
AWS
re:Invent
NOV. 28 – DEC. 2, 2022 | LAS VEGAS, NV

SVS302-R

Architecting secure serverless applications

Josh Kahn (he/him)

Tech Leader, Serverless
AWS

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless *shifts* security . . .

responsibility toward AWS

and

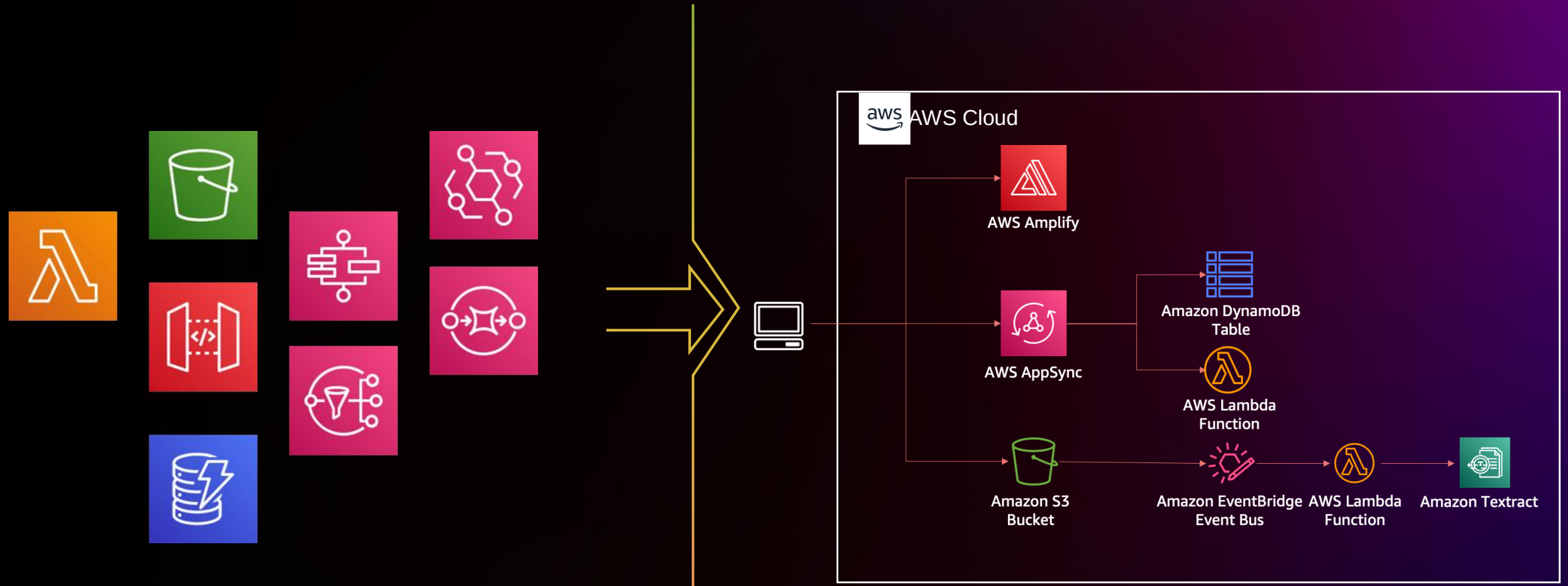
left in your organization

toward builders

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

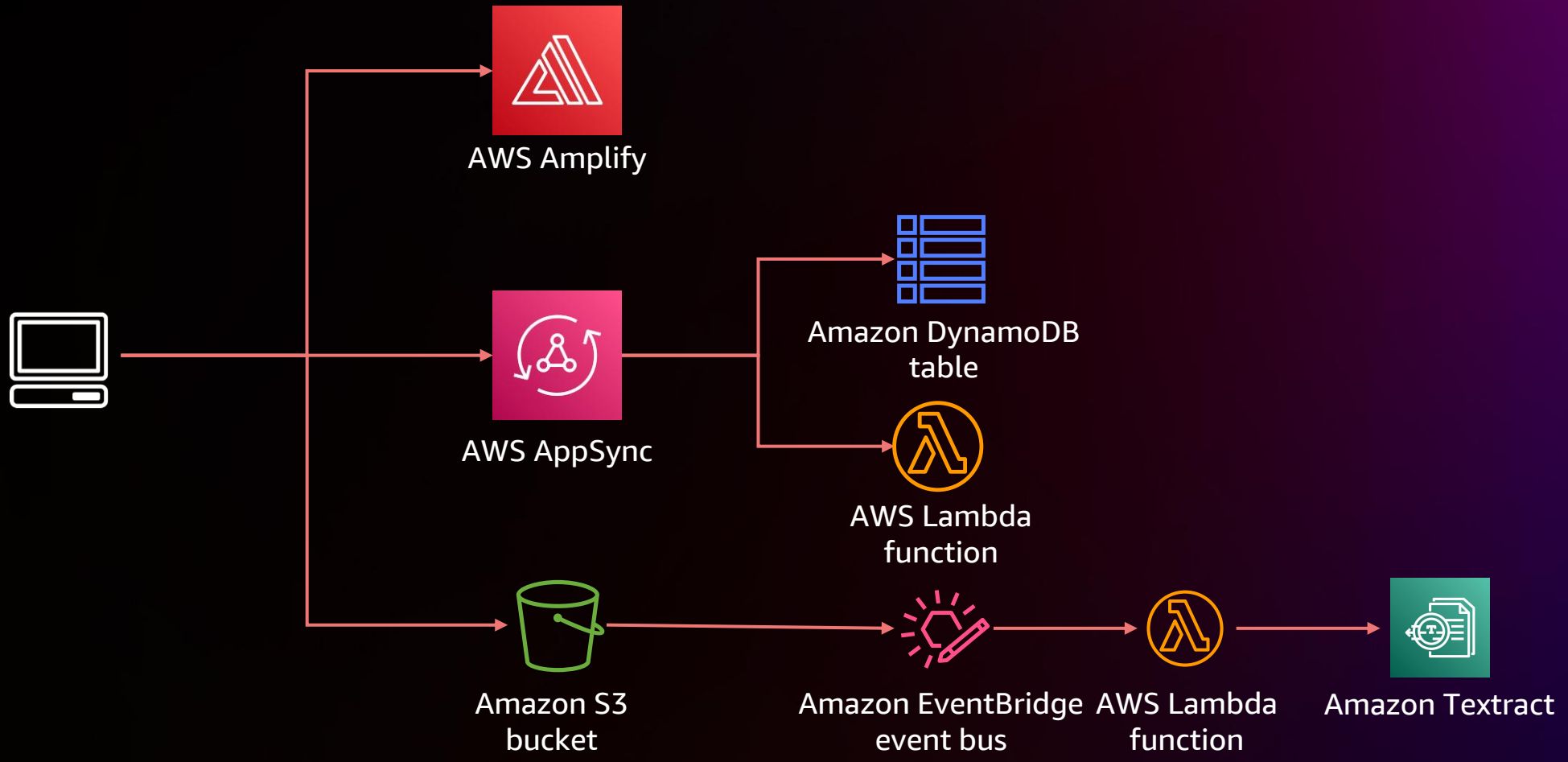
Small pieces, loosely joined

SERVERLESS ARCHITECTURES ARE COMPOSED OF MULTIPLE BUILDING BLOCKS



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

The most secure to-do app of all time?

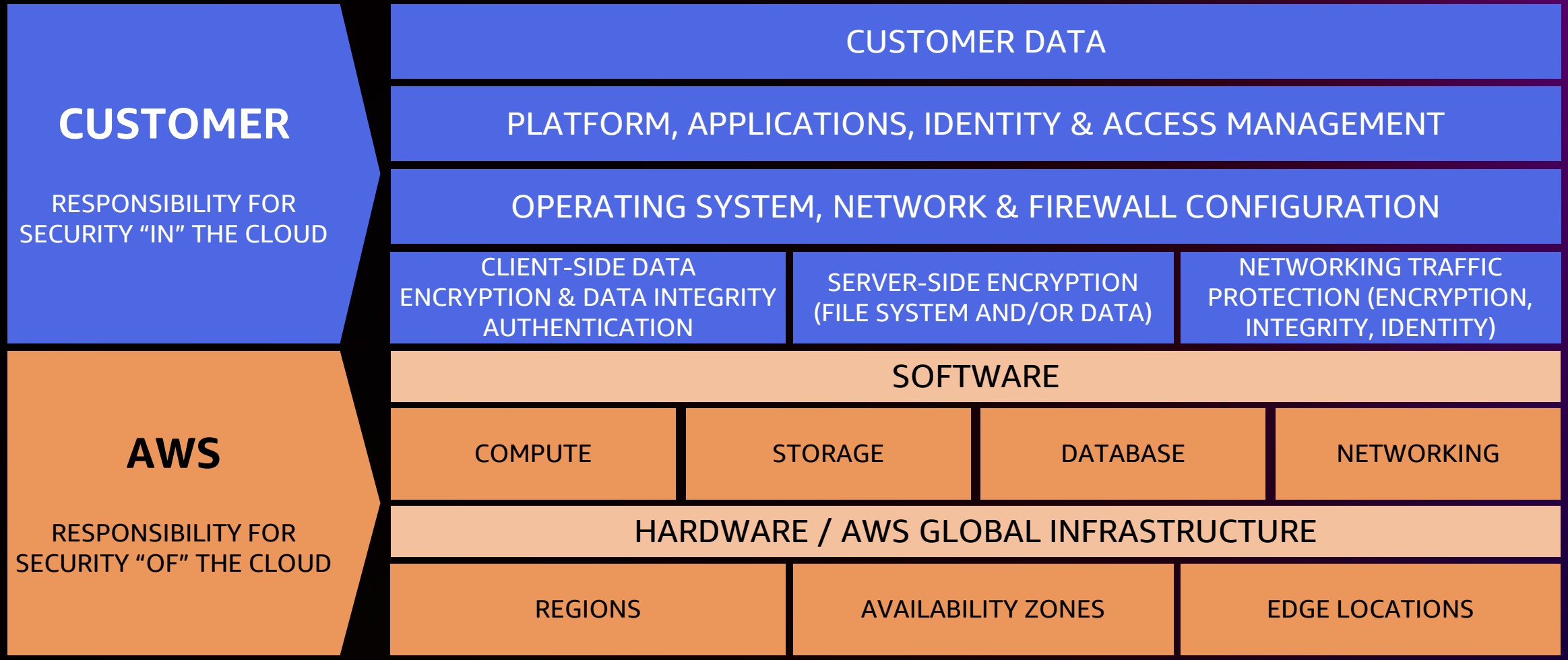


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless *shifts* security responsibility toward AWS

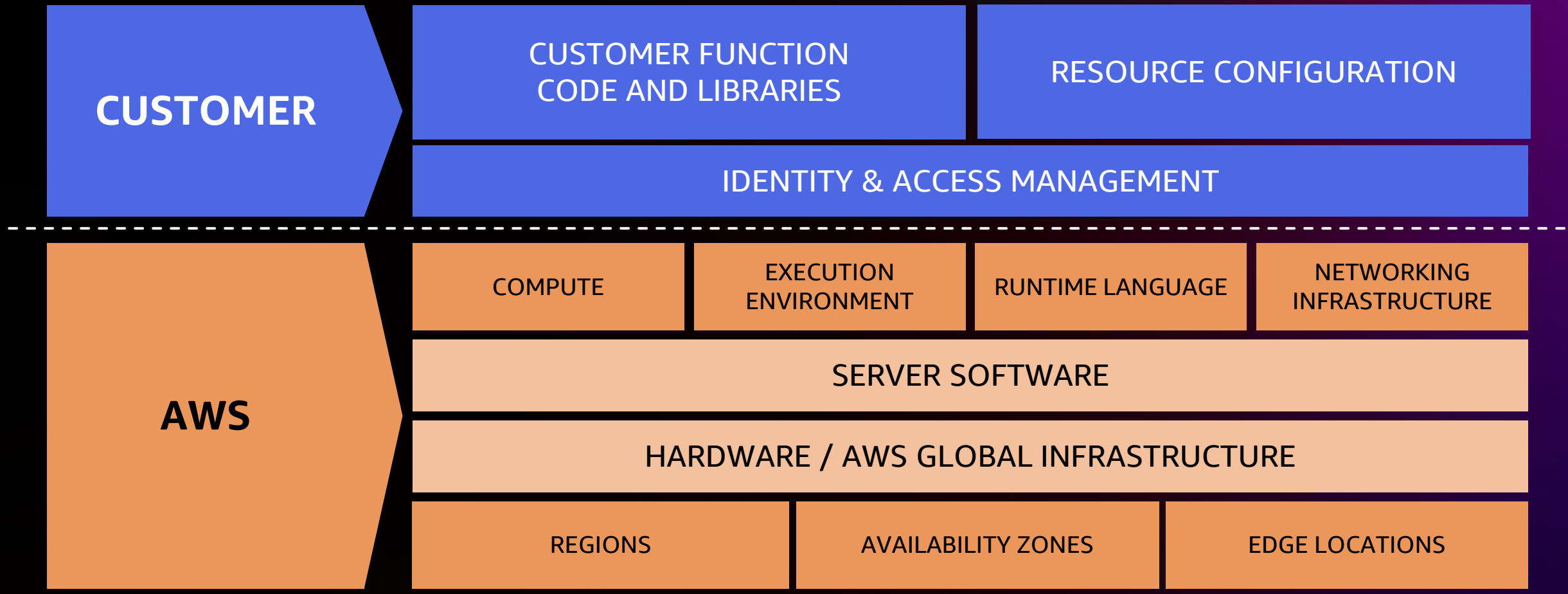
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Security is a shared responsibility



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Responsibility shifts to AWS with serverless

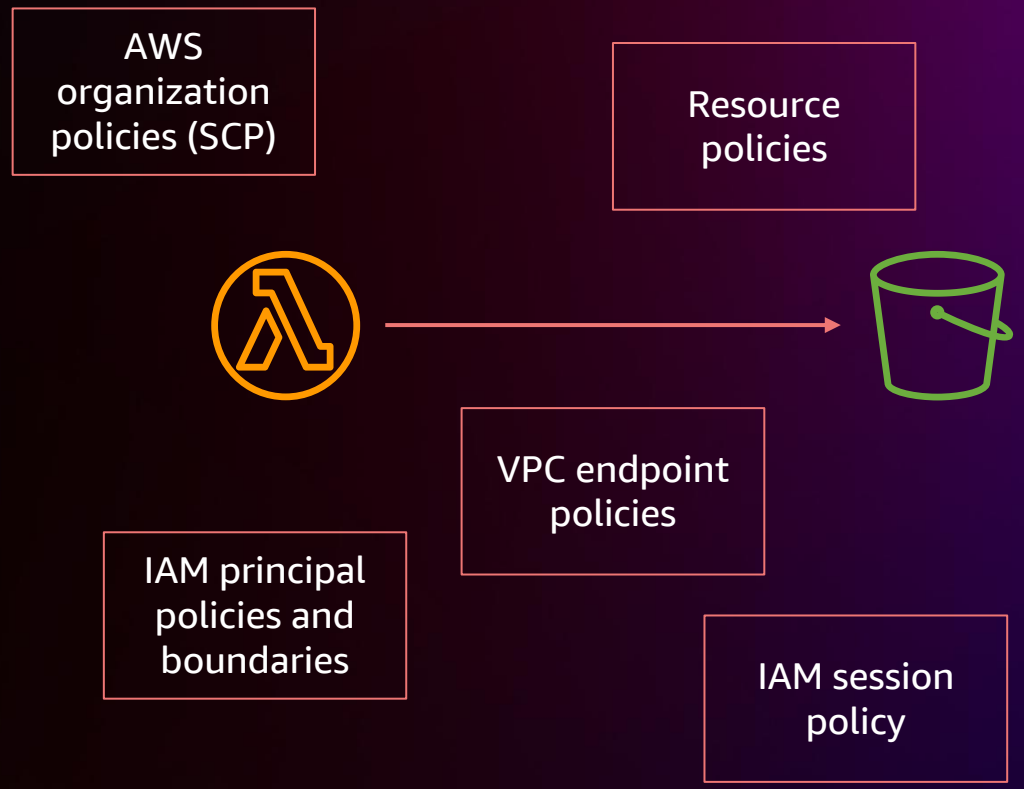


aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Control access to services with IAM

IAM deep dive

- AWS Identity and Access Management (IAM) permissions authorize access to:
  - **Control** plane actions that manage resources, such as:
    - `CreateFunction`
    - `UpdateStateMachine`
  - **Data** plane actions that execute the “daily business,” including:
    - `Invoke`
    - `PutEvents`



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Remember Log4Shell?

- Zero-day vulnerability in Log4j
- Possible for attacker to **execute arbitrary code** on server or leak sensitive data



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Case study: Lambda response to Log4Shell

- Lambda itself not affected
- **AWS Lambda** team applied mitigation built by **Amazon Corretto** team to managed Java runtimes
  - Base runtimes do not include Log4j, but customers running Java functions might
  - **Operational agility** enabled quick response at **scale**, across Java customers

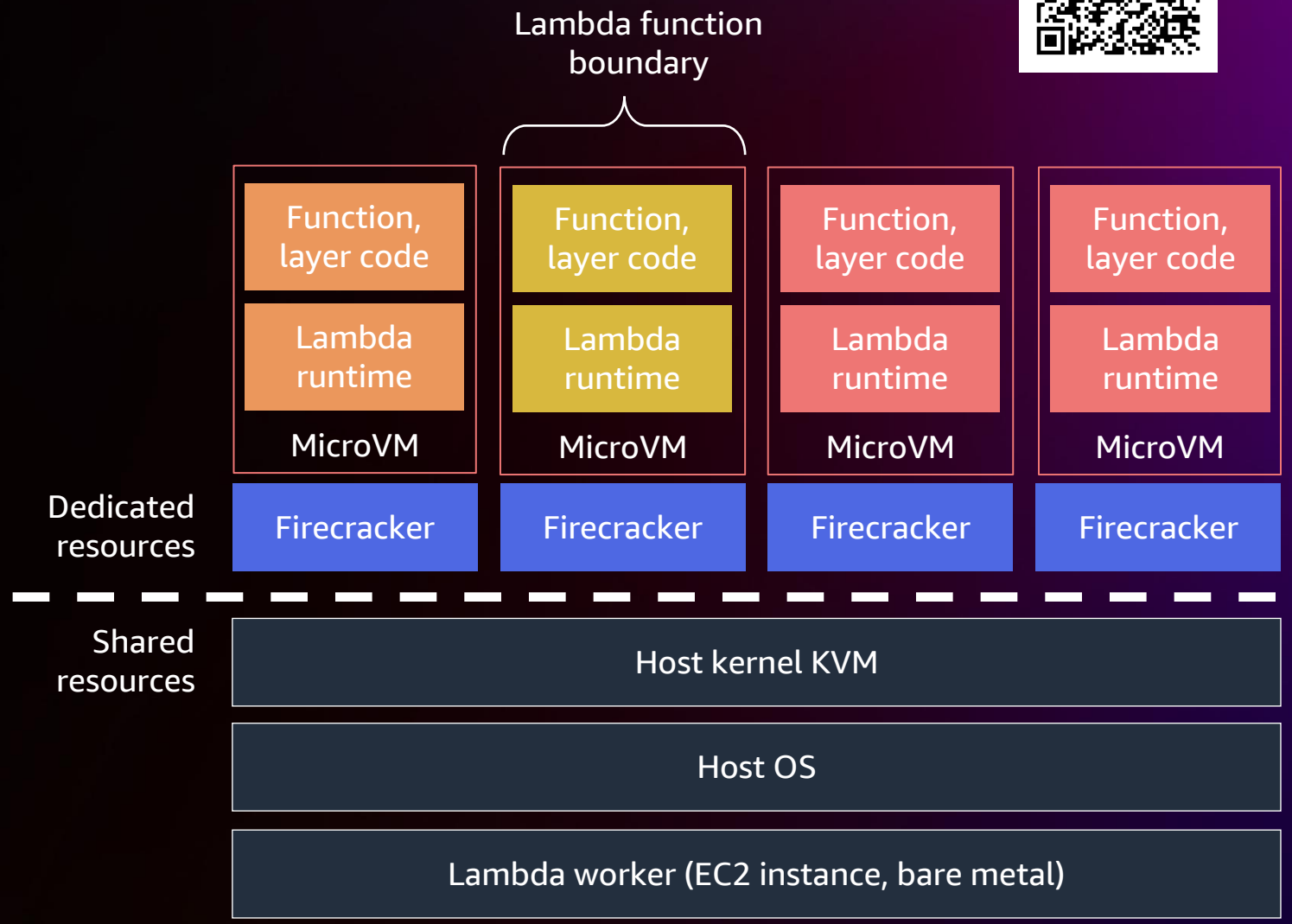


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda function isolation

Security overview of Lambda

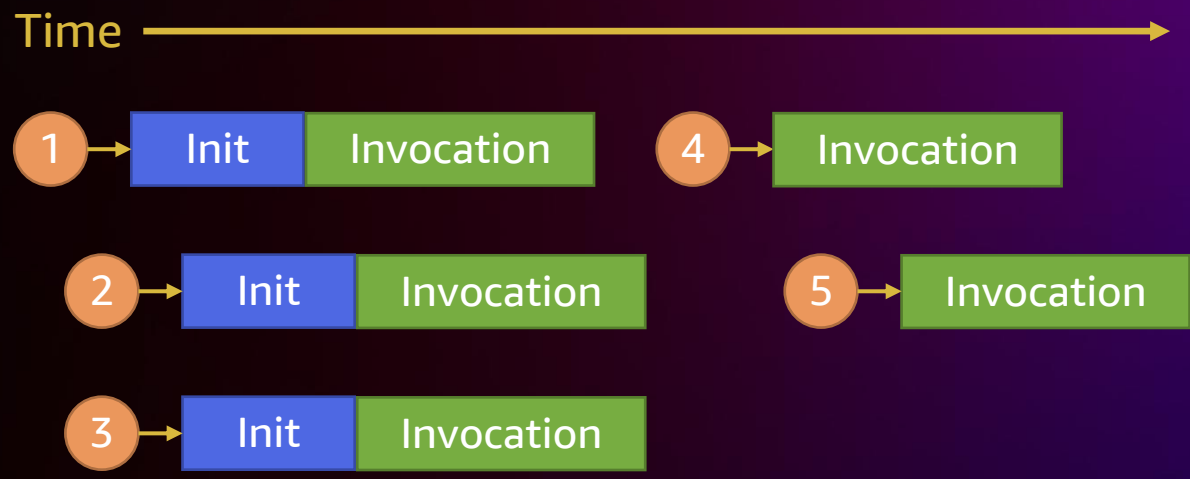
- Each function runs in a **dedicated** sandbox
  - Built on purpose-built virtualization technology: Firecracker
  - Provides enhanced security and workload isolation
- AWS maintains execution environment and **runtime**
  - Patching, etc.
  - AWS does not maintain runtime for custom runtime or container-packaged functions



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Lambda sandbox reuse

- Each sandbox **processes a single event at a time**
- Sandbox **may be reused** between invocations
  - Improves latency for warm environments
  - Reuse previously initialized connections
- Use caution when storing sensitive data in memory or `/tmp`



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless *shifts* security left in your organization

toward builders

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Modern approach shifts security left

Traditional approach

- Heavy emphasis on perimeter security and alarms
- Applications are deployed within the secured perimeter
  - “Security added at the end”

Modern approach

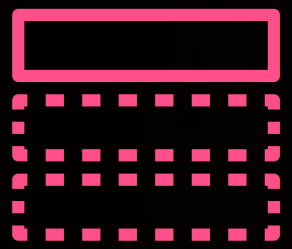
- Implement defense in depth
  - **Secure each component** of the workload
  - Focus on **IAM**
- Apply perimeter security if exposed to internet
  - Leverage AWS WAF, Amazon API Gateway, etc.
- Managed services provide a strong basis

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Least privilege

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Amazon SNS
Analysis complete topic



AWS Lambda
Handle handwriting analysis complete



Amazon DynamoDB
To Do Table



Amazon Textract

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Secure Lambda functions with IAM

Function policy

- Defines how function can be invoked
- Supports cross-account access
- Used for synchronous and asynchronous invocations

"Actions on API Gateway A can invoke Lambda function B"

Execution role

- Defines which AWS resources can access via IAM
- Used for poll-based invocations (Lambda polling)

"Lambda function A can put an object in S3 bucket B"



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Secure other resources with IAM

Resource policy

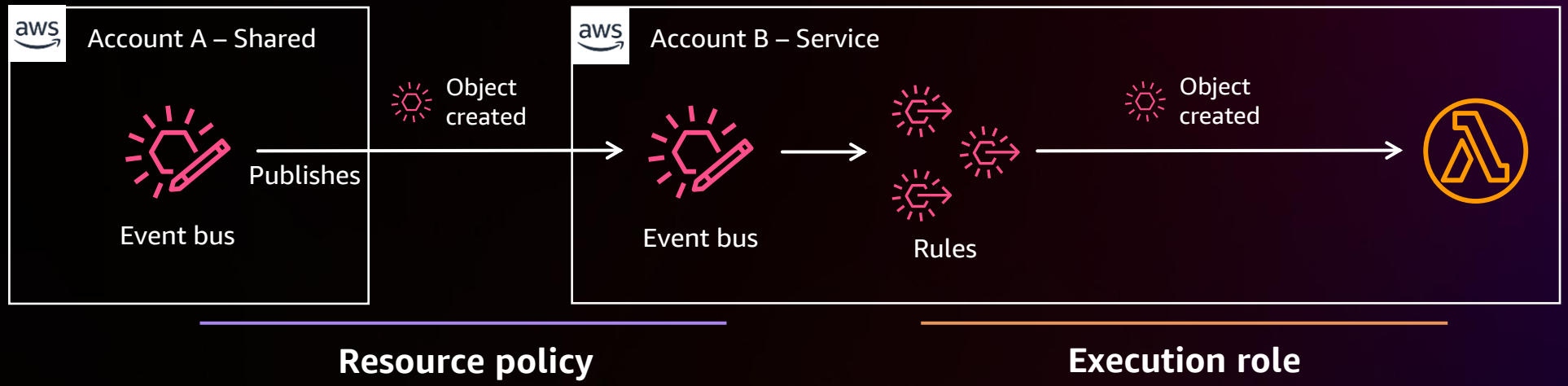
- Defines access to event bus, e.g., `PutEvent`
- Use conditions to scope permissions
- Can allow cross-account access

"Event bus A can publish events to event bus B"

Execution role

- Assumed by event bus to determine access to AWS resources
- Can allow cross-account access

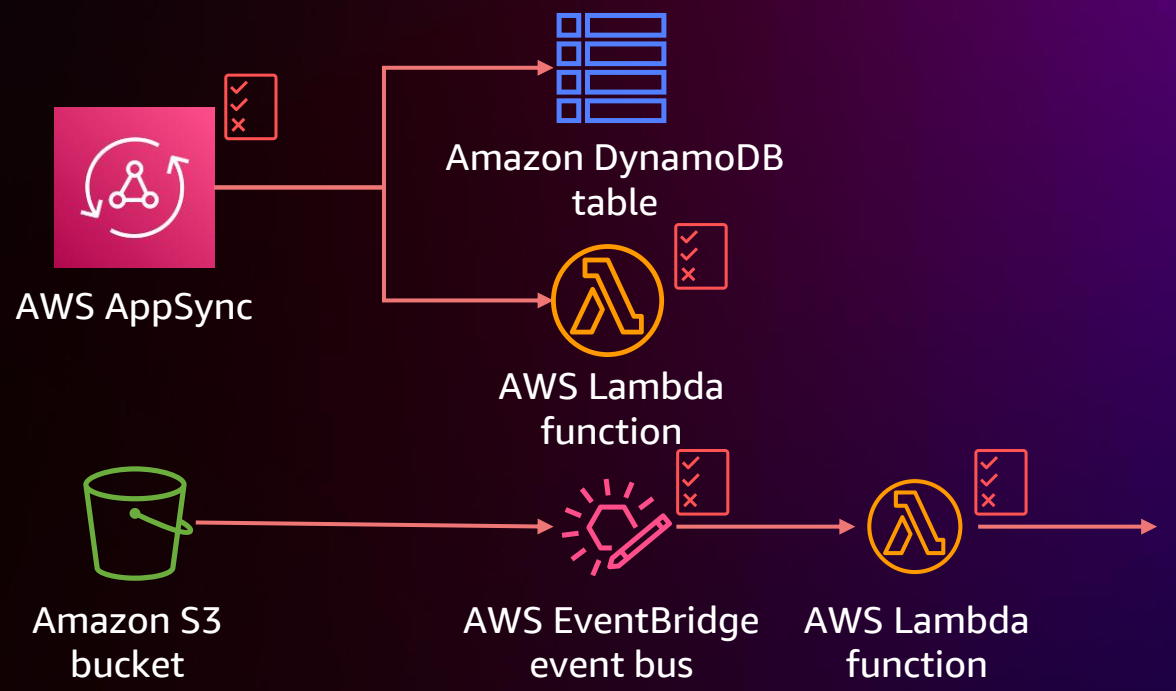
"Event bus A can invoke Lambda function B"



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Grant only essential privileges

- Least privilege is the set of **essential privileges** needed to perform intended work
- Attach via IAM execution role
  - Prefer **unique role per resource**
  - Assign **fine-grained permissions**
- Be specific: identify limited set of resources and actions allowed
  - Scrutinize use of “*”



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Least privilege requires developer discipline

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Least privilege with AWS SAM

TWO OPTIONS TO SIMPLIFY MANAGEMENT

Policy templates

```yaml
SaveToDynamoDBFunction:
  Type: AWS::Serverless::Function
  Properties:
    Handler: app.handler
    Runtime: nodejs16.x
    Policies:
      - DynamoDBWritePolicy:
          TableName: !Ref DDBTable
    Events:
      GetItem:
        Type: HttpApi
        Properties:
          Path: "/todo/{id}"
          Method: GET
```

SAM connectors

```yaml
DDBTable:
  Type: AWS::Serverless::SimpleTable
SaveToDynamoDBFunction:
  Type: AWS::Serverless::Function
  Properties:
    Runtime: nodejs16.x
    Handler: index.handler
    Environment:
      Variables:
        TABLE_NAME: !Ref DDBTable
MyConnector:
  Type: AWS::Serverless::Connector
  Properties:
    Source:
      Id: SaveToDynamoDBFunction
    Destination:
      Id: DDBTable
    Permissions:
      - Write
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Least privilege with AWS CDK

```
const saveToDynamoDBFunction = new NodejsFunction(this, "SaveToDynamoDBFunction", {
  entry: join(__dirname, "functions/save-to-ddb/app.ts"),
  handler: "handler",
  runtime: Runtime.NODEJS_16_X,
  environment: {
    TABLE_NAME: DDBTable.tableName
  }
});

DDBTable.grantWrite(saveToDynamoDBFunction);
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Grant specific function access to resource

- Specify Lambda function ARN as policy condition
  - Lambda will **automatically** add condition key to AWS API request context
- Write conditional policies with `lambda:SourceFunctionArn`
- Apply to SCP or identity-based policies

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutObjectFromUploadFunction",
      "Effect": "Allow",
      "Action": [
        "dynamodb:PutItem",
        "dynamodb:UpdateItem"
      ],
      "Resource": "arn:aws:dynamodb:us-east-1:*:table/DDBTable",
      "Condition": {
        "ArnEquals": {
          "lambda:SourceFunctionArn":
            "arn:aws:lambda:us-east-1:123456789012:function:upload_function"
        }
      }
    },
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface",
        "ec2:AssignPrivateIpAddresses",
        "ec2:UnassignPrivateIpAddresses"
      ],
      "Resource": "*"
    }
  ]
}
```

aws

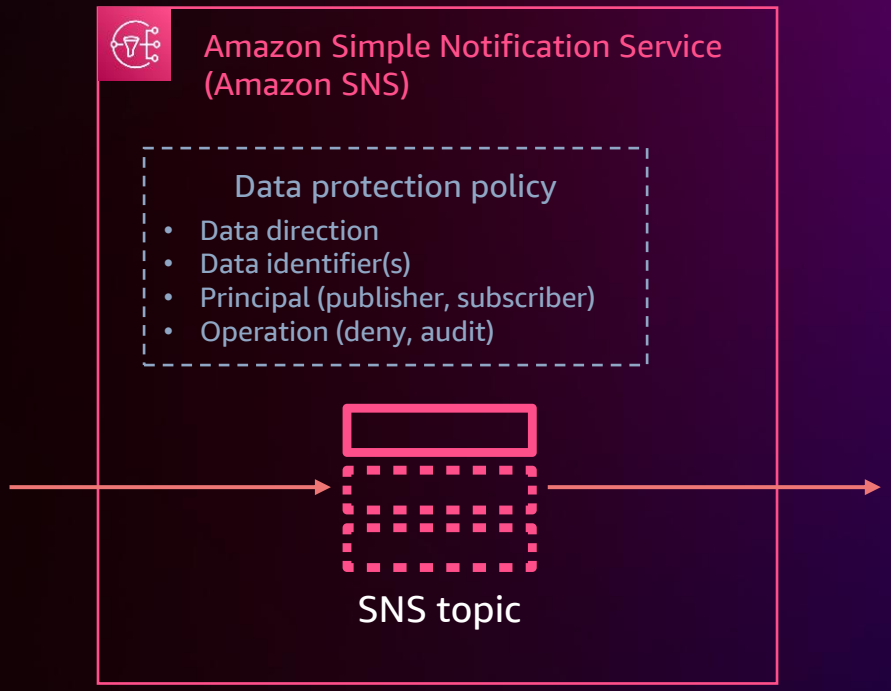
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Defense in depth

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Rich data protection options

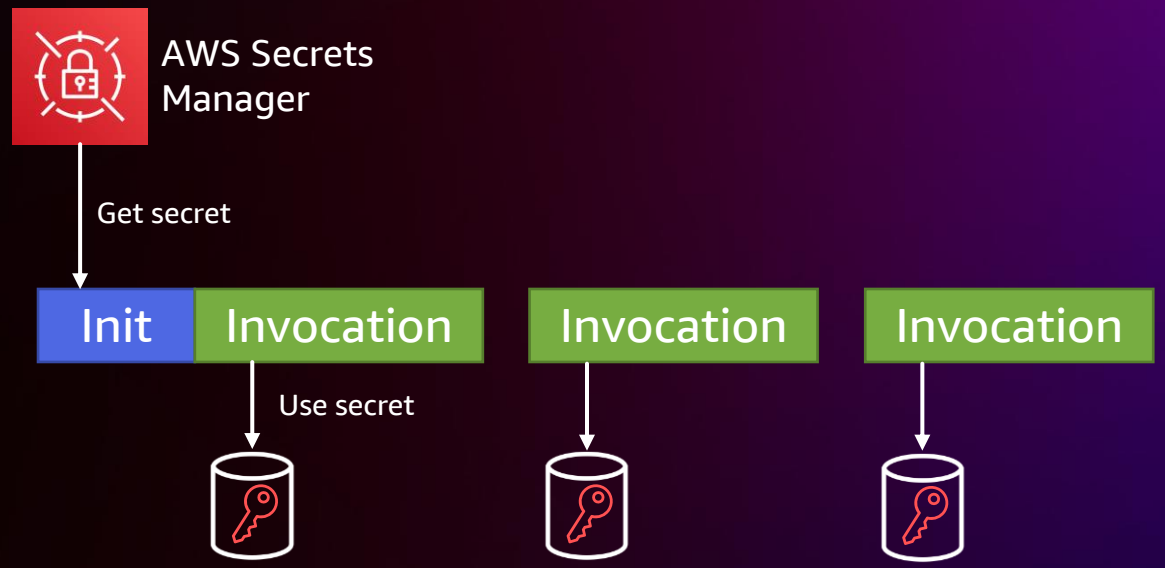
- Encryption in-transit and at-rest **provided out-of-the-box**
  - SQS, SNS, Lambda (env vars) support customer-managed keys
- NEW! New SQS queues are **encrypted by default** using SSE-SQS
- NEW! SQS support for attribute-based access controls (**ABAC**)
- NEW! Discover and protect **sensitive data in motion** with SNS Data Protection



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Securely retrieve secrets in Lambda functions

- Separate secrets from code/ configuration using external manager
  - Tightly control access to secrets
- Load secret on `init`, each `invoke` in function, or via Lambda extension
  - Requires **appropriate execution role**
  - Store secret in memory or on disk
  - Ephemeral environment, encrypted `/tmp`
- Consider **scaling** behavior



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Validate untrusted event payloads

- Validate input **before processing**, before parsing
  - Test both positive and negative cases
  - Consider fuzz testing
- Use **strict typing**
  - Parameters should not accept more than one type of data primitive
  - Apply constraints, additional validation to ambiguous types (e.g., String)
- Consider for **all event sources**
  - Particularly important for APIs

```python
import input_schemas as schemas

from aws_lambda_powertools.utilities import parameters
from aws_lambda_powertools.utilities.typing import LambdaContext
from aws_lambda_powertools.utilities.validation import validator

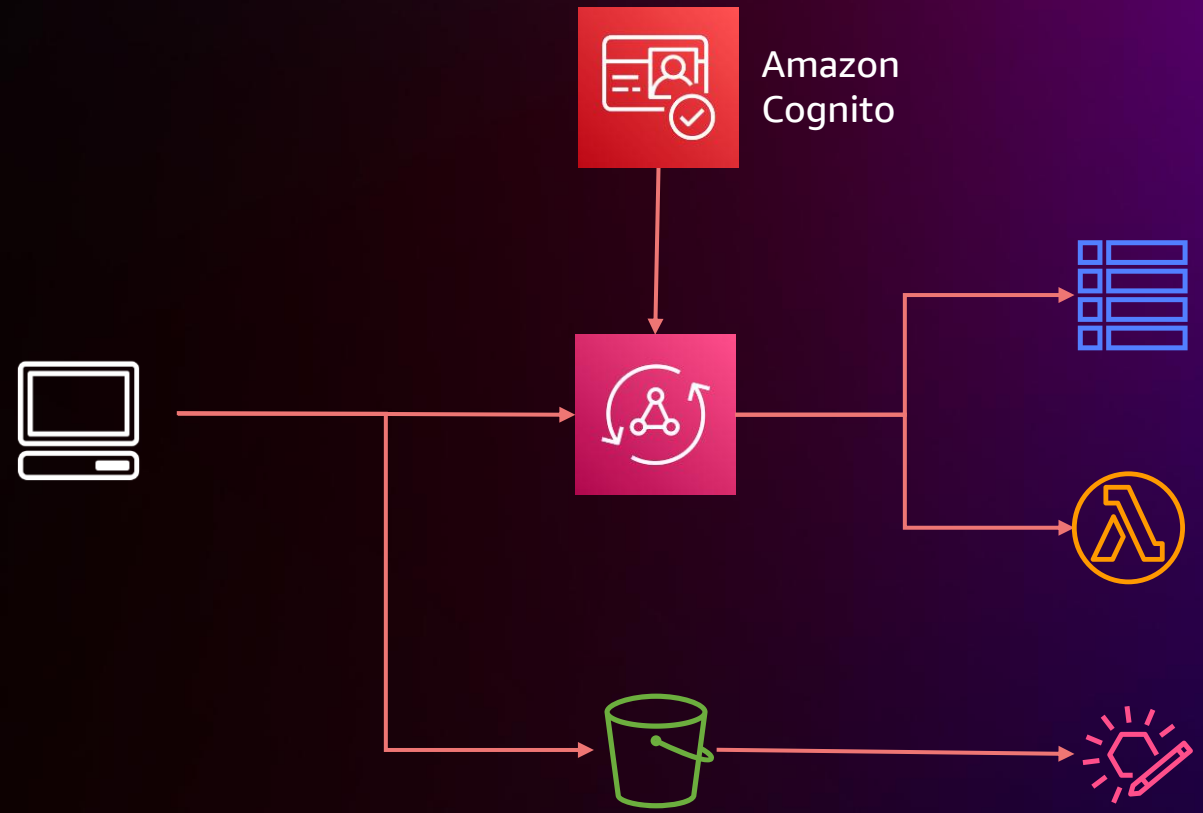
@validator(inbound_schema=schemas.INPUT)
def lambda_handler(event, context: LambdaContext) -> dict:
  # do something
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Offload authorization at the front door

- **Simplify** development with shared functionality
  - Implemented by dedicated team
- **Separate concerns** from business logic



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Other managed API options

	API Gateway	AppSync	ALB	Lambda function URL
Authorization	IAM, Cognito, Lambda	IAM, Lambda, Cognito OIDC, API Key	OIDC, Cognito	None, IAM
Request validation	JSON schema	Typed	N/A	N/A
WAF integration	Yes	Yes	Yes	No
Private network access	Yes	No	Yes	No
Rate limiting/ throttling	Yes	No	No	No
mTLS	Yes	No	No	No

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Development

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

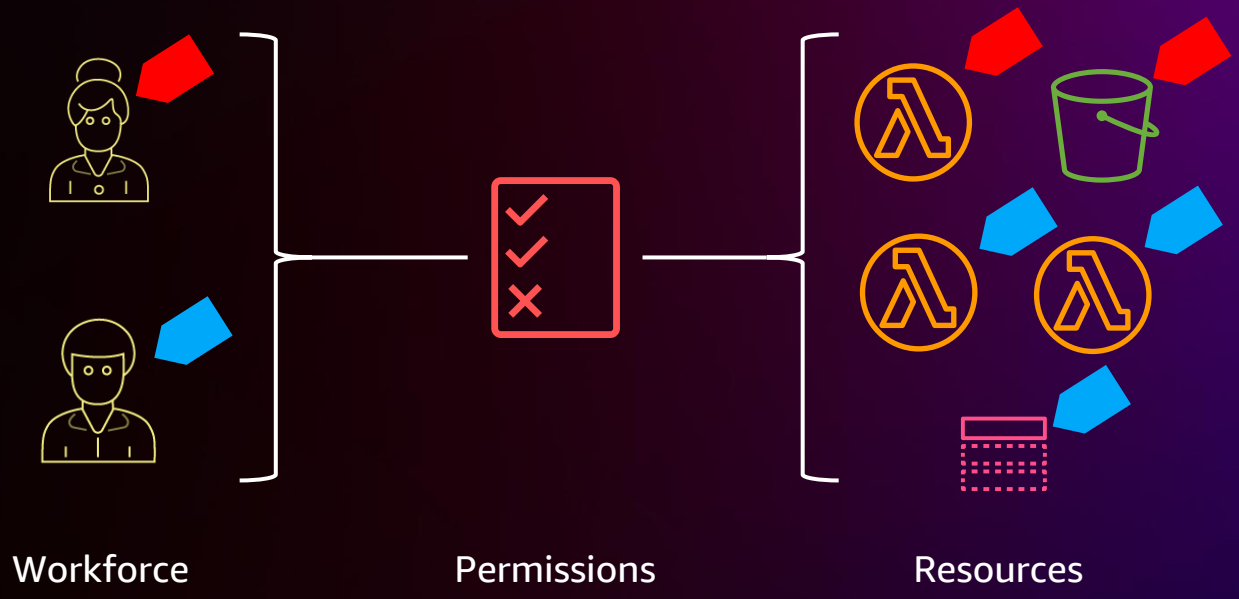
Use a framework!

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Leverage ABAC to scale workforce

Scaling Lambda permissions w/ ABAC

- Use IAM to control access to resource configuration
- **Attribute-based** access control (ABAC) relies on key-value pairs
  - Attributes implemented as tags on AWS
  - Use attributes for cost allocation, discovery, and access control



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Enforce tagging on function creation

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowLambdaTagOnCreate",
      "Effect": "Allow",
      "Action": [
        "lambda:CreateFunction",
        "lambda:TagResource"
      ]
      "Resource": "arn:aws:lambda:*:*:function:*",
      "Condition": {
        "StringEquals": {,
          "aws:RequestTag/project-name": "${aws:PrincipalTag/project-name}"
        },
        "ForAllValues:StringEquals": {
          "aws:TagKeys": [
            "project-name"
          ]
        }
      }
    }
  ]
}
```

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Allow to invoke functions in same project

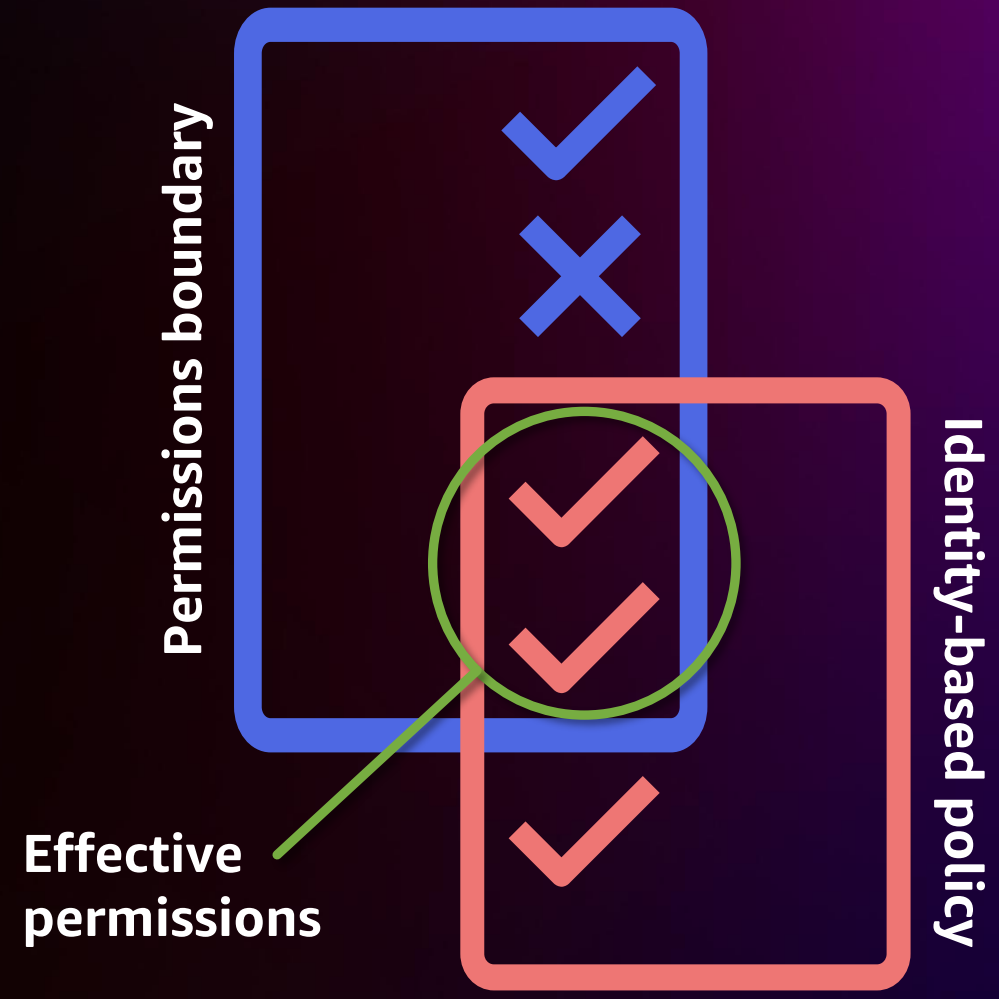
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowInvokeLambdaSameProject",
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": "arn:aws:lambda:*:*:function:*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/project-name": "${aws:PrincipalTag/project-name}"
        }
      }
    }
  ]
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

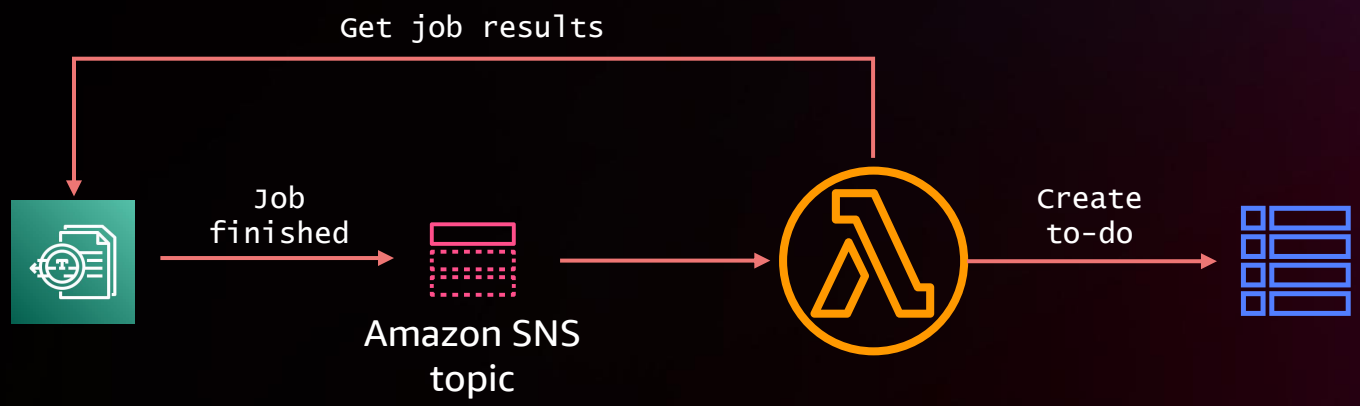
Empower developers to safely create policies

- Permissions boundaries are **attached to roles and limit their actions**
  - Can explicitly deny actions (`Deny`)
  - Can implicitly deny actions (no `Allow`)
  - Never grant an entitlement
- Permission boundaries are **guardrails for builders to manage permissions**
  - Identify maximum allowable permissions



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Create to-do on handwriting job completion





aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Developer role ensures boundary is used

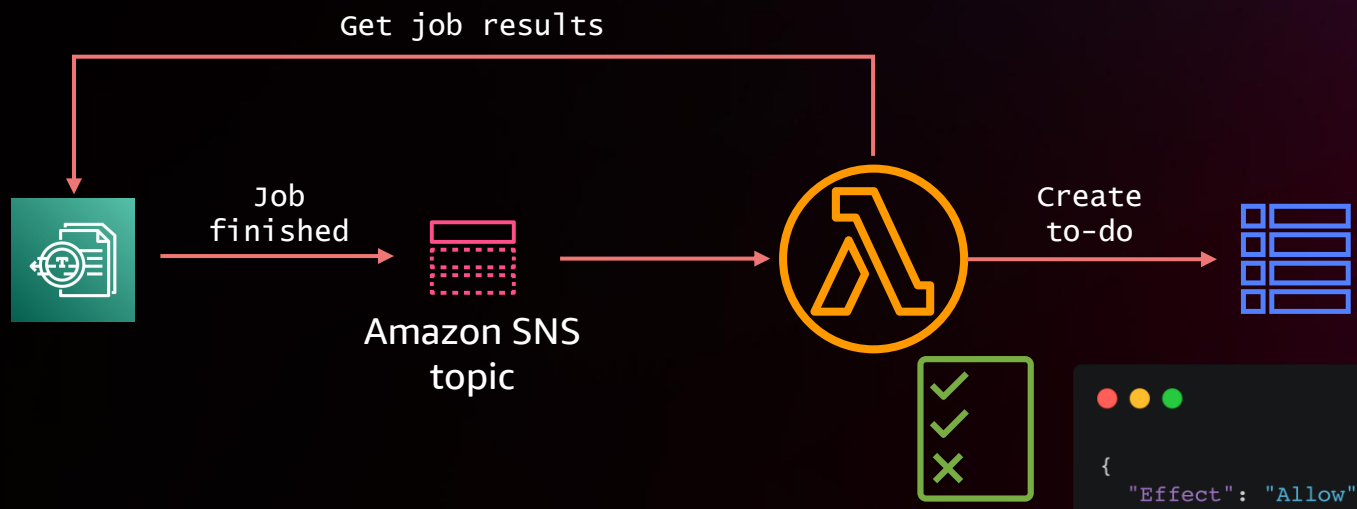


```
{
  "Effect": "Deny",
  "Action": [
    "iam:AttachRolePolicy",
    "iam:CreateRole",
    "iam:DetachRolePolicy",
    "iam:PutRolePolicy",
    "iam:DeleteRolePolicy",
    "iam:PutRolePermissionsBoundary"
  ],
  "Resource": "*",
  "Condition": {
    "StringNotLike": {
      "iam:PermissionsBoundary": "arn:aws:iam::*:policy/permissionboundarypolicy"
    }
  }
}
```

aws

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Developer creates function permissions



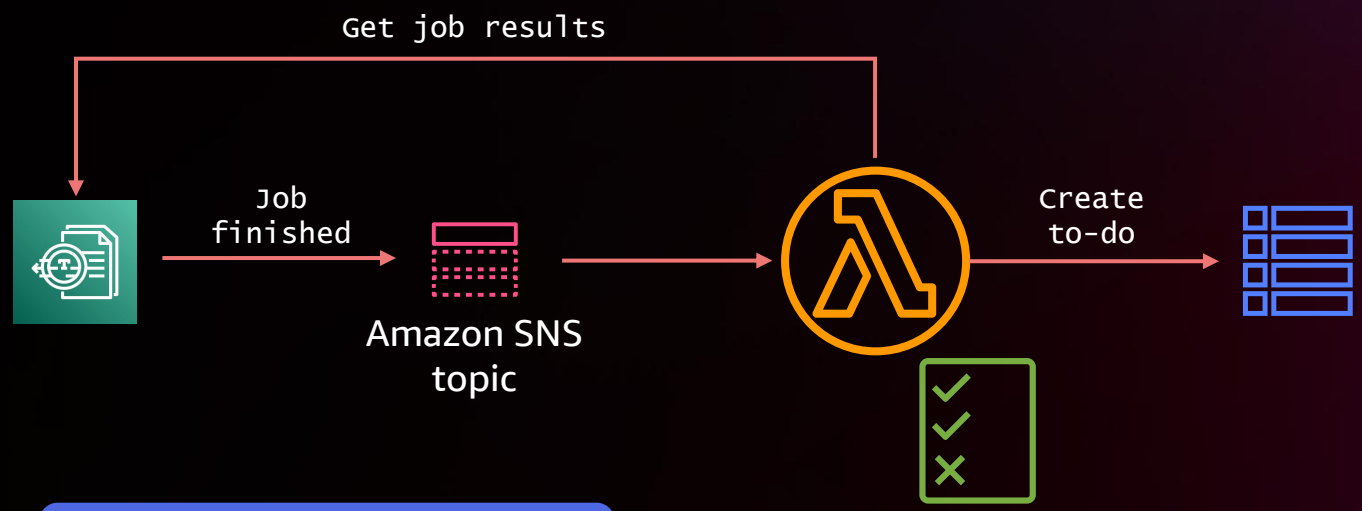


```
{
  "Effect": "Allow",
  "Action": [
    "dynamodb:PutItem"
  ],
  "Resource": "arn:aws:dynamodb:us-east-1:1234567890:table/..."
},
{
  "Action": "textract:GetDocumentTextDetection",
  "Resource": "*",
  "Effect": "Allow"
}
```

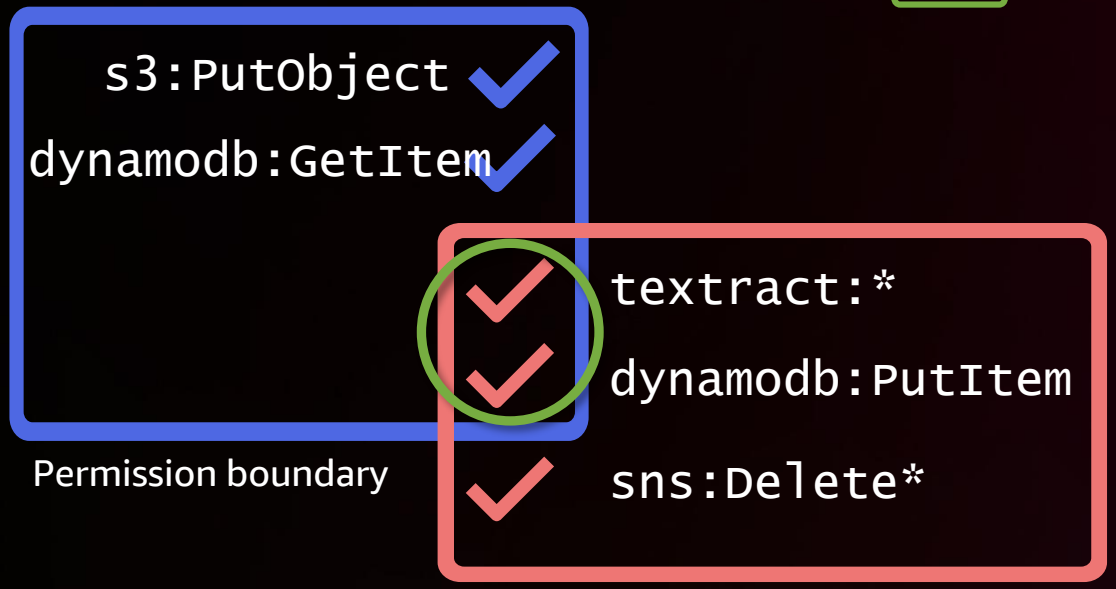
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Effective permissions within boundary

Permission boundaries sample





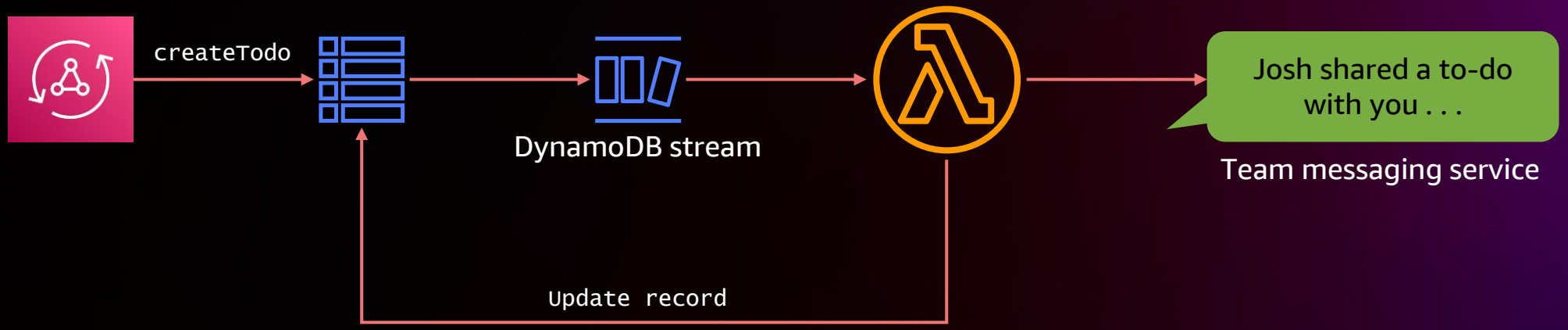


© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Networking

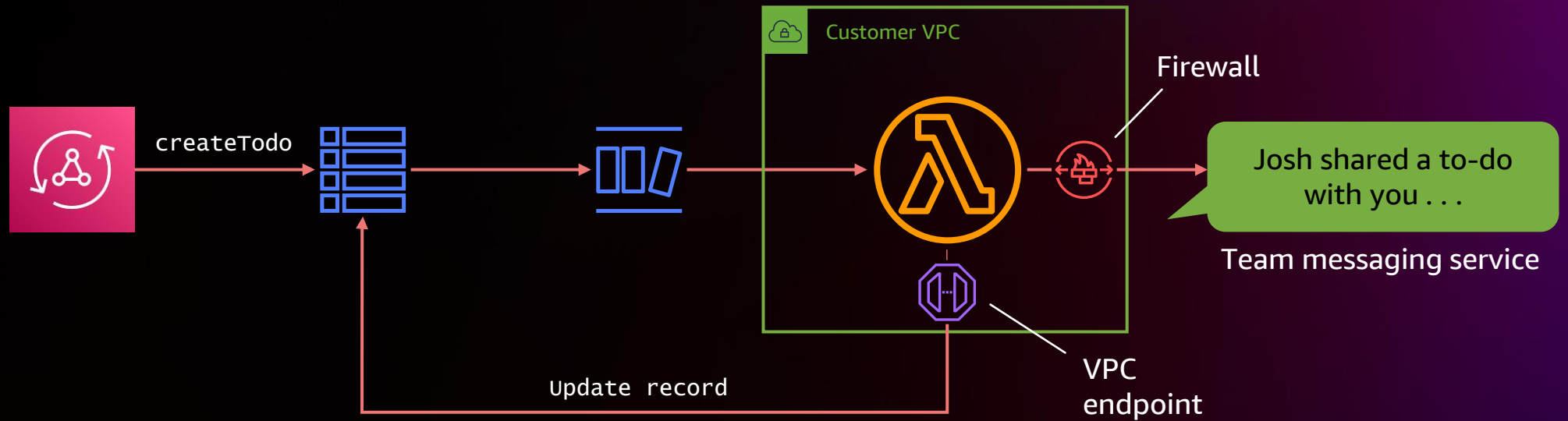
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Sharing to-dos via team messaging system



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

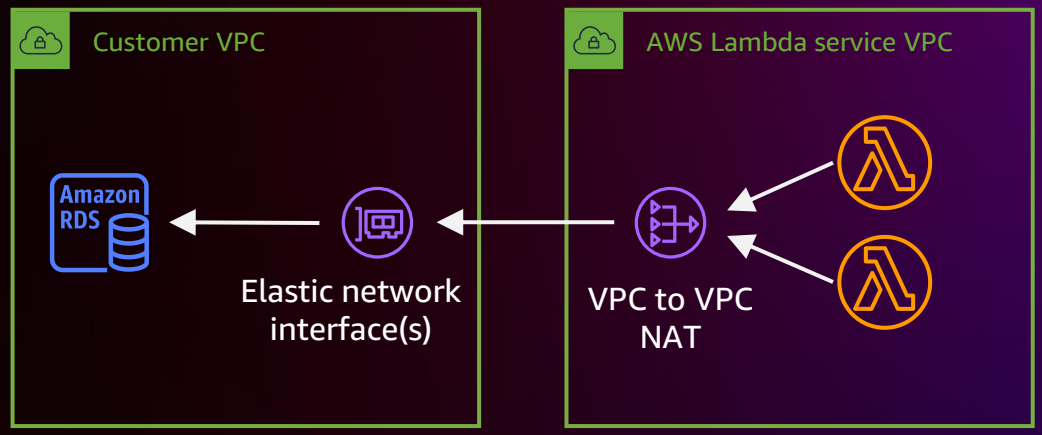
Attach function to VPC to control networking



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Should my Lambda function be VPC-enabled?

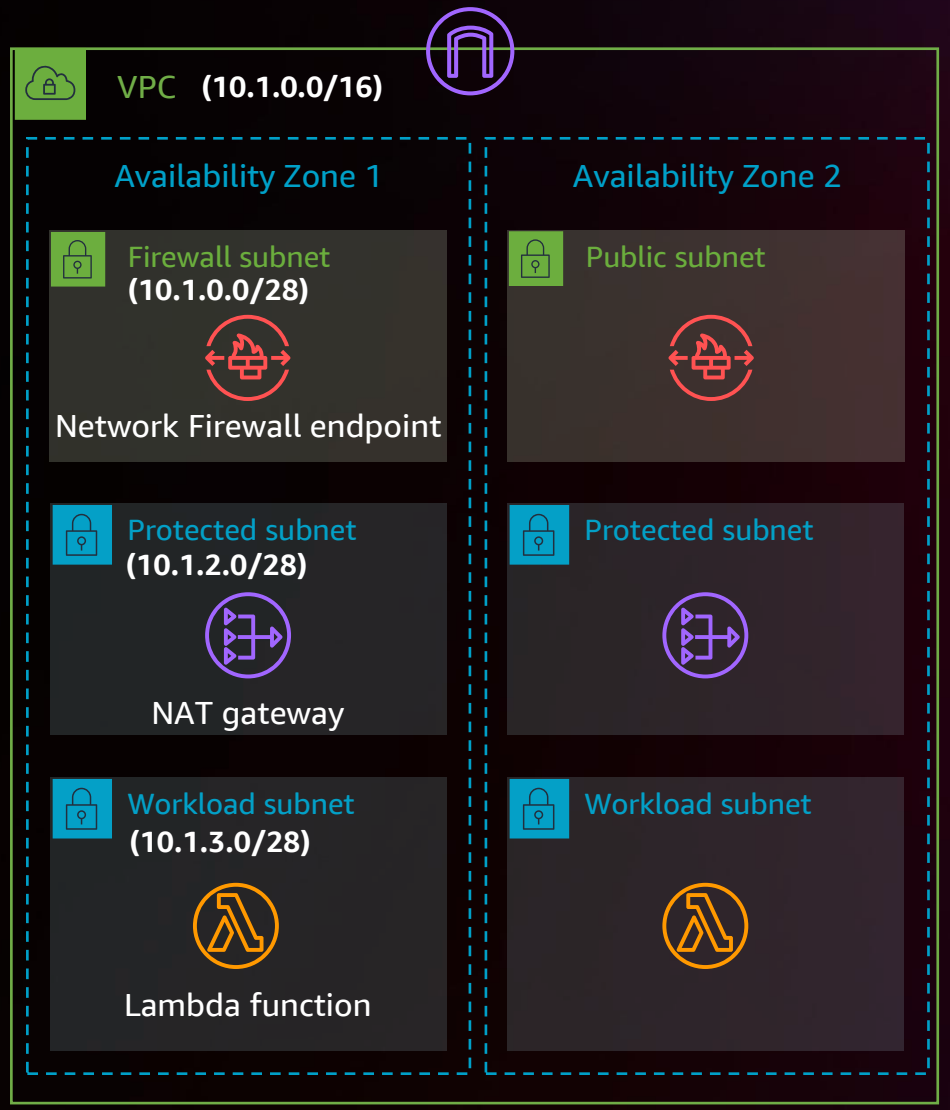
- Lambda functions **always run in VPCs owned by the Lambda service** team
  - When VPC-enabled, configured with **access to your VPC via an ENI**
- Lambda functions are **always invoked** via `Invoke` action
  - Access controlled by IAM
- Only if (1) **function needs access to VPC resource** or (2) must **meet a security control**



aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Implementing egress control and flow logs

- Leverage **AWS Network Firewall** (or similar) to implement egress controls
  - Employ allow-list to protect against exfiltration
  - Consider **VPC endpoints** for access to AWS services
- Enable flow logs for post-event analysis
- Approach incurs additional **cost** and **complexity**



IGW ingress route table	
Destination	***Target***
10.1.0.0/16	local
10.1.2.0/28	vpce-id (firewall)

Firewall route table (AZ1)	
Destination	***Target***
10.1.0.0/16	local
0.0.0.0/0	igw

Protected route table (AZ1)	
Destination	***Target***
10.1.0.0/16	local
0.0.0.0/0	vpce-id (firewall)

Workload route table (AZ1)	
Destination	***Target***
10.1.0.0/16	local
0.0.0.0/0	nat-gw-id

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Serverless shifts security . . .

- Responsibility toward AWS
  - Understand **Shared Responsibility Model**
  - Leverage **IAM** to control access
- Left in your organization (toward builders)
  - Enforce least privilege; it requires **developer discipline**
  - Implement defense in depth **multiple, redundant measures** across system
  - Follow best development practices (check out **AWS Lambda Powertools**)
  - Leverage ABAC, permissions boundaries to enable developer agility

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Check out these other sessions

SVS309: Securing serverless applications workshop

Tuesday (Nov 29) @ 11:45a – Wynn Cristal 5
Wednesday (Nov 30) @ 3:15p – Mandalay Bay Ballroom A

SVS404: A closer look at AWS Lambda

Tuesday (Nov 29) @ 5:45p – MGM Grand 121
Wednesday (Nov 30) @ 10:00a – Mandalay Bay South Pacific F

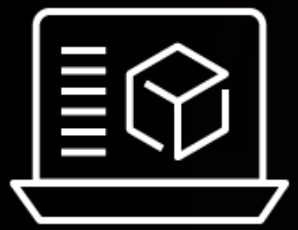
SVS401: Best practices for advanced serverless developers

Tuesday (Nov 29) @ 12:30p – Mandalay Bay South Pacific F

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Continue your AWS serverless learning

Learn at your own pace



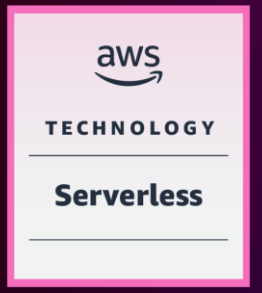
Expand your serverless skills with our learning plan on **AWS Skill Builder**

Increase your knowledge



Use our **Ramp-Up Guides** to build your serverless knowledge

Earn AWS Serverless badge



Demonstrate your knowledge by achieving **digital badges**

https://s12d.com/serverless-learning

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Josh Kahn
@joshuaakahn

Session resources
https://s12d.com/reinvent2022-svs302





Please complete the session survey in the **mobile app**

aws
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.