



Serverless Best Practices

Danilo Poccia

Principal Evangelist, Serverless

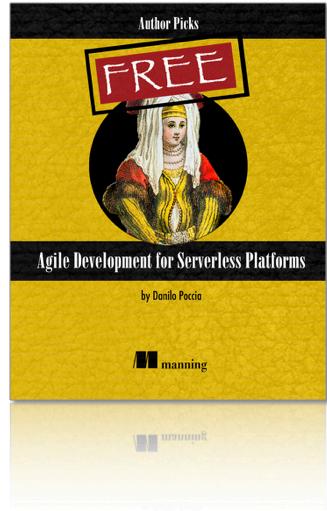
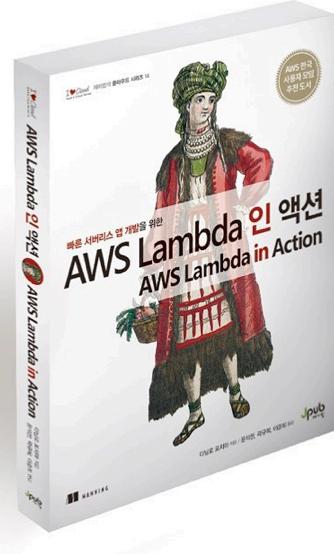
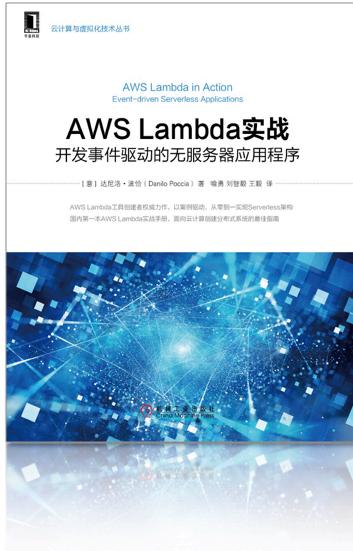
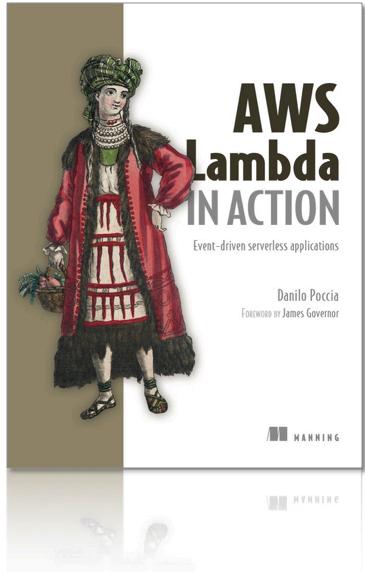
 danilop@amazon.com

 @danilop

 danilop



SSID: Guest
Password: Cube@11999



Managing Infrastructure as Code

Provision and manage a collection of related AWS resources.

Your application = CloudFormation stack

Input .yaml file and output provisioned AWS resources



**Meet
SAM!**

Serverless Application Model (SAM)

CloudFormation extension optimized for serverless

New serverless resource types: functions, APIs, and tables

Supports anything CloudFormation supports

Open specification (Apache 2.0)

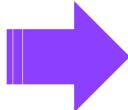
<https://github.com/awslabs/serverless-application-model>

SAM template

```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31 ←  
Resources:  
  GetHtmlFunction:  
    Type: AWS::Serverless::Function ←  
    Properties:  
      CodeUri: s3://demo-bucket/todo_list.zip  
      Handler: index.js  
      Runtime: nodejs6.1  
      Policies: AmazonDynamoDBReadOnlyAccess  
    Events:  
      GetHtml:  
        Type: Api  
        Properties:  
          Path: /{proxy+}  
          Method: ANY  
    } }
```

SAM template

```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31  
Resources:  
  GetHtmlFunction:  
    Type: AWS::Serverless::Function  
    Properties:  
      CodeUri: s3://demo-bucket/todo_list.zip  
      Handler: index.js  
      Runtime: nodejs6.1  
      Policies: AmazonDynamoDBReadOnlyAccess  
    Events:  
      GetHtml:  
        Type: Api  
        Properties:  
          Path: /{proxy+}  
          Method: ANY
```



AWS::Lambda::Function
AWS::IAM::Role
AWS::IAM::Policy

AWS::ApiGateway::RestApi
AWS::ApiGateway::Stage
AWS::ApiGateway::Deployment

AWS::Lambda::Permission

CloudFormation template

```
AWS::TemplateFormatVersion: '2010-09-09'
Resources:
  GetHtmlFunctionGetHtmlPermissionProd:
    Type: AWS::Lambda::Permission
    Properties:
      Action: lambda:invokeFunction
      Principal: apigateway.amazonaws.com
      FunctionName:
        Ref: GetHtmlFunction
      SourceArn:
        Fn::Sub: arn:aws:execute-api:${AWS::Region}:${AWS::AccountId}: ${ServerlessRestApi}/Prod/ANY/*
  ServerlessRestApiProdStage:
    Type: AWS::ApiGateway::Stage
    Properties:
      DeploymentId:
        Ref: ServerlessRestApiDeployment
      RestApiId:
        Ref: ServerlessRestApi
      StageName: Prod
  ListTable:
    Type: AWS::DynamoDB::Table
    Properties:
      ProvisionedThroughput:
        WriteCapacityUnits: 5
        ReadCapacityUnits: 5
      AttributeDefinitions:
        - AttributeName: id
          AttributeType: S
      KeySchema:
        -KeyType: HASH
          AttributeName: id
  GetHtmlFunction:
    Type: AWS::Lambda::Function
    Properties:
      Handler: index.gethtml
      Code:
        S3Bucket: flourish-demo-bucket
        S3Key: todo_list.zip
      Role:
        Fn::GetAtt:
          - GetHtmlFunctionRole
          - Arn
      Runtime: nodejs4.3
  GetHtmlFunctionRole:
    Type: AWS::IAM::Role
    Properties:
      ManagedPolicyArns:
        - arn:aws:iam::aws:policy/AmazonDynamoDBReadOnlyAccess
        - arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
      AssumeRolePolicyDocument:
        version: '2012-10-17'
        Statement:
          - Action:
              - sts:AssumeRole
            Effect: Allow
            Principal:
              Service:
                - lambda.amazonaws.com
  ServerlessRestApiDeployment:
    Type: AWS::ApiGateway::Deployment
    Properties:
      RestApiId:
        Ref: ServerlessRestApi
      Description: 'RestApi deployment id: 127e3fb91142ab1ddc5f5446adb094442581a90d'
      StageName: Stage
  GetHtmlFunctionGetHtmlPermissionTest:
    Type: AWS::Lambda::Permission
    Properties:
      Action: lambda:invokeFunction
      Principal: apigateway.amazonaws.com
      FunctionName:
        Ref: GetHtmlFunction
      SourceArn:
        Fn::Sub: arn:aws:execute-api:${AWS::Region}:${AWS::AccountId}: ${ServerlessRestApi}/*/ANY/*
  ServerlessRestApi:
    Type: AWS::ApiGateway::RestApi
    Properties:
      Body:
        info:
          version: '1.0'
          title:
            Ref: AWS::StackName
        paths:
          "/{proxy+}":
            x-amazon-apigateway-any-method:
              x-amazon-apigateway-integration:
                httpMethod: ANY
                type: aws_proxy
                uri:
                  Fn::Sub: arn:aws:apigateway:${AWS::Region}:lambda:path/2015-03-31/functions/${GetHtmlFunction.Arn}/invocations
```

Introducing the new SAM CLI



Usage: **sam** [OPTIONS] COMMAND [ARGS]...

AWS Serverless Application Model (SAM) CLI

The AWS Serverless Application Model extends AWS CloudFormation to provide a simplified way of defining the Amazon API Gateway APIs, AWS Lambda functions, and Amazon DynamoDB tables needed by your serverless application. You can find more in-depth guide about the SAM specification here: <https://github.com/awslabs/serverless-application-model>.

Options:

- debug** Turn on debug logging to print debug message generated by SAM CLI.
- version** Show the version and exit.
- help** Show this message and exit.

Commands:

- validate** Validate an AWS SAM template.
- init** Initialize a serverless application with a...
- package** Package an AWS SAM application. This is an alias for 'aws cloudformation package'.
- deploy** Deploy an AWS SAM application. This is an alias for 'aws cloudformation deploy'.
- logs** Fetch logs for a function
- local** Run your Serverless application locally for...

Introducing the new SAM CLI



Usage: **sam local [OPTIONS] COMMAND [ARGS]...**

Run your Serverless application locally for quick development & testing

Options:

--help Show this message and exit.

Commands:

- | | |
|-----------------------|--|
| generate-event | You can use this command to generate sample... |
| invoke | Invokes a local Lambda function once. |
| start-api | Sets up a local endpoint you can use to test your API.
Supports hot-reloading so you don't need to restart this service when you make changes to your function. |
| start-lambda | Starts a local endpoint you can use to invoke your local Lambda functions. |

Introducing the new SAM CLI



<https://github.com/awslabs/aws-sam-cli>

`pip install --user aws-sam-cli`

Introducing the new SAM CLI

```
sam init --runtime nodejs --name <NAME>
```

```
cd <NAME>/  
more README.md  
cd hello_world/  
more app.js  
npm install  
cd ..
```

```
sam validate
```

```
sam local start-api
```

```
sam package --template-file template.yaml \  
--s3-bucket <BUCKET> \  
--output-template-file packaged.yaml
```

```
sam deploy --template-file packaged.yaml \  
--stack-name <STACK> --capabilities CAPABILITY_IAM
```



New

Safe deployments baked into SAM!

Lambda aliases now enable traffic shifting

CodeDeploy integration for deployment automation

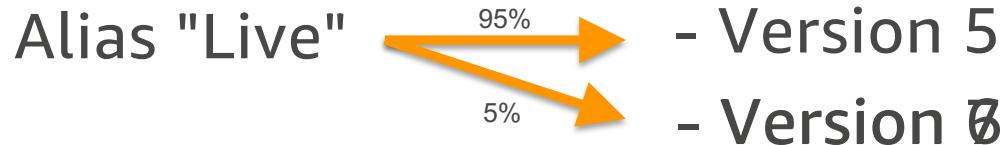
Deployment automation natively supported in SAM

New

Safe deployments baked into SAM!

Version – immutable deployment unit
Alias – pointer to a version

Lambda Function Foo:





New

Safe deployments baked into SAM!

- CodeDeploy integration
 - Preconfigured canary and linear deployments
 - Auto alarm-based rollbacks
 - Pre and post traffic validation hooks
 - Monitor through the CodeDeploy console
- Natively supported in SAM!



New

Safe deployments baked into SAM!

```
AWSTemplateFormatVersion: '2010-09-09'  
Transform: AWS::Serverless-2016-10-31  
Resources:  
  GetHtmlFunction:  
    Type: AWS::Serverless::Function  
    Properties:  
      CodeUri: s3://demo-bucket/todo_list.zip  
      Handler: index.js  
      Runtime: nodejs6.1
```



New

Safe deployments baked into SAM!

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Globals:
  Function:
    AutoPublishAlias: Live
    DeploymentPreference:
      Type: Canary10Percent10Minutes
Resources:
  GetHtmlFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: s3://demo-bucket/todo_list.zip
      Handler: index.js
      Runtime: nodejs6.1
    Policies: AmazonDynamoDBReadOnlyAccess
```



New

Safe deployments baked into SAM!

```
AWSTemplateFormatVersion: '2010-09-09'
Transform: AWS::Serverless-2016-10-31
Globals:
  Function:
    AutoPublishAlias: Live
    DeploymentPreference:
      Type: Canary10Percent10Minutes
    Hooks:
      PreTraffic: !Ref CodeDeployHook_PreTest
      PostTraffic: !Ref CodeDeployHook_PostTest
    Alarms:
      - !Ref DurationAlarm
      - !Ref ErrorAlarm
Resources:
  GetHtmlFunction:
    Type: AWS::Serverless::Function
    Properties:
      CodeUri: s3://demo-bucket/todo_list.zip
      Handler: index.js
      Runtime: nodejs6.1
      Policies: AmazonDynamoDBReadOnlyAccess
```

Code Deploy console

The screenshot shows the AWS CodeDeploy console for a deployment named "d-4MZ1SZJWP". The deployment status is "Deployment In progress".

Deployment status:

- Step 1: Pre-deployment validation - Completed
- Step 2: Traffic shifting - 40% complete
- Step 3: Post-deployment validation - Not started

Traffic shifting status:

The deployment will shift 10% of traffic from the current version to the replacement version every 1 minute(s) until all of the traffic is routed to the new version. [Learn more](#)

Original version	Replacement version
60%	40%

60% of traffic 40% of traffic

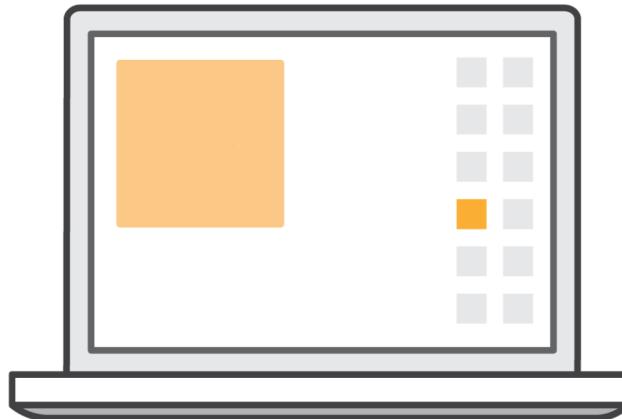
Deployment details: ▶

Events:

Event	Start time	End time	Status
BeforeAllowTraffic	Nov 20, 2017 5:57:28 PM UTC	Nov 20, 2017 5:57:29 PM UTC	Succeeded
AllowTraffic	Nov 20, 2017 5:57:29 PM UTC		In progress
AfterAllowTraffic			Pending

Demo:

AWS SAM CLI



Keep you updated on AWS SAM

1. <https://github.com/awslabs/serverless-application-model/releases/tag/v1.6.1>
2. <https://github.com/awslabs/serverless-application-model/releases/tag/v1.7.0>



Serverless Best Practices

Danilo Poccia

Principal Evangelist, Serverless

 danilop@amazon.com

 @danilop

 danilop



SSID: Guest
Password: Cube@11999