AWS re:Invent

NOV. 29 – DEC. 3, 2021 | LAS VEGAS, NV

ARC313-R2

Operational excellence with serverless workloads

Joe Mann
Sr. Partner Solutions Architect, Red Hat
AWS

Brian Zambrano
Enterprise Solutions Architect
AWS

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Session agenda

- AWS Well-Architected Framework overview
- Serverless Applications Lens overview
- Interactive serverless scenario walkthrough

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What is the Well-Architected Framework?



Pillars



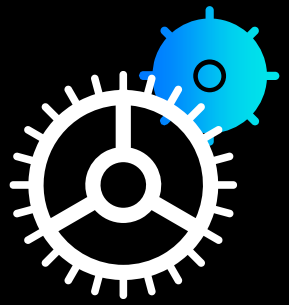
Design principles



Questions

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Operational excellence



How your organization supports your business objectives, your ability to run workloads effectively, gain insight into their operations, and to continuously improve supporting processes and procedures to deliver business value

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS Well-Architected Lenses



Management and Governance Lens



Serverless Applications Lens



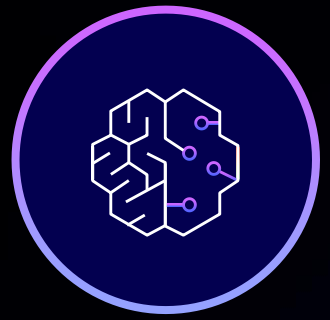
Data Analytics Lens



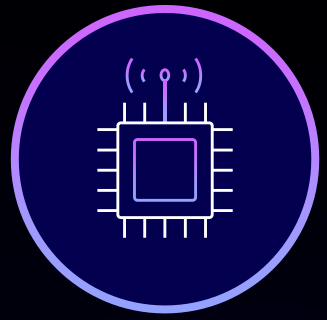
IoT Lens



SaaS Lens



Machine Learning Lens



HPC Lens



Financial Services Industry Lens

aws

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Logging

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Structured logging

- The consistent use of structured logging is critical to quickly and easily find problems and understand workload behavior
- Use Lambda Powertools for supported languages: Python, Java, TypeScript

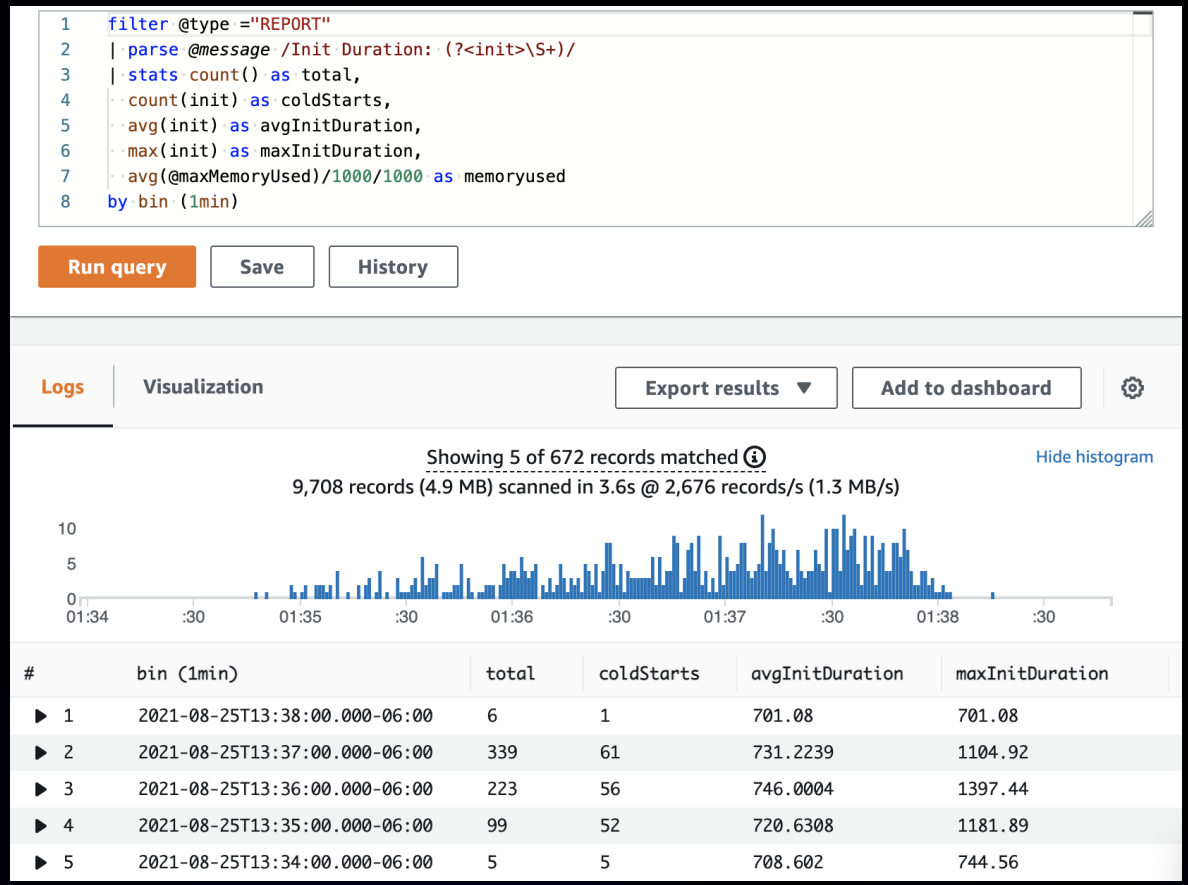
```
from aws_lambda_powertools import Logger

logger = Logger()


@logger.inject_lambda_context
def handler(event, context):
    records = event['Records']

    logger.info({
      'EVENT_TYPE': 'RECEIVE',
      'message': f'Got {len(records)} records',
    })

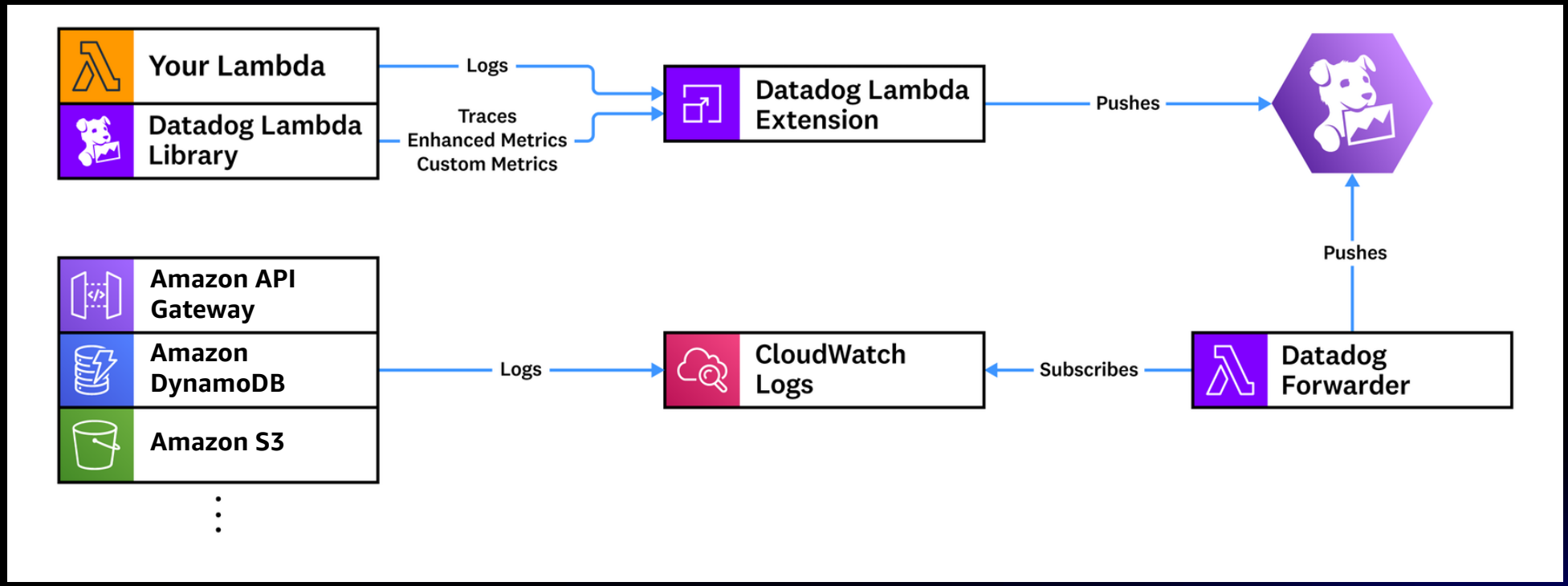
    for r in records:
        payload = r["body"]
        logger.info({'EVENT_TYPE': 'PROCESS', 'payload': str(payload)})
```



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Structured logging

- Use Amazon CloudWatch Logs Insights to find data
- Use the Lambda Logs API and corresponding Lambda layer to send logs to third-party providers



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Metrics/tracing

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon CloudWatch metrics

- **AWS Lambda:** Duration, Errors, Throttles, and ConcurrentExecutions; for stream-based invocations, alert on IteratorAge; for asynchronous invocations, alert on DeadLetterErrors; when provisioned concurrency is enabled, use ProvisionedConcurrencySpilloverInvocations
- **Amazon API Gateway:** IntegrationLatency, Latency, and 5XXError; for WebSocket API, use ClientError, IntegrationError, and ExecutionError
- **Application Load Balancer:** HTTPCode_ELB_5XX_Count, RejectedConnectionCount, HTTPCode_Target_5XX_Count, UnHealthyHostCount, LambdaInternalError, and LambdaUserError
- **AWS AppSync:** 5XX and Latency
- **Amazon SQS:** ApproximateAgeOfOldestMessage
- **Amazon Kinesis Data Streams:** ReadProvisionedThroughputExceeded, WriteProvisionedThroughputExceeded, GetRecords.IteratorAgeMilliseconds, PutRecord.Success, PutRecords.Success (if using Kinesis Producer Library), and GetRecords.Success
- **Amazon SNS:** NumberOfNotificationsFailed and NumberOfNotificationsFilteredOut-InvalidAttributes
- **AWS Step Functions:** ExecutionThrottled, ExecutionsFailed, ExecutionsTimedOut, ActivitiesTimedOut, and LambdaFunctionsTimedOut
- **Amazon EventBridge:** FailedInvocations and ThrottledRules

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Custom Amazon CloudWatch metrics

- Tools
  - Lambda Powertools
  - CloudWatch Embedded Metrics Format (EMF)
- Synthesize metrics from Amazon CloudWatch Logs Insights

```
from aws_lambda_powertools import Logger, Metrics
from aws_lambda_powertools.metrics import MetricUnit


logger = Logger()
metrics = Metrics()

def handler(event, context):
    records = event['Records’]

    metrics.add_metric(name="SuccessfulReceipt", unit=MetricUnit.Count, value=1)

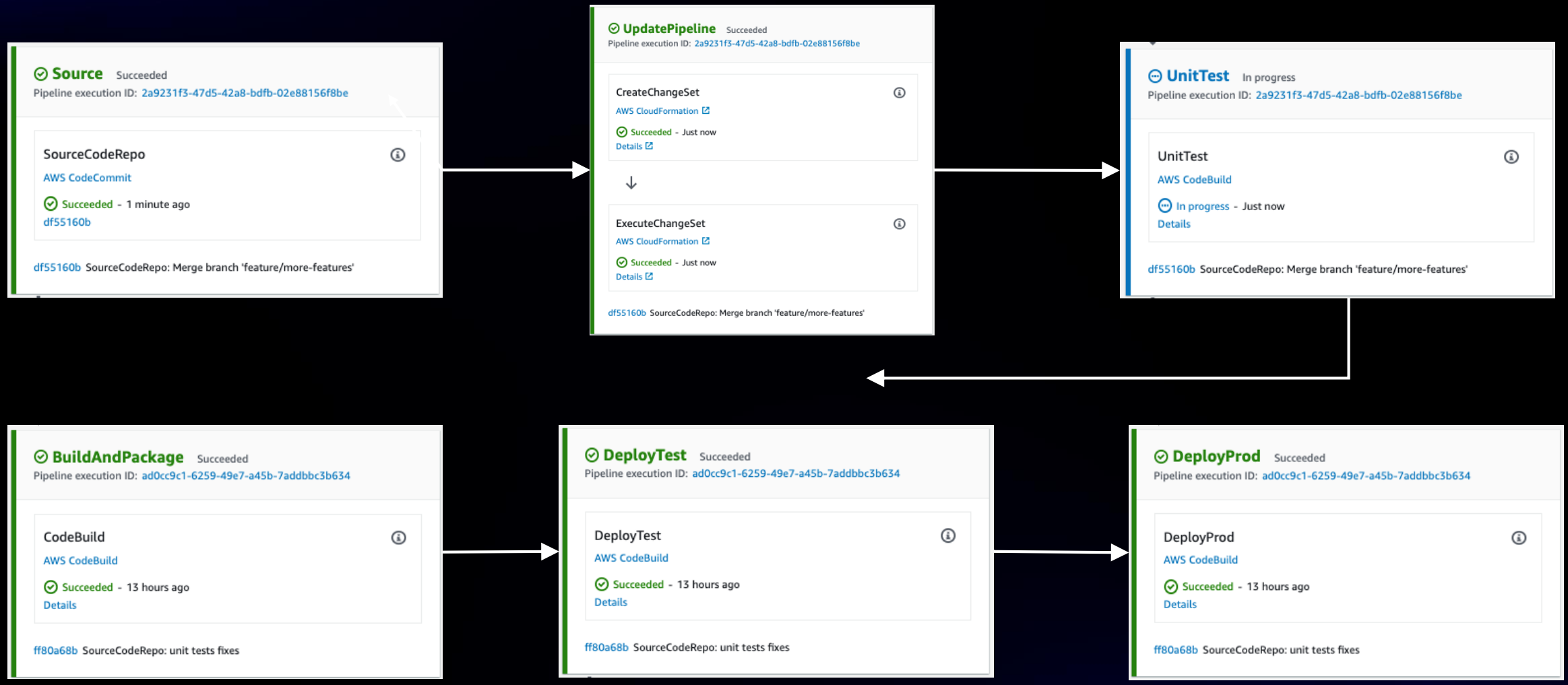
    logger.info({'EVENT_TYPE': 'RECEIVE', 'message': f’Got {len(records)} records’})
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployments

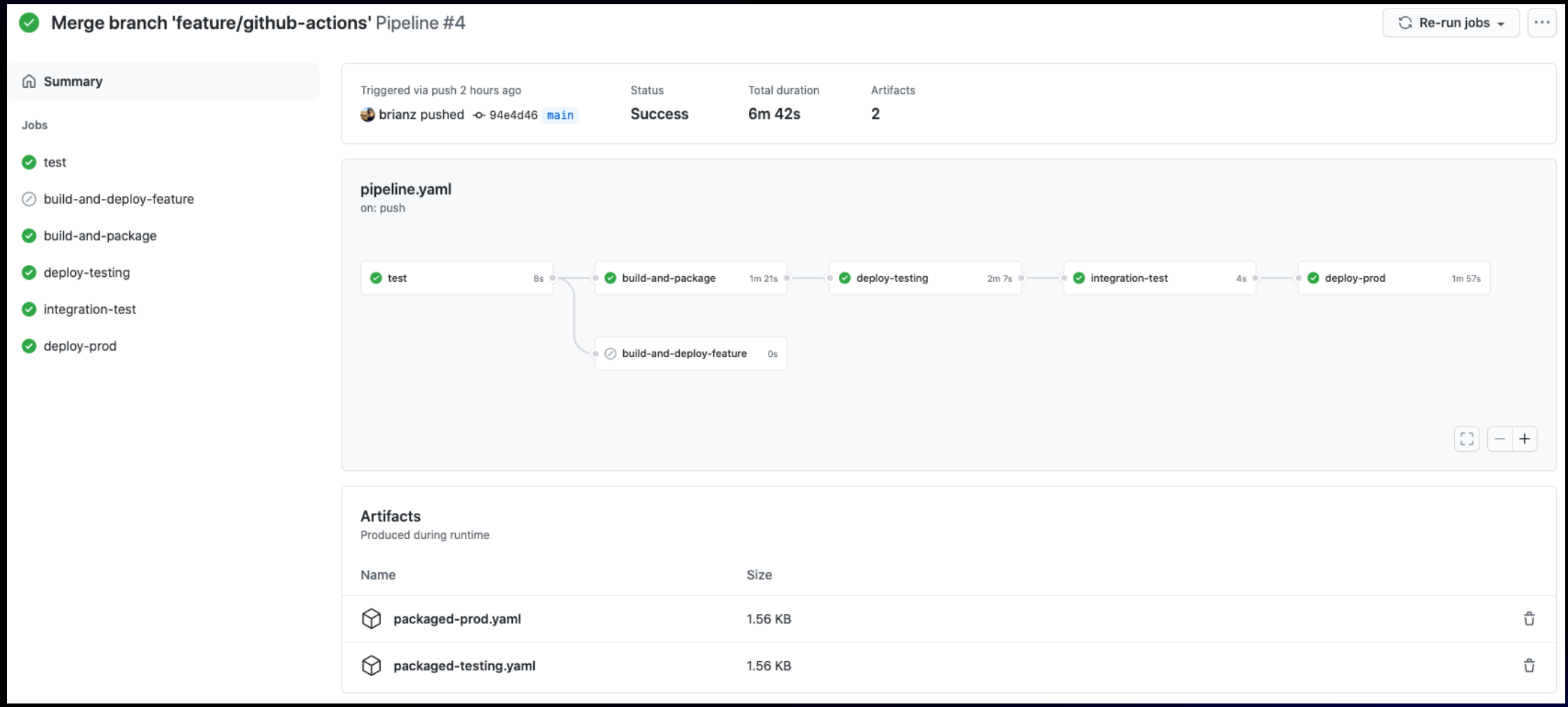
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS SAM Pipelines – AWS CodePipeline



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS SAM Pipelines – GitHub actions



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment capabilities

- Built-in integration with AWS CodeDeploy
- Uses function aliases to traffic shift
- Use Amazon CloudWatch alarms for monitoring deployment
- Use pre-/post-traffic hooks for validation, bootstrapping, etc.

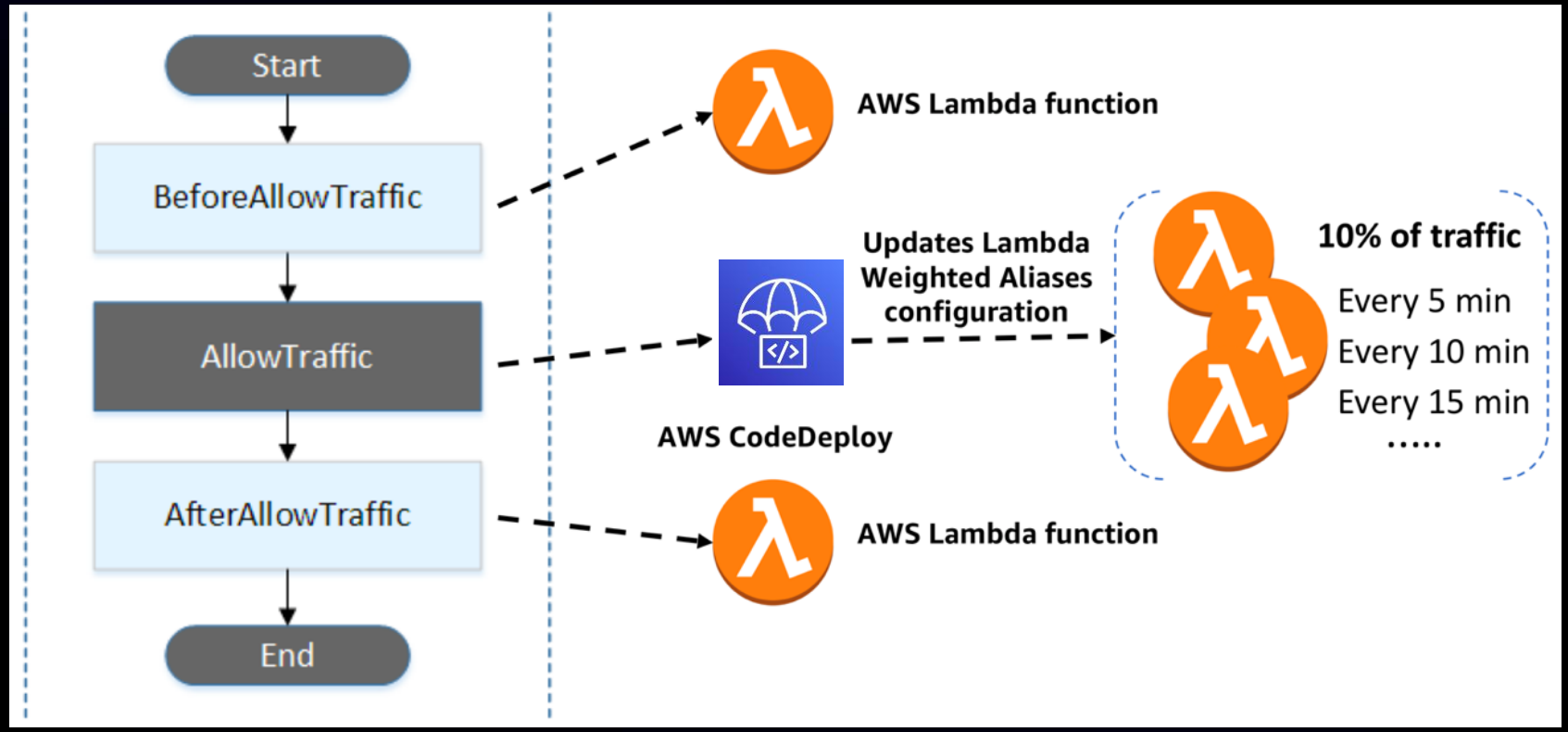
```
HelloWorldFunction:
  Type: AWS::Serverless::Function
  Properties:
    CodeUri: hello_world/
    Handler: app.hello_handler
    AutoPublishAlias: live

    Events:
      HelloEndpoint:
        Type: HttpApi
        Properties:
          Path: /hello
          Method: GET

    DeploymentPreference:
      Type: Canary10Percent5Minutes
      TriggerConfigurations:
        - TriggerTargetArn: !Ref AlertTopic
      Alarms:
       - !Ref AliasErrorGreaterThanZeroAlarm
       - !Ref
LatestVersionErrorGreaterThanZeroAlarm
     Hooks: # Validation functions
       PreTraffic: !Ref PreTrafficLambdaFunction
       PostTraffic: !Ref PostTrafficLambdaFunction
```

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment example



© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Deployment preference types

- AllAtOnce
  - Moves all traffic at once
- Canary
  - Shifts traffic in two batches: canary, everything else
  - Canary10Percent5Minutes . . . Canary10Percent30Minutes
- Linear
  - Shifts traffic in equal increments
  - Linear10PercentEvery1Minute, Every2Minutes, Every3Minutes, Every10Minutes

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Thank you!

Joe Mann
ajmann@amazon.com

Brian Zambrano
zambb@amazon.com

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.