
AWS
re:Invent

DOP302

Canaries in the code mines: Monitoring deployment pipelines

Lucas Schejtman

Principal Solutions Architect

AWS

AWS re:Invent

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

AWS observability services

Services / features



Synthetics



Contributor Insights



Container Insights



Logs Insights



Composite alarms



Anomaly detection



Amazon CloudWatch ServiceLens

Foundation



Amazon CloudWatch metrics



Amazon CloudWatch Logs



AWS X-Ray traces



Amazon CloudWatch alarms

What is Amazon CloudWatch Synthetics?



Website and API endpoint monitoring

Availability monitoring
Latency and performance monitoring



Outside in monitoring

Client side to transaction monitoring



Continuous monitoring

Acts like a user
24/7 continuously running health checks

Why do you need Synthetics monitoring?

To ensure a consistently great customer experience

Observe	See what your customer sees Spot intermittent issues See when issues were introduced
Inspect	Debug on issues Discover performance issues Distributed applications means more dependencies
Respond	Reduce mean time to recovery Reduce customer impact Fix issues / initiate rollbacks Find root cause post recovery

AWS “code” services

Software release steps:



AWS
CodeStar

Source → Build → Test → Deploy



AWS CodeCommit



AWS CodeBuild



AWS CodeBuild +
third party



AWS CodeDeploy



AWS CodePipeline

What is CodePipeline?



Workflow modeling

Model and visualize your software release process



AWS integrations

Natively integrate with AWS products and services



Built-in and custom plugins

One-click plugin integration for source, build, test, and deploy

re:invent book engine

Your place for everything books

Go ahead and search for something

enter your search criteria... Search

Author Title ISBN Year

What I showed you

1. Sample website overview
2. How to configure a CloudWatch Synthetics canary
3. How to configure an AWS CodePipeline pipeline

What is AWS Step Functions?



Fully-managed state machine

Resilient workflow automation and built-in error handling



Integrations

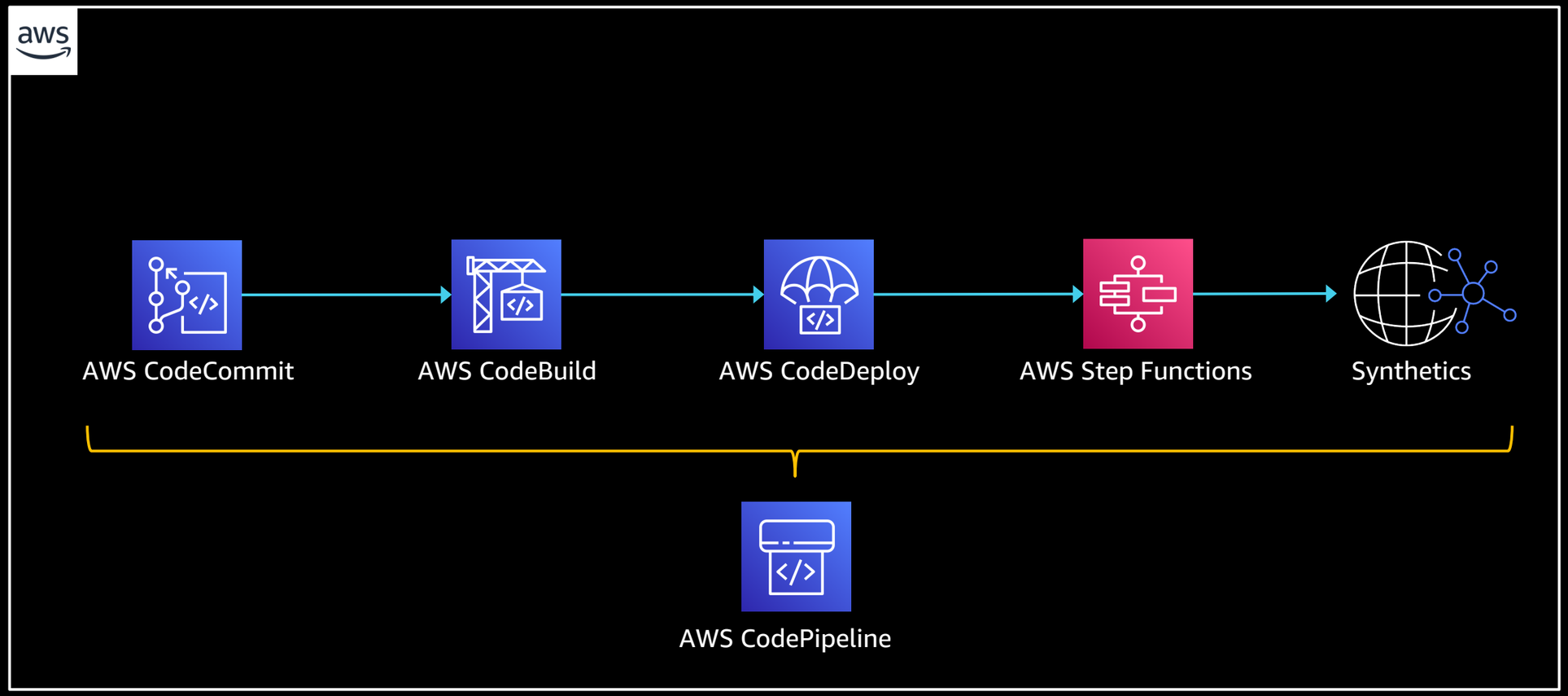
AWS service integrations and first-class support for your own services



Auditing

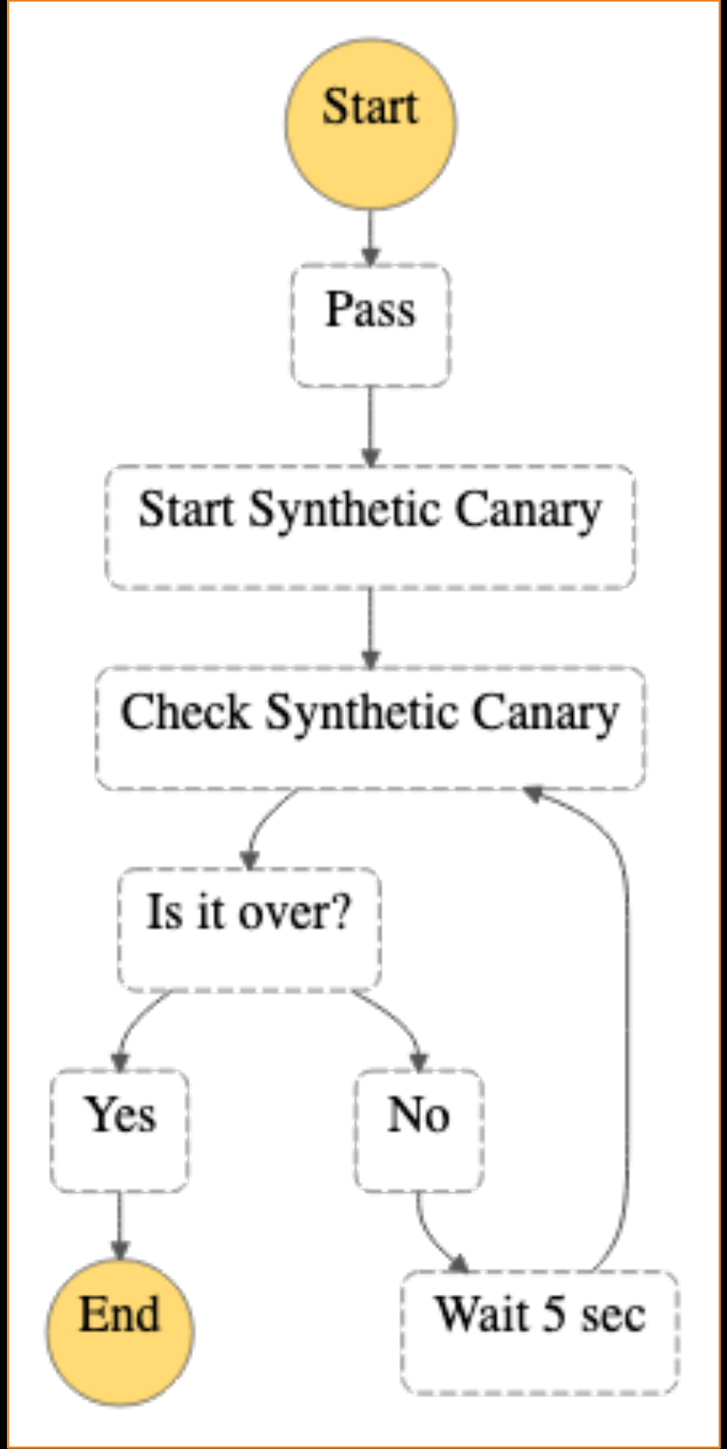
Auditable execution history and visual monitoring

Solution


aws
AWS CodeCommit
AWS CodeBuild
AWS CodeDeploy
AWS Step Functions
Synthetics
AWS CodePipeline

Solution: Continued

- Several Synthetics checks
  - Heartbeat
  - UI / API
  - Performance
- One state machine per check
- Checks are run once or multiple times
- Checks raise alarms for anomalies
- Checks can trigger pipeline rollbacks



Step Functions ×

State machines
Activities

Feature spotlight
Join our feedback panel

Synthetics

Edit | Start execution | Delete | Actions ▼

Details

ARN
arn:aws:states:us-east-1:440695699313:stateMachine:Synthetics

IAM role ARN
arn:aws:iam::440695699313:role/service-role/StepFunctions-Synthetics-role-3a122651

Type
Standard

Creation date
Oct 7, 2020 02:48:39.938 PM

Executions | Logging | Definition | Tags

Definition

Export ▼ Layout ▼



© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

What I showed you

1. Step Functions state machine

2. How to integrate and execute a canary as part of a pipeline

3. Implementing types of canaries for pipelines (Heartbeat, UI / API testing)

Related sessions

BLD205 – Monitoring production services at Amazon

David Yanacek, Sr. Principal Engineer, AWS

BLD207 – Hands-off: Automating continuous delivery pipelines at Amazon

Clare Liguori, Principal Engineer, AWS

Thank you!



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.