
AWS
re:Invent

SVS204

Building serverless applications with infrastructure as code

Eric Johnson

Senior Developer Advocate, Serverless

AWS

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Agenda

Introduction to infrastructure as code

Creating your first serverless application

Making the template reusable

Preparing for deployment

Deploying your application

Demo



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Download all resources in the presentation at https://s12d.com/svs204

Infrastructure as code (IaC)



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Infrastructure as code

- Automate the provisioning process
- Instantiate infrastructure using configuration files
- Treat configuration files as software code
- Eliminate configuration drift through automation

AWS IaC solutions

- AWS CloudFormation
- AWS Serverless Application Model (AWS SAM)
- AWS Cloud Development Kit (AWS CDK)

S A M



Serverless Application Model



AWS SAM comes in 2 parts



AWS SAM comes in 2 parts



AWS SAM templates

Using shorthand syntax to express resources and event source mappings, it provides IaC for serverless applications

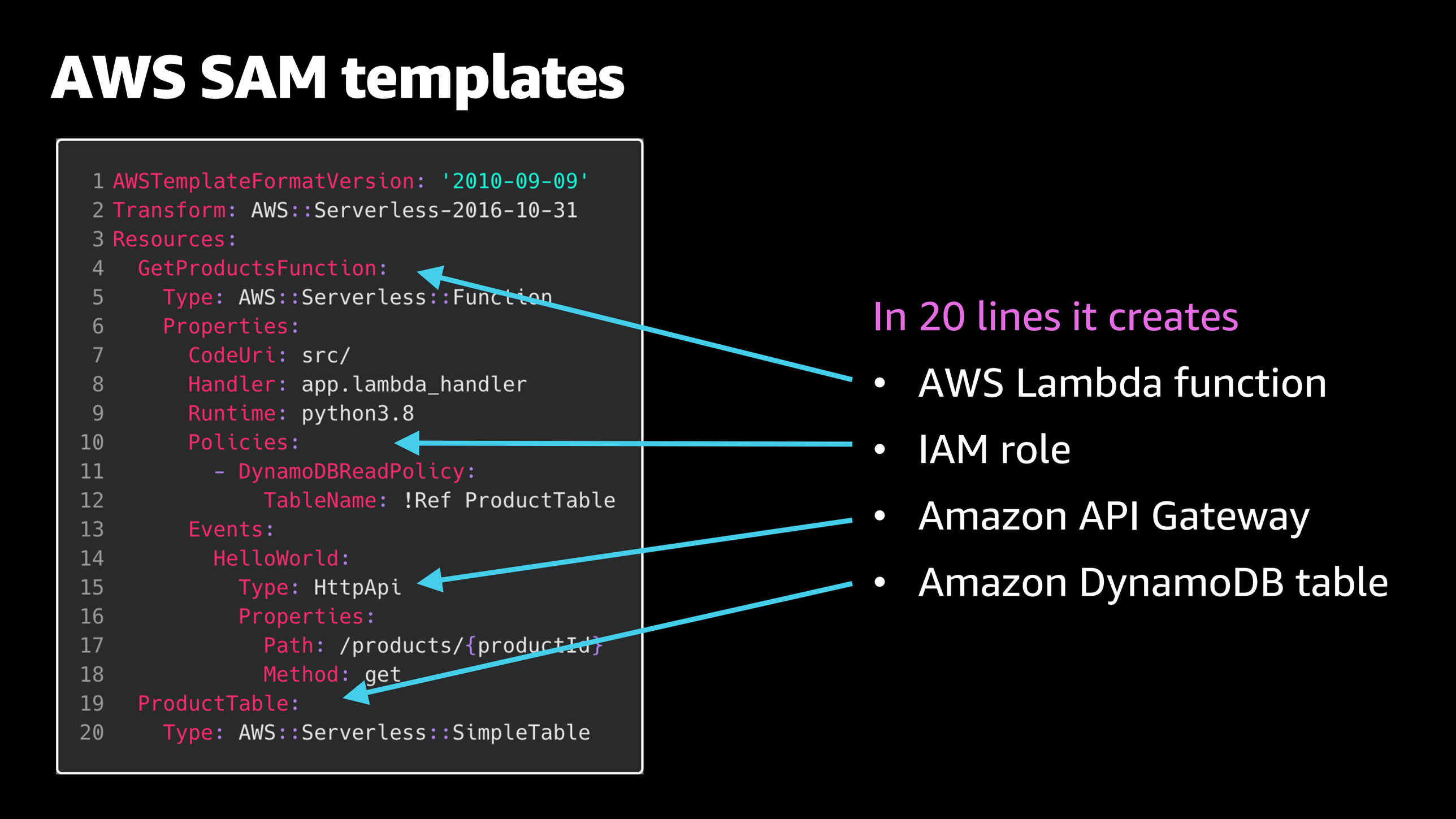
AWS SAM CLI

Provide tooling for local development, debugging, build, packaging, and deployment for serverless applications



AWS SAM templates

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Resources:
  GetProductsFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: src/
      Handler: app.lambda_handler
      Runtime: python3.8
      Policies:
        - DynamoDBReadPolicy:
            TableName: !Ref ProductTable
      Events:
        HelloWorld:
          Type: HttpApi
          Properties:
            Path: /products/{productId}
            Method: get
  ProductTable:
    Type: AWS::Serverless::SimpleTable
```



In 20 lines it creates

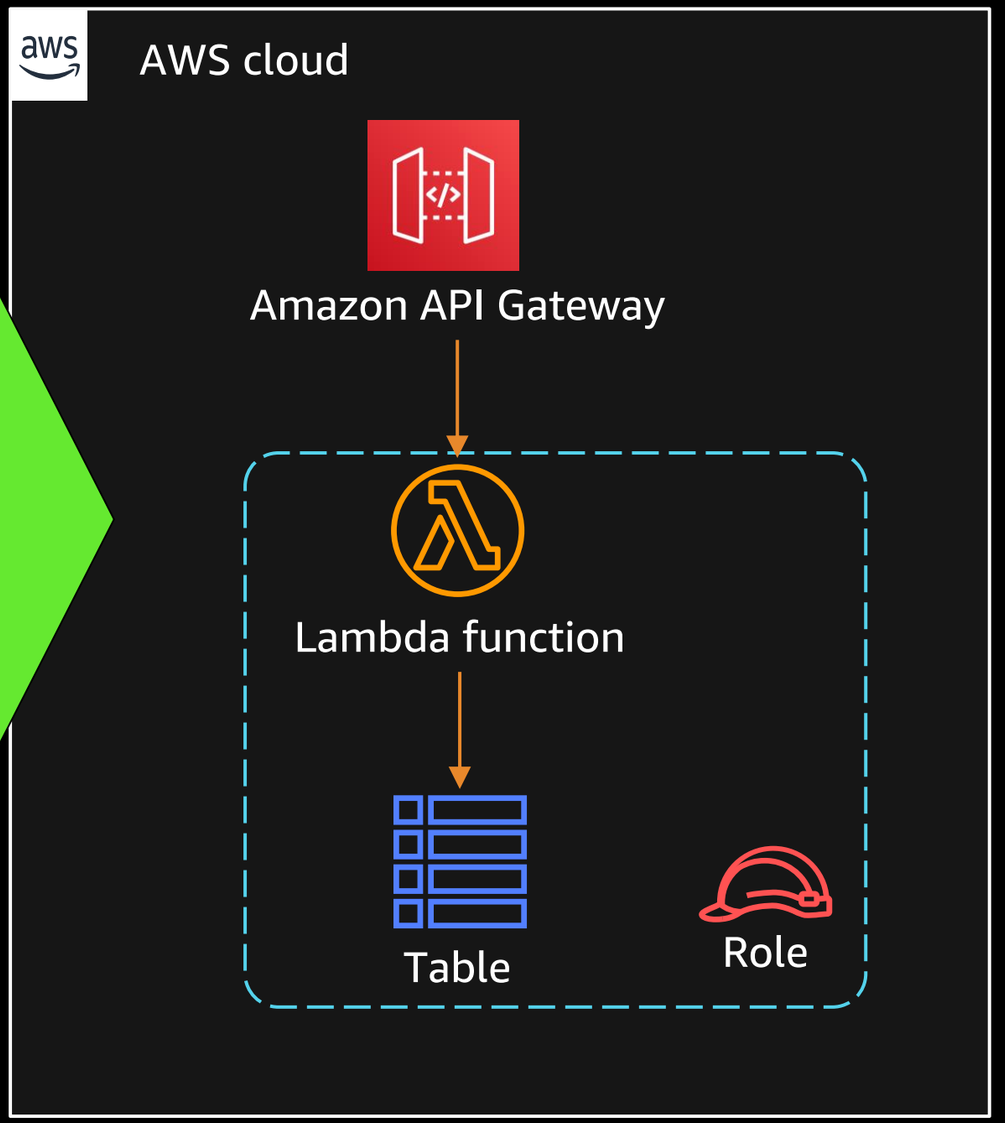
- AWS Lambda function
- IAM role
- Amazon API Gateway
- Amazon DynamoDB table

AWS SAM templates

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Resources:
  GetProductsFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: src/
      Handler: app.lambda_handler
      Runtime: python3.8
      Policies:
        - DynamoDBReadPolicy:
            TableName: !Ref ProductTable
      Events:
        HelloWorld:
          Type: HttpApi
          Properties:
            Path: /products/{productId}
            Method: get
  ProductTable:
    Type: AWS::Serverless::SimpleTable
```

Allowing ← this

To become this →



More on AWS SAM



https://s12d.com/master-sam

Creating your first serverless application



AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Installing AWS SAM



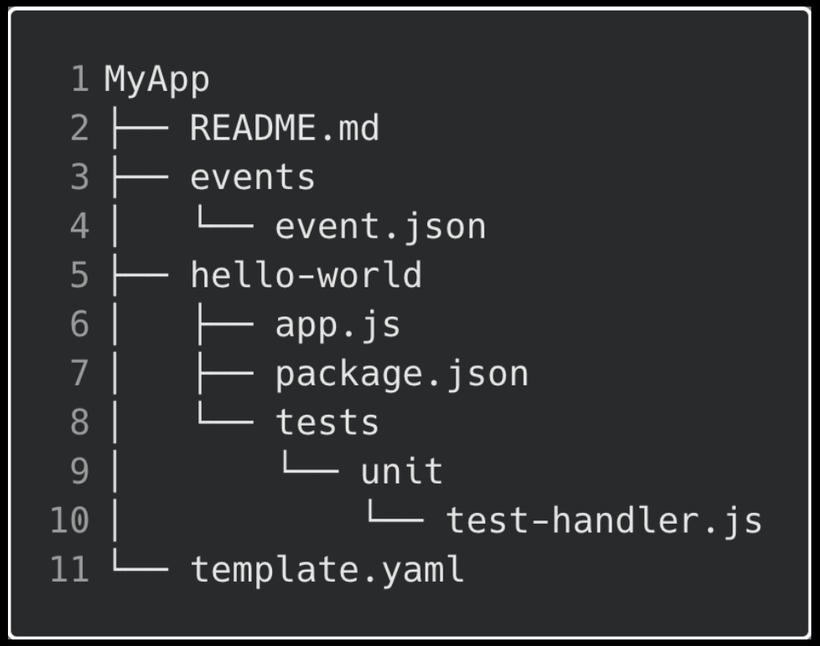
https://s12d.com/sam-install

AWS SAM init

Creates a serverless application based on user input

- Select an AWS managed template or a custom template of your own
- Select runtime
- Select dependency manager
- Name application

Results



More on AWS SAM init



https://s12d.com/sam-init-vid

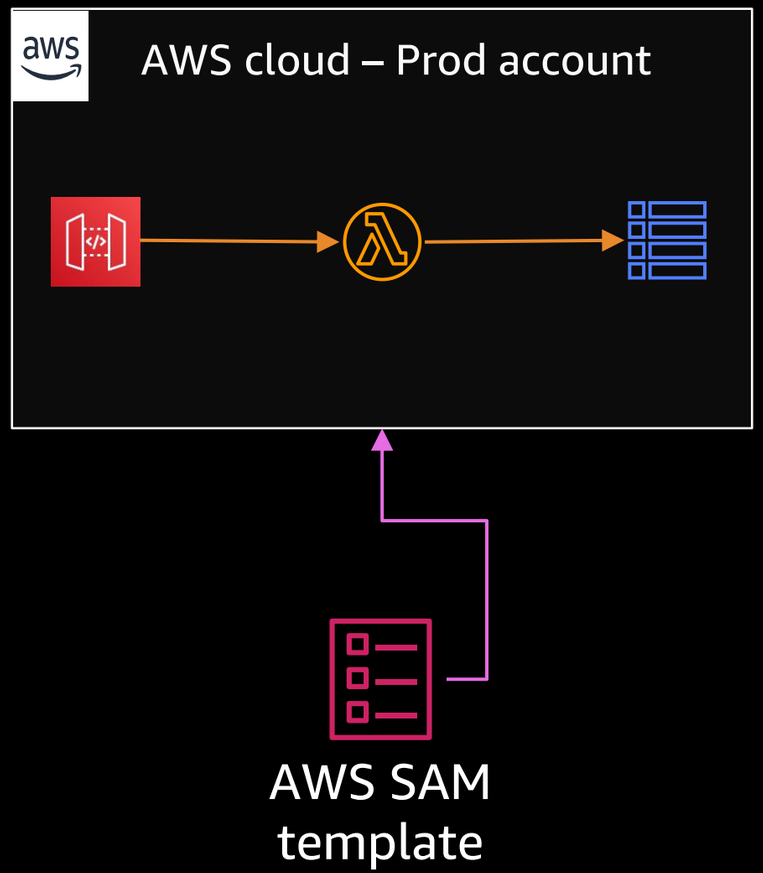
Making the template reusable



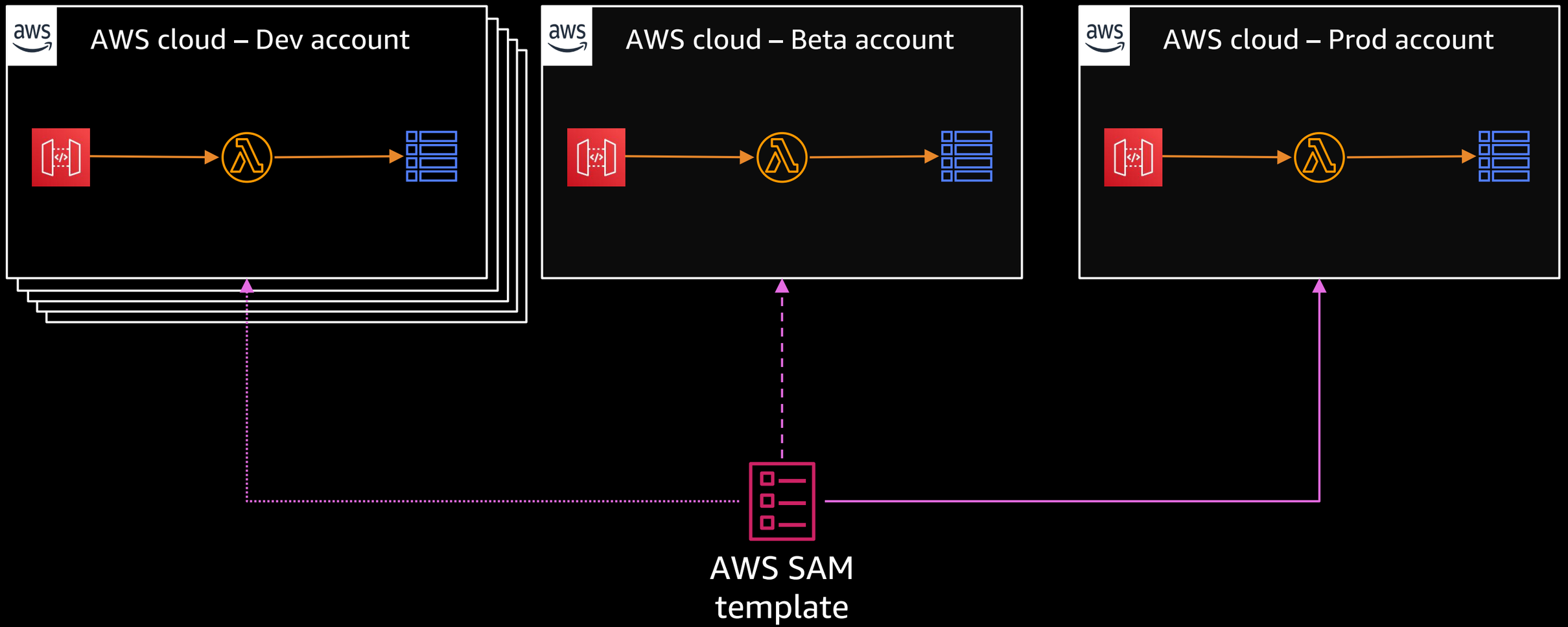
AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

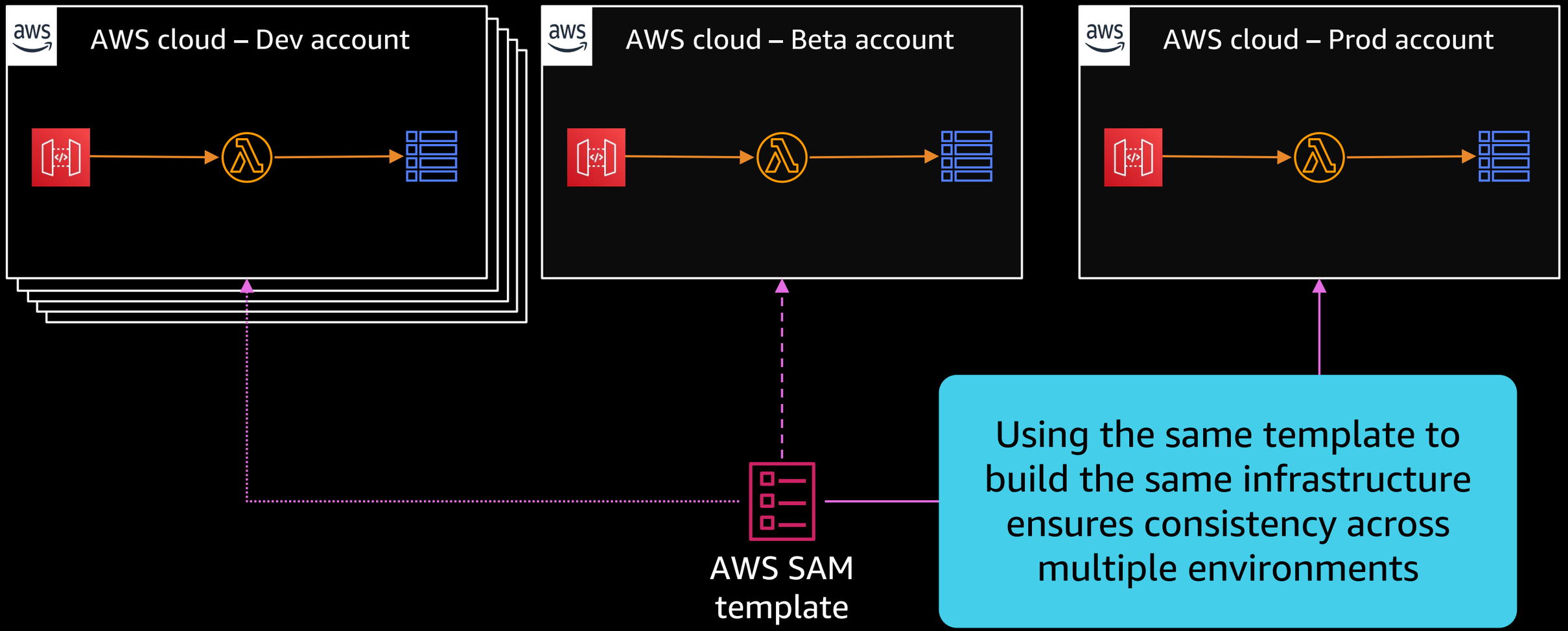
Why reusable templates?



Why reusable templates?



Why reusable templates?



Making a template reusable

USER DATA AND CONFIGURATION

Template parameters

- Added at creation or update
- Can be typed and validated
- Stored on deployment machine

Environment parameters

- Added at creation or update
- Stored in open text on console
- Dynamically referenced in template

AWS Systems Manager Parameter Store

- Added at creation, update, or runtime
- Compatible with some resources
- Encrypted values only at runtime

AWS Secrets Manager

- Added at creation, update, or runtime
- Compatible with most resources
- Open text or encrypted values

AWS AppConfig

- Added at runtime
- Works well with the new AWS Lambda Extensions

Making a template reusable

DYNAMIC REFERENCE

Intrinsic functions

- Fn::Base64
- Fn::Cidr
- Condition functions
  - Fn::And
  - Fn::Equals
  - Fn:If
  - Fn::Not
  - Fn::Or
- Fn::FindInMap
- Fn::GetAtt
- Fn::GetAZs
- Fn::ImportValue
- Fn::Join
- Fn::Select
- Fn::Split
- Fn::Sub
- Fn::Transform
- Ref

Pseudo parameters

- AWS::AccountId
- AWS::NotificationARNs
- AWS::NoValue
- AWS::Partition
- AWS::Region
- AWS::StackId
- AWS::StackName
- AWS::URLSuffix

More on parameters



https://s12d.com/secrets

More on AWS AppConfig



https://s12d.com/appconfig-ext

Preparing for deployment

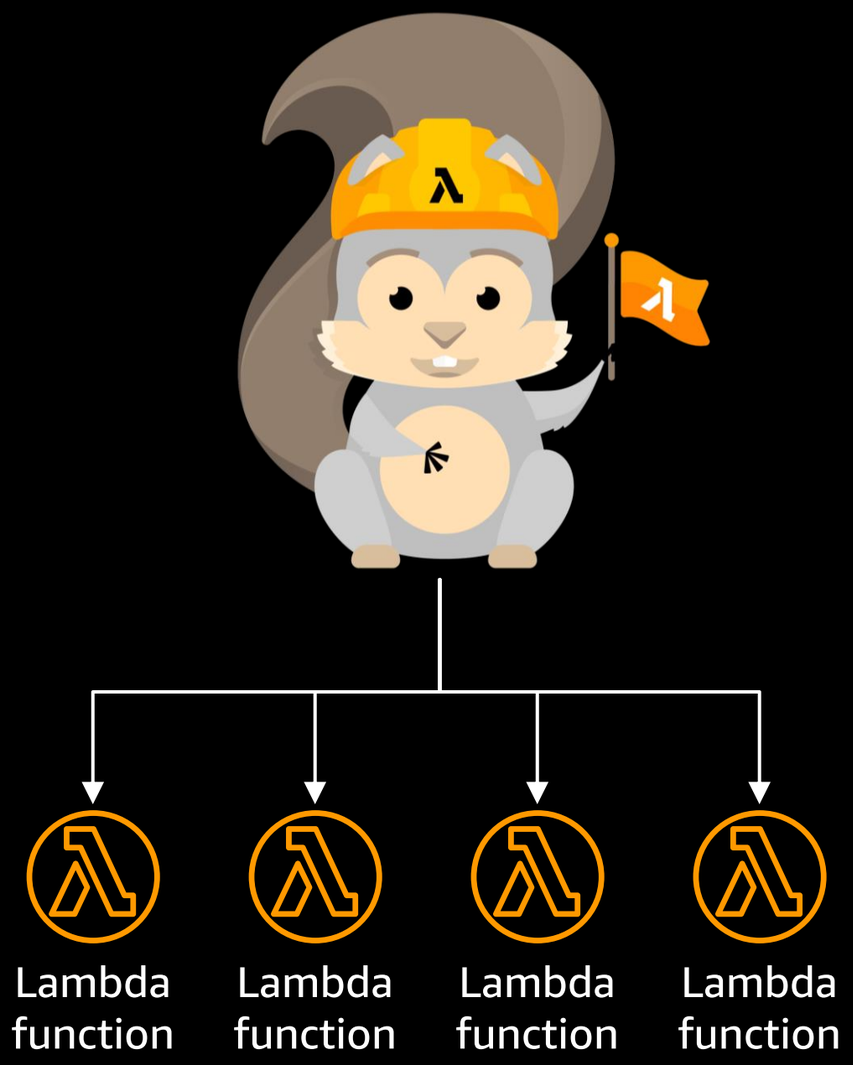


AWS re:Invent

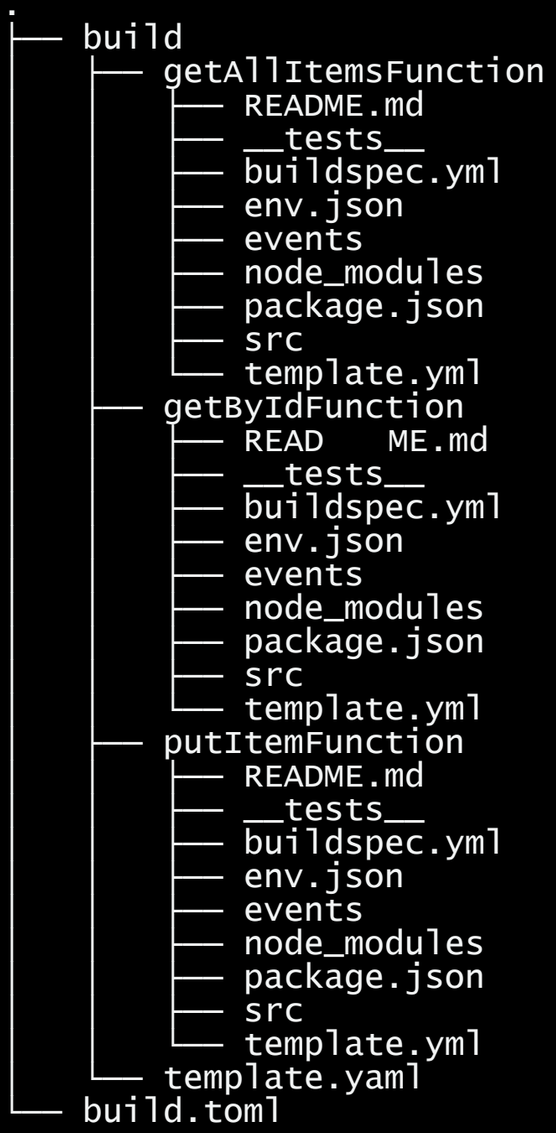
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS SAM build

- Generates deployment artifacts targeting Lambda's execution environment
- Options
  - Uses container to compile for native dependencies on Lambda execution environment
  - Caches builds for unchanged Lambda functions
  - Builds multiple functions in parallel
  - Utilizes native build tools like Dotnet, Go, and Maven for compiling
  - Allows for file omission in final artifacts
  - Uses Makefile for custom builds

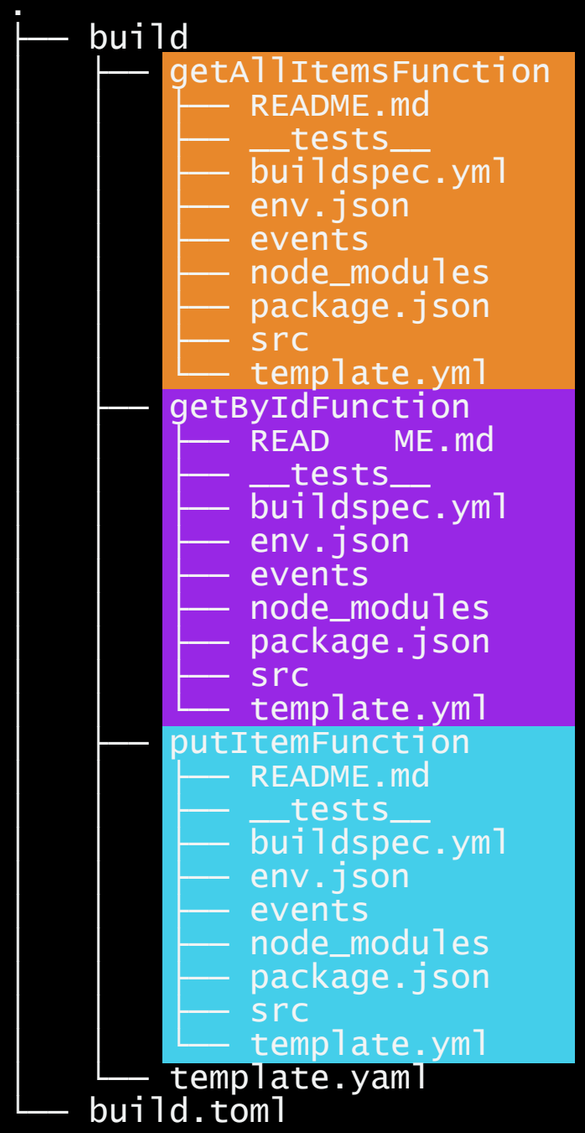


Build results



Each Lambda function is prepared as a self-contained package with code and dependencies

Build results



Each Lambda function is prepared as a self-contained package with code and dependencies

More on AWS SAM build



https://s12d.com/sam-build

Deploying your application



AWS re:Invent

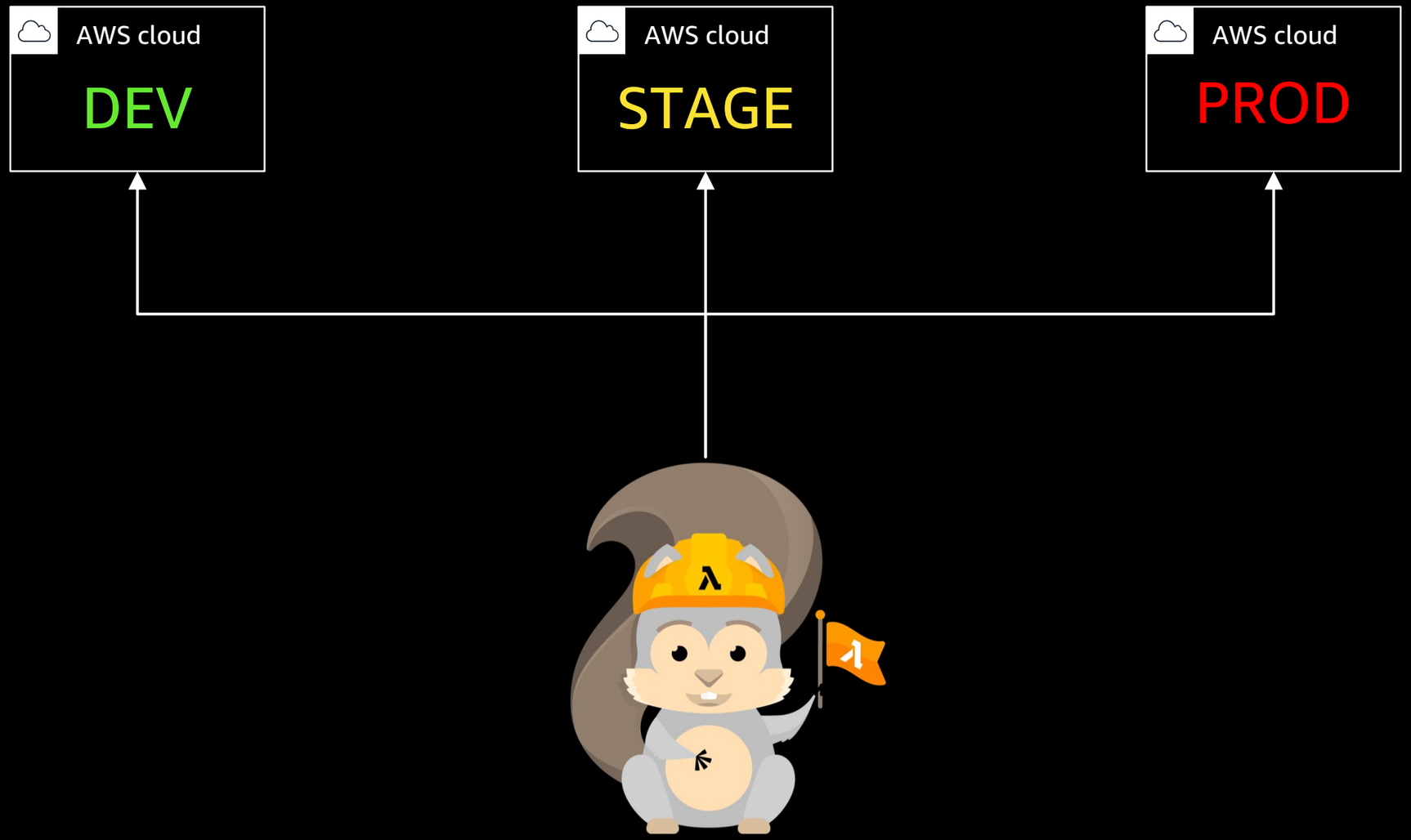
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS SAM deploy

- Deploys or updates serverless applications on AWS cloud
- Deploys and manages Amazon Simple Storage Service (Amazon S3) bucket for deployment artifacts
- Saves deploy options to a config file
- Guided deploy to save options
- Creates and validates AWS CloudFormation changeset
- Uses changeset for deployment



AWS SAM deploy to multiple environments



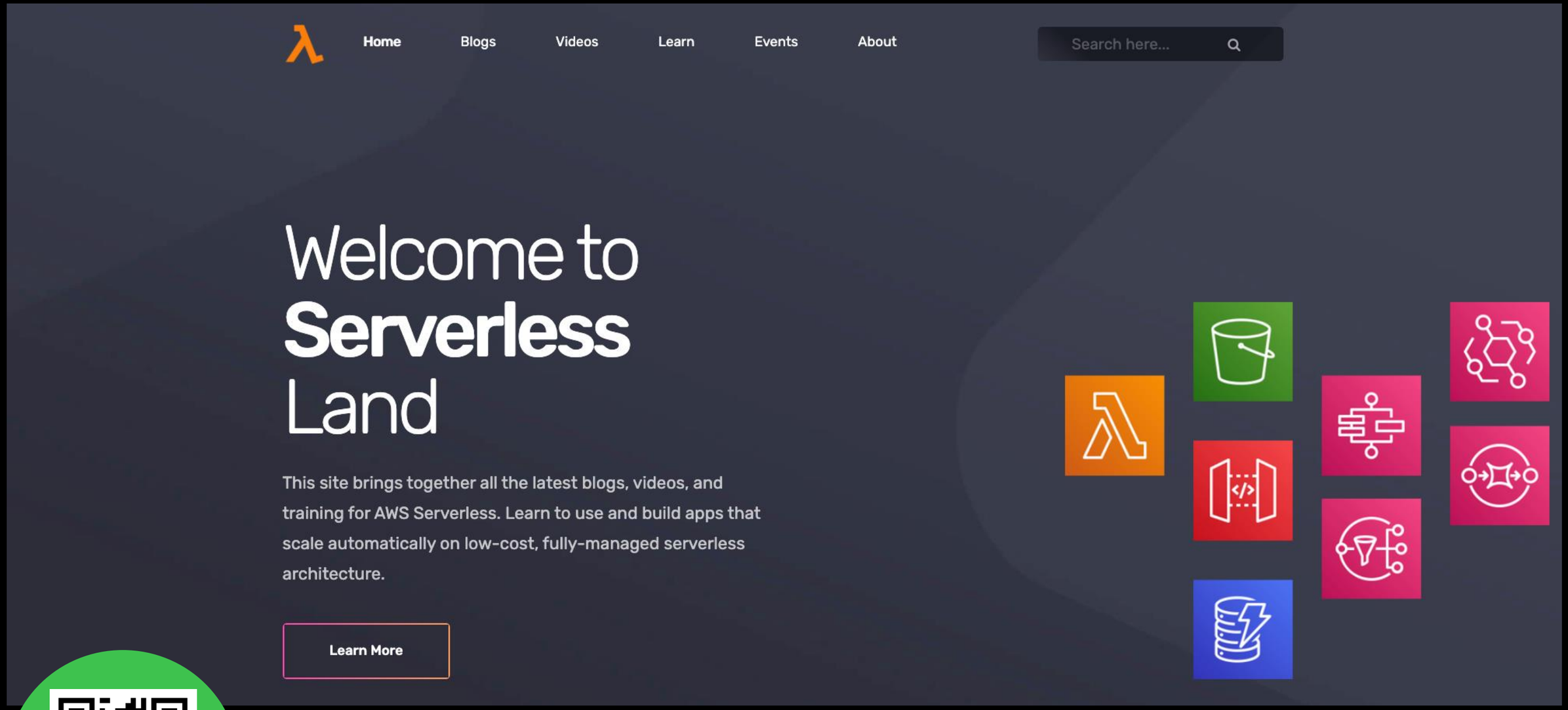
Let’s see some code!



© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Download all resources in the presentation at https://s12d.com/svs204



https://serverlessland.com

Thank you!

Eric Johnson

@edjgeek

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Please complete the session survey

AWS re:Invent

© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.