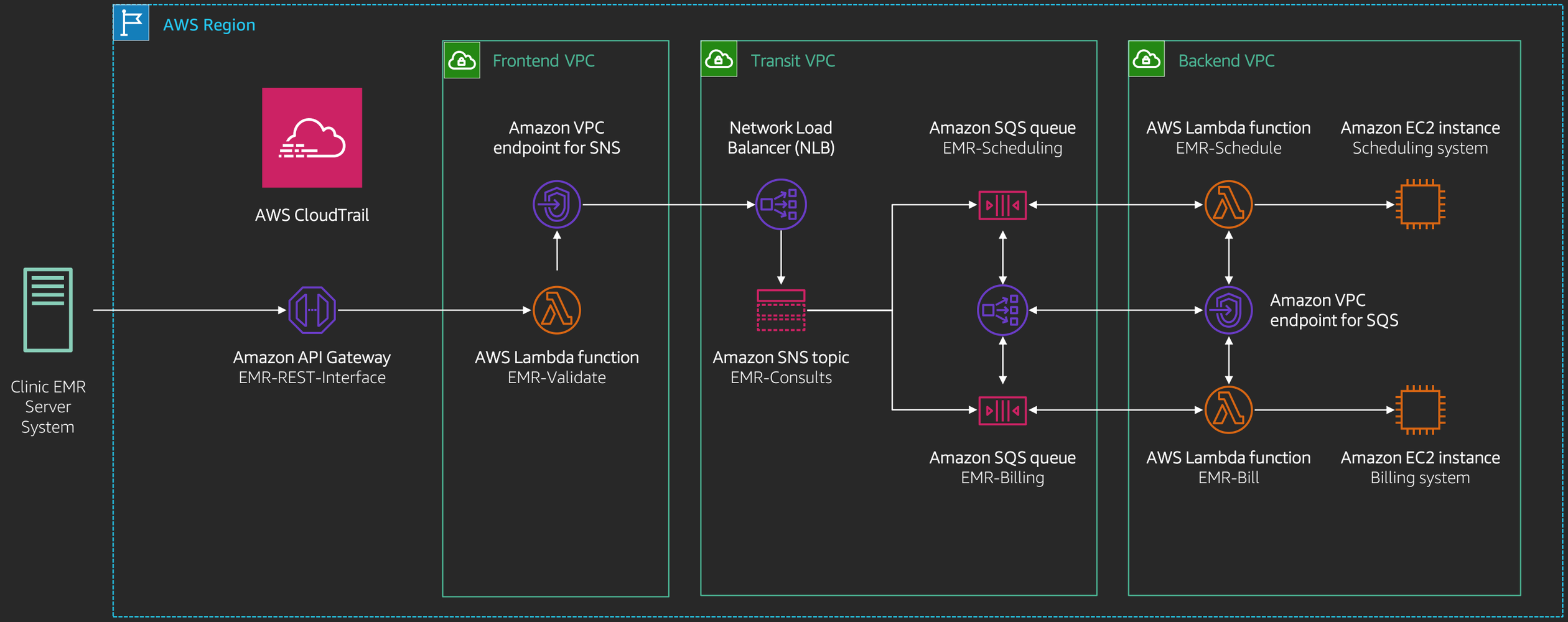

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAJ7PQSU42LKEHOPNEC:caller",
        "arn": "arn:aws:sts::453276652360:assumed-role/EMR-Caller",
        "accountId": "313276652360",
        "accessKeyId": "ASIAIS7TEZ35KHNWG5JQ",
        "sessionContext": {…},
        "invokedBy": "sns.amazonaws.com”
    },
    "eventID": "ab1sdfb9-54e0-4f64-b771-45032136e7cd",
    "eventType": "AwsApiCall",
    "eventTime": "2019-11-17T20:08:01Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "sns.amazonaws.com",
    "userAgent": "sns.amazonaws.com",
    "errorCode": "AccessDenied",
    "errorMessage": "User is not authorized to perform kms:GenerateDataKey",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "a809006a-5daf-46d1-81d5-6445dd62047d",
    "recipientAccountId": ”453276652360”
}
```

Securing the architecture | Auditing (VPCE operation)


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAJ7PQSU42LKEHOPNEC:admin",
        "arn": "arn:aws:sts::313276652310:assumed-role/EMR-Admin",
        "accountId": "313276652310",
        "accessKeyId": "ASIAUR4F6LNEG2HZO5LL",
        "sessionContext": {…}
    },

    "eventID": "c7f665e7-a585-4466-8785-d2c35de2ce3d",
    "eventType": "AwsApiCall",
    "eventTime": "2019-11-19T07:06:49Z",
    "eventSource": "ec2.amazonaws.com",
    "eventName": "CreateVpcEndpoint",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "52.46.80.17",
    "userAgent": "console.ec2.amazonaws.com",
    "requestParameters": {…},
    "responseElements": {…},
    "requestID": "71c2084d-f8b8-4df9-8850-c5cbee23efc0",
    "recipientAccountId": "313276652310”
}
```

Securing the architecture | Message privacy



Our progress so far…

	Security scope	Addressed yet?
Authentication	Defines identity for each user	Yes
Authorization	Defines resource access permissions for each user	Yes
Message encryption	Prevents data from being read in transit and at rest	Yes
Message privacy	Prevents data from traversing the public internet	Yes
Auditing	Keeps track of resource access over time	Yes

Thank you!

Otavio Ferreira

Sr. Manager, Software Development
AWS Serverless / AWS Messaging
@otaviofff

AWS re:Invent

aws

Please complete the session survey in the mobile app.