
AWS
re:Invent

SVS306

I didn’t know AWS SAM could do that!

Alex Wood

Senior Software Development Engineer

AWS

AWS re:Invent

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

I didn’t know AWS SAM could do that!

- Customized builds
- AWS SAM CLI configuration
- Custom local testing events
- Log analysis



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

What are AWS SAM and the AWS SAM CLI?

- AWS Serverless Application Model (AWS SAM)
  - AWS CloudFormation transform
  - Generally available
  - Open source: https://github.com/aws/serverless-application-model
- AWS SAM command line interface (CLI)
  - Command line tool
  - Generally available
  - Open source: https://github.com/aws/aws-sam-cli

What are AWS SAM and the AWS SAM CLI?

- AWS SAM and AWS SAM CLI provide an integrated experience
- AWS SAM provides useful abstractions over serverless resources
- AWS SAM CLI helps you create, build, test, and deploy applications
- Provide templates and builders for all native AWS Lambda runtimes

Customized builds



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

AWS SAM CLI build command

- The `sam build` command provides best practice build processes
  - Built In – JavaScript, Python, Java, Ruby, Go, .NET
- Convenient for building in containers, and packaging for AWS Lambda

AWS SAM CLI build command

- The `sam build` command provides best practice build processes
  - Built In – JavaScript, Python, Java, Ruby, Go, .NET
- Convenient for building in containers, and packaging for AWS Lambda

Open question – What if I know a better way to build my functions?

Custom build with Makefiles: Go Example

```
CustomBuildGoFunction:
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: src/
    Handler: bootstrap
    Runtime: provided.al2
    Events:
      WebRoute:
        Type: Api
        Properties:
          Path: /
          Method: GET
  Metadata:
    BuildMethod: makefile
```

```
build-CustomBuildGoFunction:
	GOOS=linux go build
	cp src $(ARTIFACTS_DIR)/bootstrap
```

Custom build with Makefiles: Layers

```
MyLayer:
  Type: AWS::Serverless::LayerVersion
  Properties:
    ContentUri: my_layer
    CompatibleRuntimes:
      - python3.8
  Metadata:
    BuildMethod: makefile
```

```
build-MyLayer:
  mkdir -p "$(ARTIFACTS_DIR)/python”
  cp *.py "$(ARTIFACTS_DIR)/python”
  python -m pip install \
    -r requirements.txt \
    -t "$(ARTIFACTS_DIR)/python"
```

AWS SAM CLI custom builds

- AWS SAM CLI provides helpful, built-in build instructions for Lambda runtimes
- For custom runtimes, layers, and special build instructions, you can provide your own Makefile instead of the built-in build steps
- Sessions with SAM – https://www.youtube.com/watch?v=wpccutnSbAk

AWS SAM CLI configuration



AWS re:Invent

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

AWS SAM CLI configuration files

- With `sam deploy --guided` you can save arguments for packaging and deploying AWS SAM applications
- These configuration files act as “default arguments” for the parameters used in the `sam deploy` command

AWS SAM CLI configuration files

- With `sam deploy --guided` you can save arguments for packaging and deploying AWS SAM applications
- These configuration files act as “default arguments” for the parameters used in the `sam deploy` command

What can we do with this?

samconfig.toml simplifies deployment

```
version = 0.1
[default]
[default.deploy]
[default.deploy.parameters]
stack_name = "my-sam-stack"
region = "us-east-2"
# etc.
```

Why not other commands?

```
version = 0.1
[default]
[default.deploy]
[default.deploy.parameters]
stack_name = "my-sam-stack"
region = "us-east-2"
# etc.
```

```
version = 0.1
[default]
[default.build]
[default.build.parameters]
use_container = true
```

Why not multiple profiles?

```
version = 0.1
[beta]
[beta.deploy]
[beta.deploy.parameters]
stack_name = "sam-app-beta"
s3_prefix = "sam-app-beta"
region = "us-east-1"
capabilities = "CAPABILITY_IAM"
```

```
version = 0.1
[prod]
[prod.deploy]
[prod.deploy.parameters]
stack_name = "sam-app"
s3_prefix = "sam-app"
region = "us-east-1"
capabilities = "CAPABILITY_IAM"
```

What can you do with AWS SAM CLI config?

- With guided deployments, you can save deployment configuration for re-use
- You can use multiple configuration files and/or multiple configuration profiles, e.g., for beta vs. prod stacks
- You can use configuration files to provide default arguments for other AWS SAM CLI commands, like build and package

Custom local testing events



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

AWS SAM CLI local testing

- Three testing modes – invoke, start-api, start-lambda
  - E.g., Start-api is helpful for testing Amazon API Gateway web applications
  - E.g., Invoke is helpful for testing async Lambda functions

AWS SAM CLI local testing

- Three testing modes – invoke, start-api, start-lambda
  - E.g., Start-api is helpful for testing Amazon API Gateway web applications
  - E.g., Invoke is helpful for testing async Lambda functions

How do we effectively test async events?

Example: sam local generate-event

```
sam local generate-event sqs receive-message --body "{\\\"json_message\\\":true}"
{
  "Records": [
    {
      "messageId": "19dd0b57-b21e-4ac1-bd88-01bbb068cb78",
      "receiptHandle": "MessageReceiptHandle",
      "body": "{"json_message":true}",
      "attributes": {
        "ApproximateReceiveCount": "1",
        "SentTimestamp": "1523232000000",
        "SenderId": "123456789012",
        "ApproximateFirstReceiveTimestamp": "1523232000001"
      },
      "messageAttributes": {},
      "md5OfBody": "1219c0cc803d1aa0189869e07e60d89c",
      "eventSource": "aws:sqs",
      "eventSourceARN": "arn:aws:sqs:us-east-1:123456789012:MyQueue",
      "awsRegion": "us-east-1"
    }
  ]
}
```

AWS SAM CLI local testing

- Useful for fast iterations in container environments similar to Lambda
- The `generate-event` command is helpful for testing specific service integrations

Log analysis



AWS re:Invent

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

Log analysis

- Lambda functions output their log events and execution metadata to Amazon CloudWatch Logs
- Many use the AWS Management Console to diagnose failures and issues from log statements

Log analysis

- Lambda functions output their log events and execution metadata to Amazon CloudWatch Logs
- Many use the AWS Management Console to diagnose failures and issues from log statements

How can we do this from the command line?

sam logs

```
# Fetch recent logs
sam logs -n MyServerlessFunction --stack-name sam-app
```

sam logs

```
# Fetch recent logs
sam logs -n MyServerlessFunction --stack-name sam-app

# Tail logs, live!
sam logs -n MyServerlessFunction --stack-name sam-app --tail
```

sam logs

```
# Fetch recent logs
sam logs -n MyServerlessFunction --stack-name sam-app

# Tail logs, live!
sam logs -n MyServerlessFunction --stack-name sam-app --tail

# View logs in a time range
sam logs -n MyServerlessFunction --stack-name sam-app -s '10min ago' -e '2min ago'
```

sam logs

```
# Fetch recent logs
sam logs -n MyServerlessFunction --stack-name sam-app

# Tail logs, live!
sam logs -n MyServerlessFunction --stack-name sam-app --tail

# View logs in a time range
sam logs -n MyServerlessFunction --stack-name sam-app -s '10min ago' -e '2min ago'

# Filter logs
sam logs -n MyServerlessFunction --stack-name sam-app --filter "ERROR"
```

“Use samconfig files to simplify repeated command arguments.”

Me

Several Minutes Ago



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

Wrap-up



AWS re:Invent

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

Wrap-up

- AWS SAM and the AWS SAM CLI provide an integrated development experience
- AWS SAM CLI provides built-in steps to build Lambda functions, but you can bring your own build process if you need it
- AWS SAM CLI configuration files simplify your common command patterns, and you can split across files and profiles for different use cases
- AWS SAM CLI provides local testing and can help simulate service integrations with generated events
- AWS SAM CLI can help you with log diving to diagnose issues in production

What's next?

- Go build! All of these examples are runnable in just a few minutes
- Read the developer guide –
  - https://docs.aws.amazon.com/serverless-application-model/index.html
- Watch Sessions with SAM –
  - https://www.youtube.com/playlist?list=PLJo-rJlep0ED198FJnTzhIB5Aut_1vDAd
- Reach out with questions, feature requests, or just to share what you build

Thank you!

Alex Wood

Senior Software Development Engineer

Twitter @alexwwood

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.