splunk> .conf2017

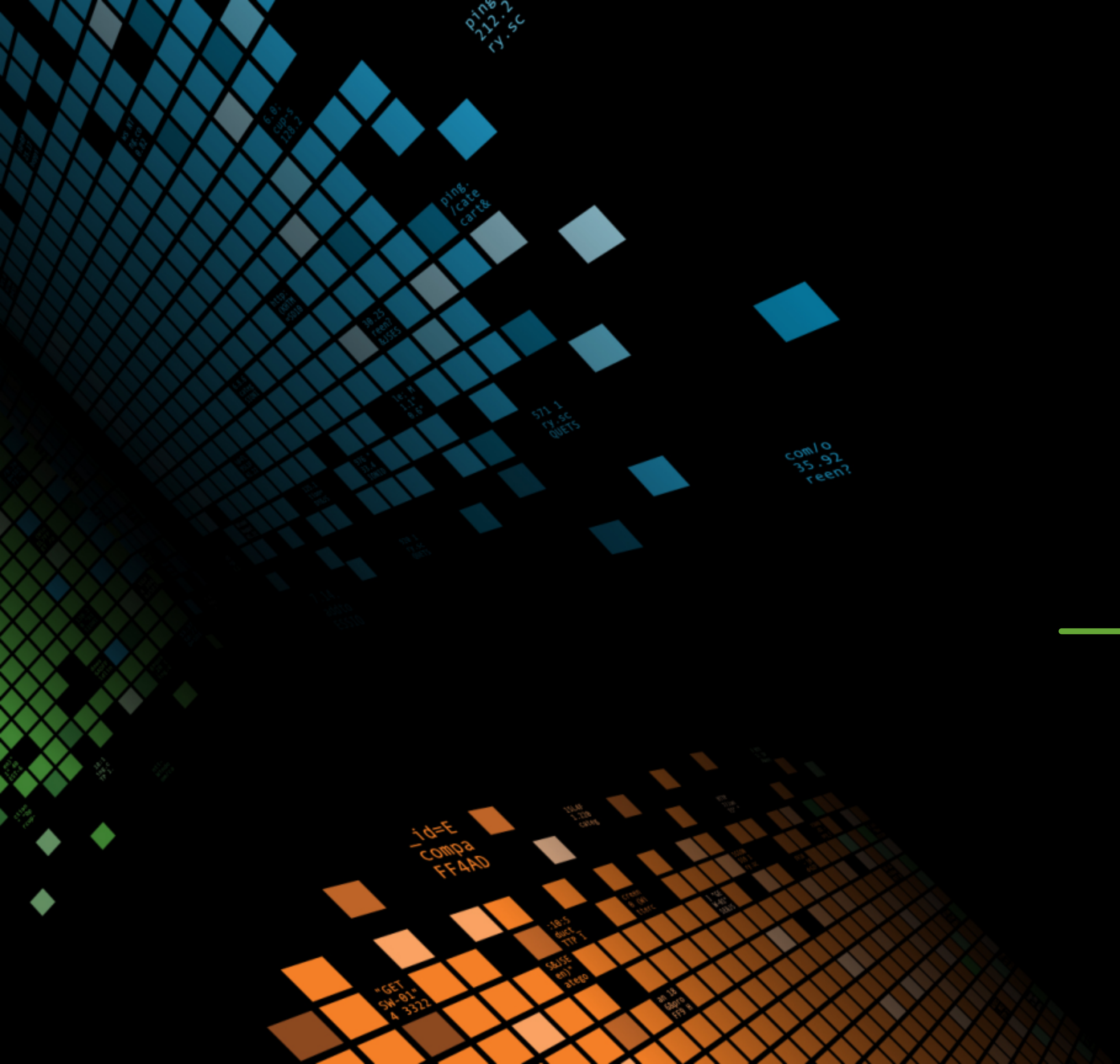
Automating Incident Response In The Cloud with Splunk Adaptive Response & AWS Lambda

September 2017| Washington, DC

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

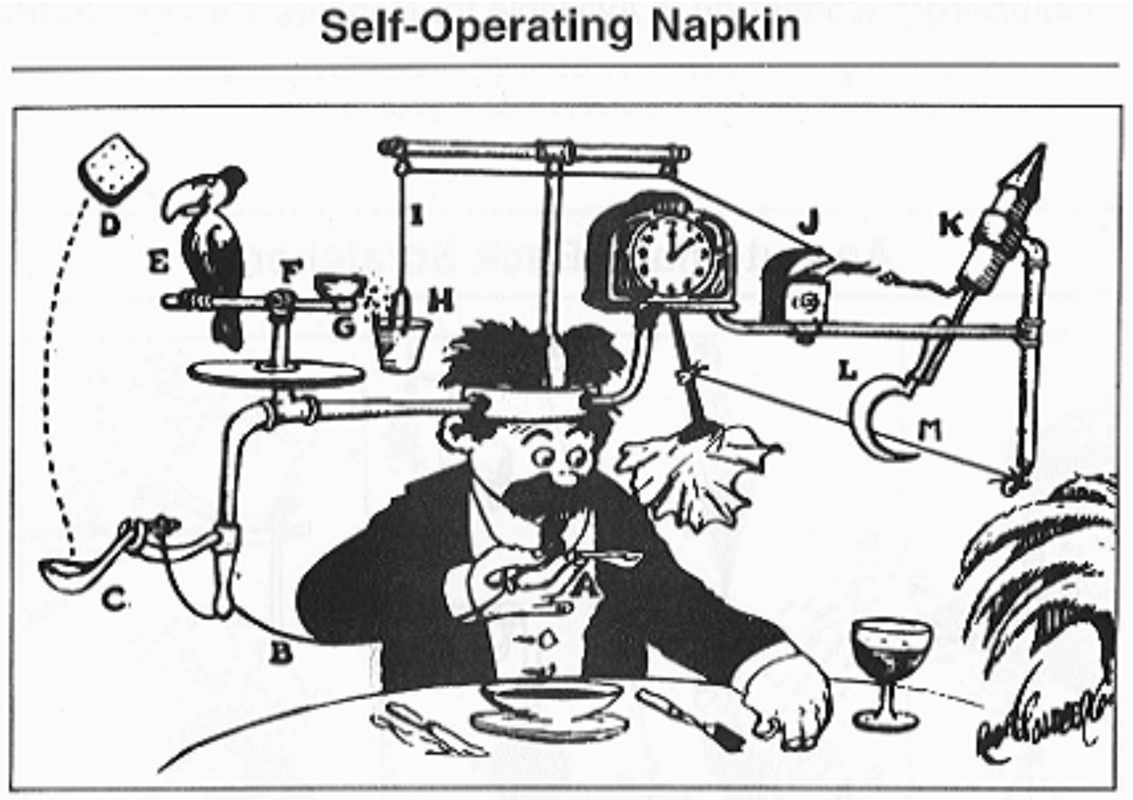


Introduction

Our Elevator Pitch

...

- If this then do that, oh and also that, then wait of a whatsit from a whoist
  - Depending on whoist do:
    - More that
    - Other this



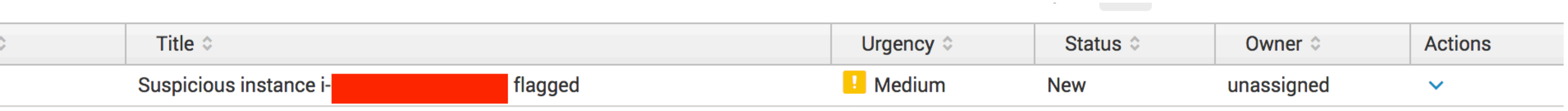
IRL

Splunk AR

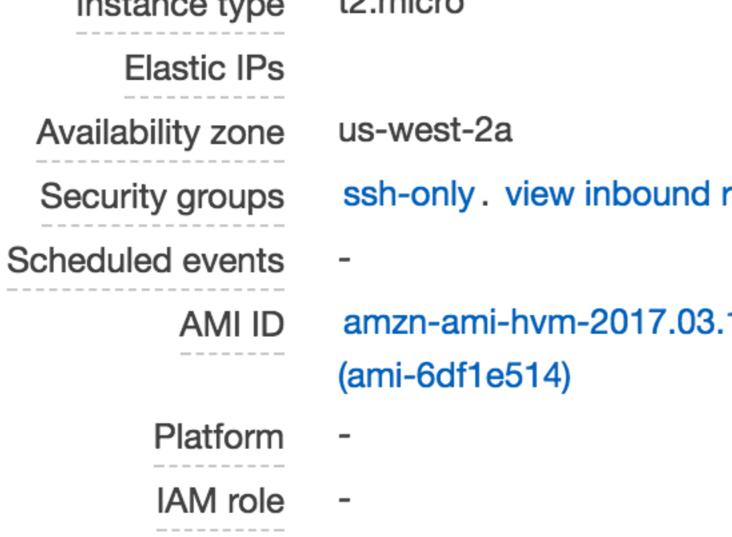
What is is

Where to get it

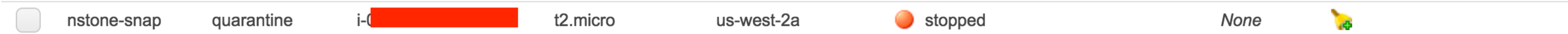
How to use it



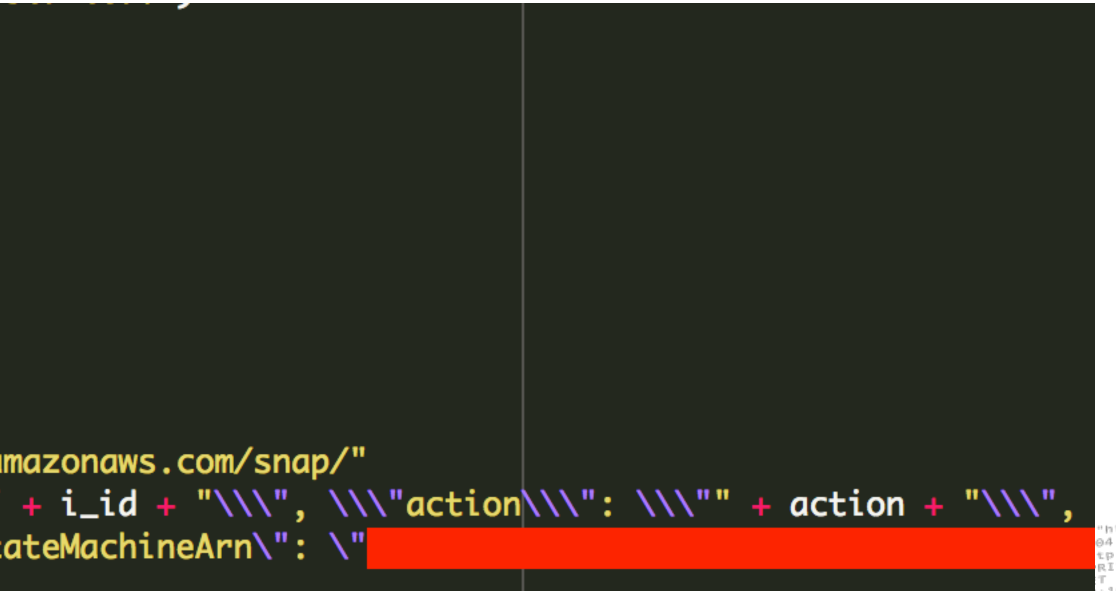
Title	Urgency	Status	Owner	Actions
Suspicious instance i-[redacted] flagged	! Medium	New	unassigned	⌄



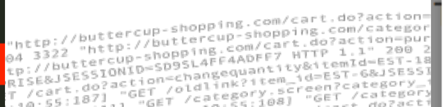
Instance type	t2.micro
Elastic IPs	
Availability zone	us-west-2a
Security groups	ssh-only . view inbound r
Scheduled events	-
AMI ID	amzn-ami-hvm-2017.03.1 (ami-6df1e514)
Platform	-
IAM role	-



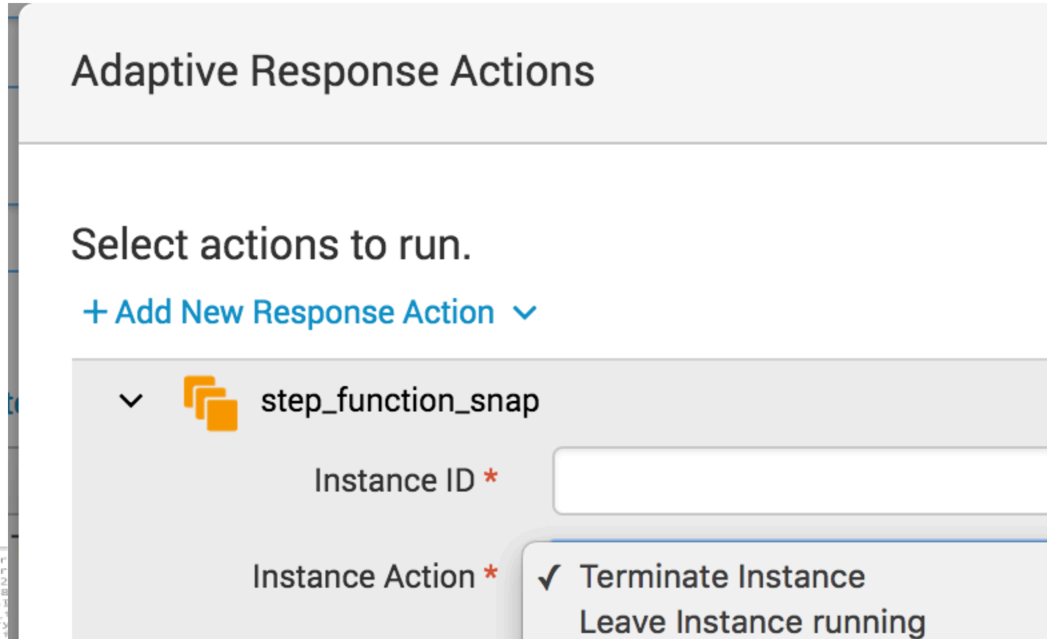
□	nstone-snap	quarantine	i-[redacted]	t2.micro	us-west-2a	● stopped	None	



```
amazonaws.com/snap/"
 + i_id + "\\\", \\\"action\\\": \\\"" + action + "\\\",
ateMachineArn\": \"[redacted]
```



```
                                    cup-shopping.com/cart.do?action=
"http://buttercup-shopping.com/categor
04 3322 "http://buttercup-shopping.com/cart.do?action=pur
tp://buttercup-shopping.com/cart.do?action=pur HTTP 1.1" 200 2
RISE&JSESSIONID=SD9SL4FF4ADFF7 HTTP 1.1" 200 2
T /cart.do?action=changequantity&itemId=EST-18
:10:55:187] "GET /oldlink?item_id=EST-6&JSESSI
```



Adaptive Response Actions

Select actions to run.

+ Add New Response Action ⌄

⌄ step_function_snap

Instance ID *

Instance Action *

✓ Terminate Instance
Leave Instance running

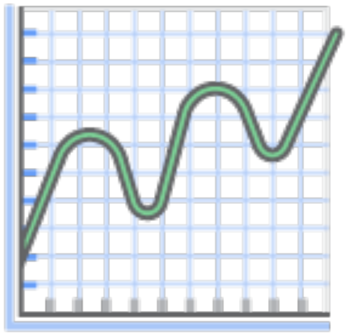
Benefits of AWS Lambda

No Servers to Manage

AWS Lambda handles:

- Operations and management
- Provisioning and utilization
- Scaling
- Availability and fault tolerance

Continuous Scaling



Automatically scales your application, running code in response to each trigger

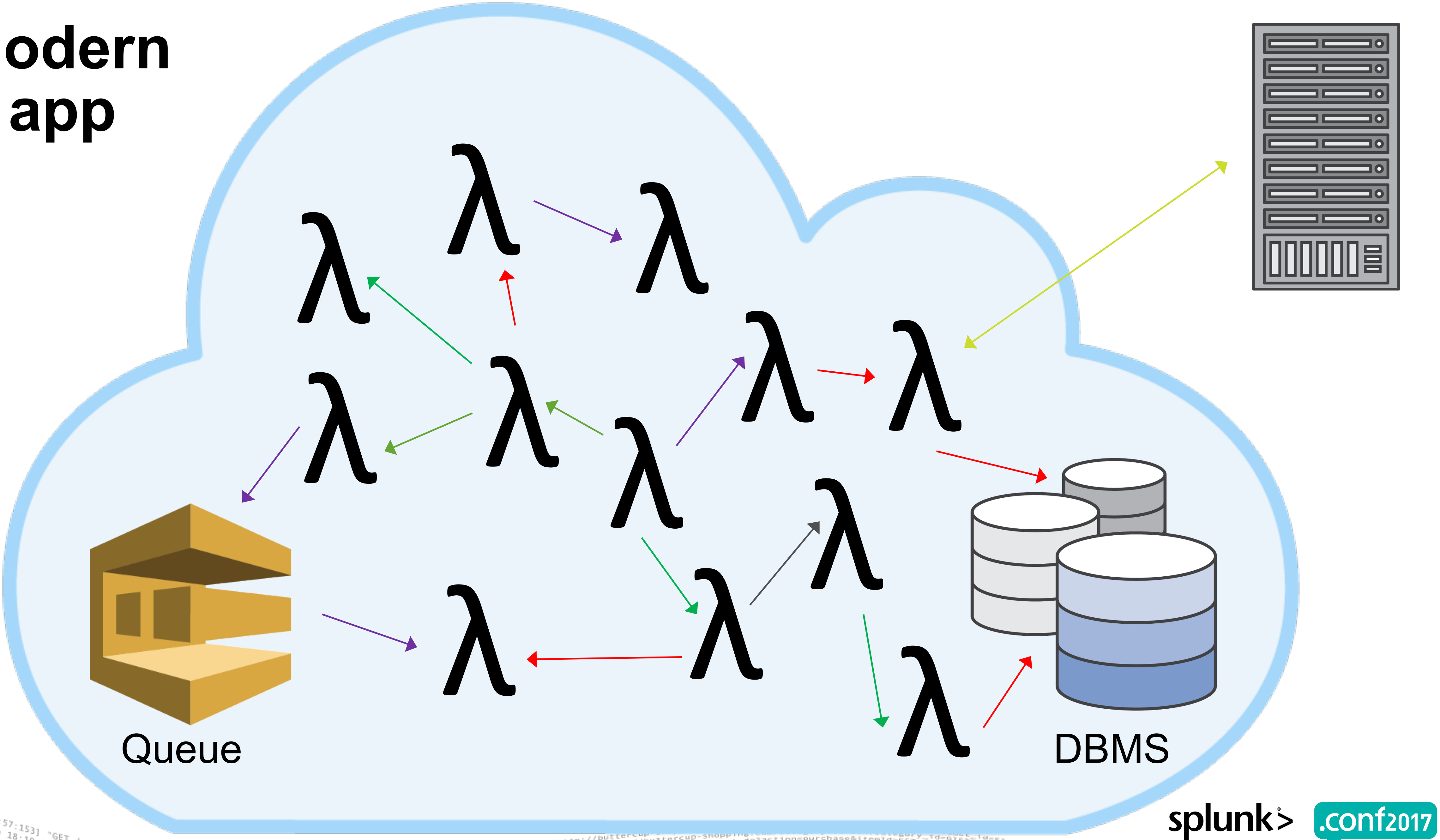
Your code runs in parallel and **processes each trigger individually**, scaling precisely with the size of the workload

Subsecond Metering

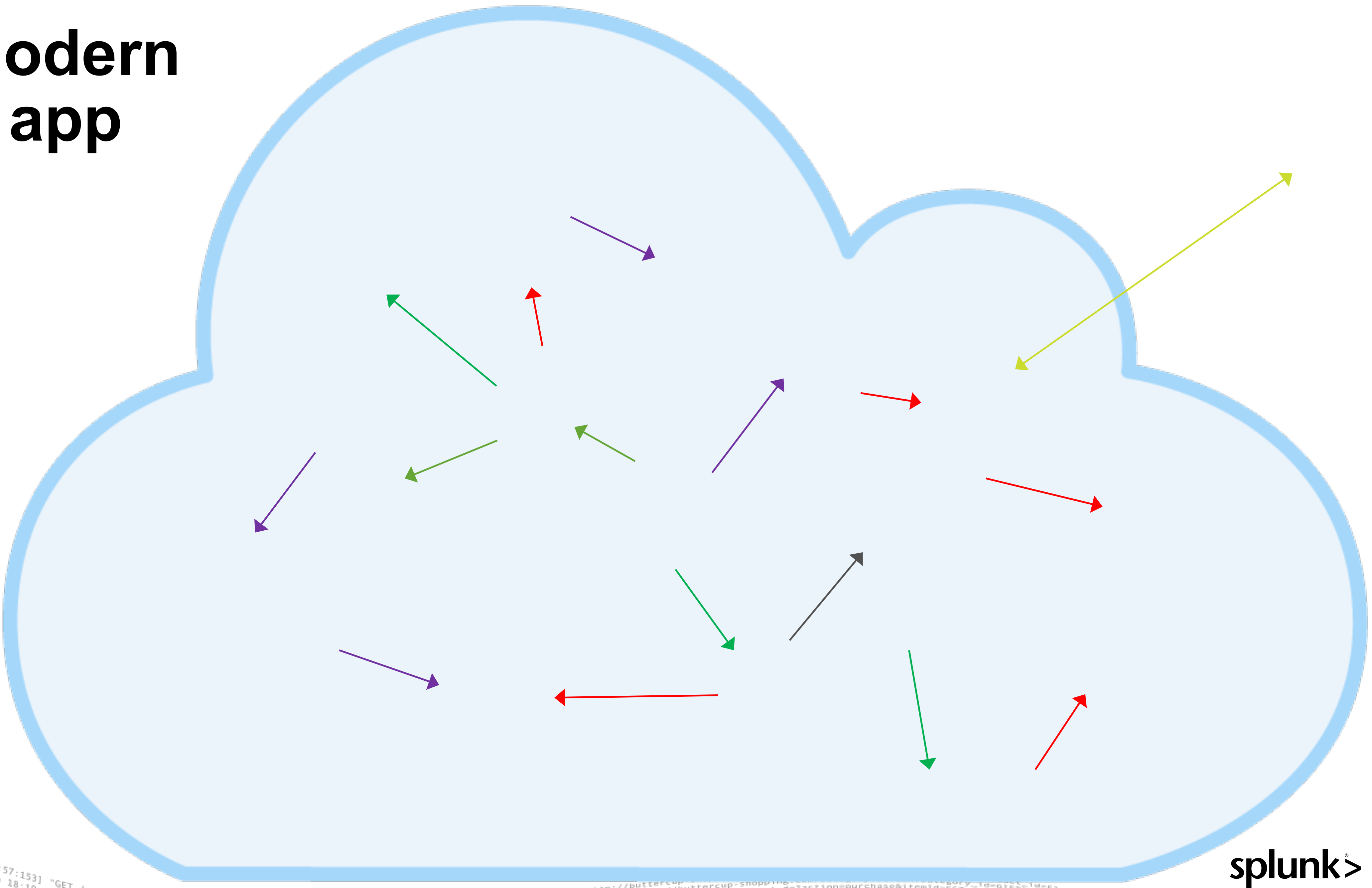
Pricing

- CPU and Network scaled based on RAM (128 MB to 1500 MB)
- $0.20 per 1M requests
- Price per 100ms

Modern app



Modern app



Turning functions into apps

“I want to sequence functions”

“I want to run functions in parallel”

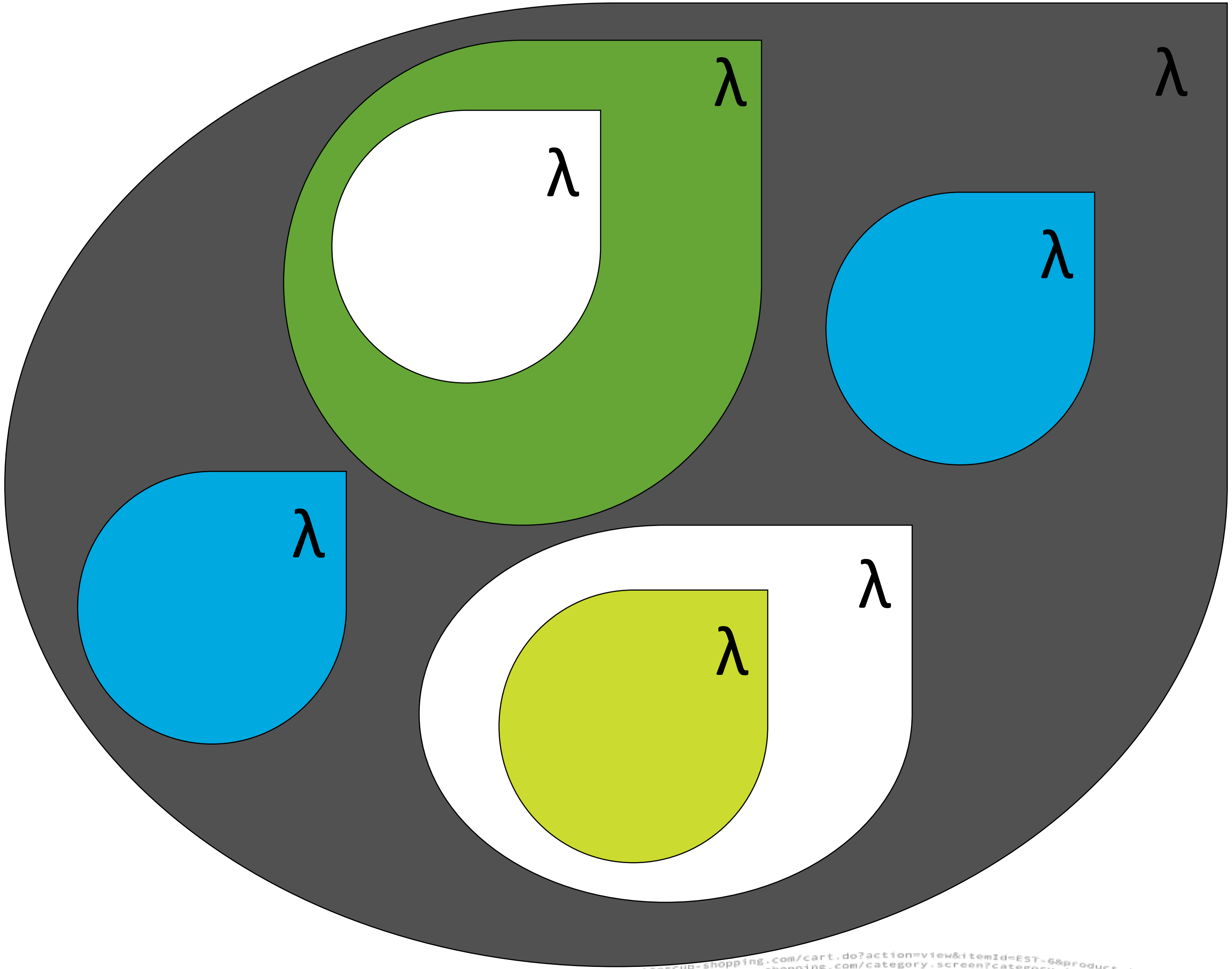
“I want to select functions based on data”

“I want to retry functions”

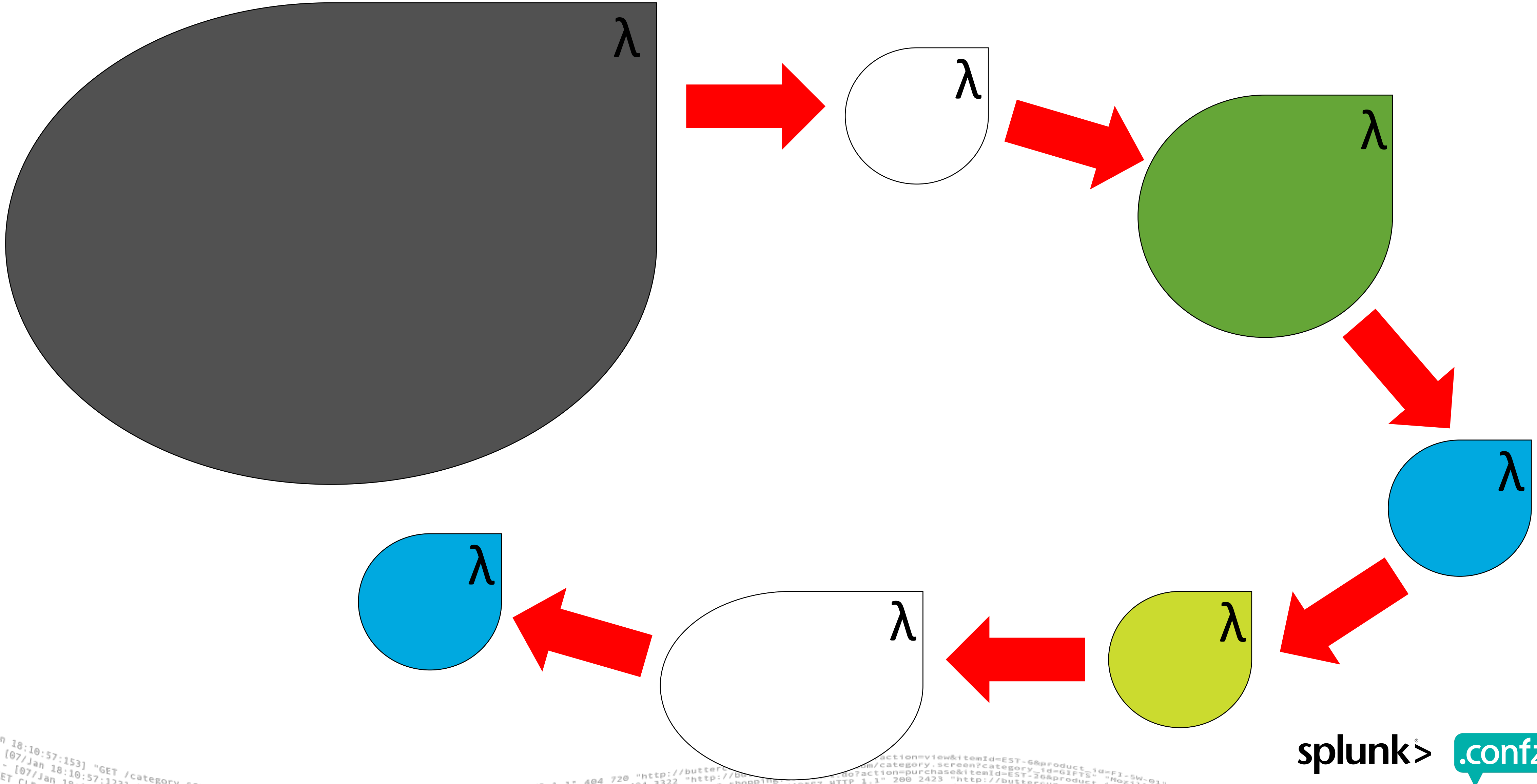
“I want try/catch/finally”

“I have code that runs for hours”

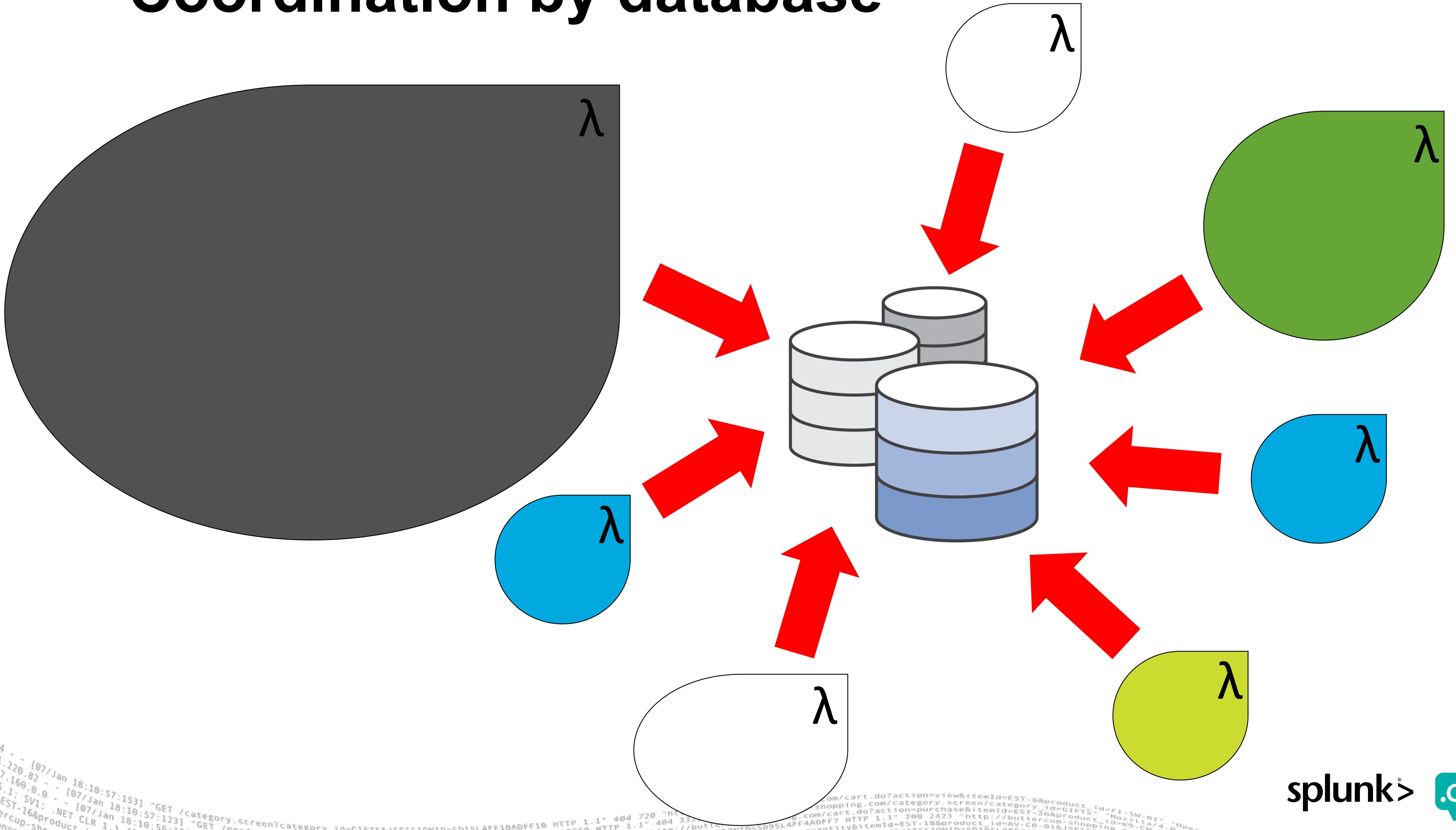
Coordination by method call



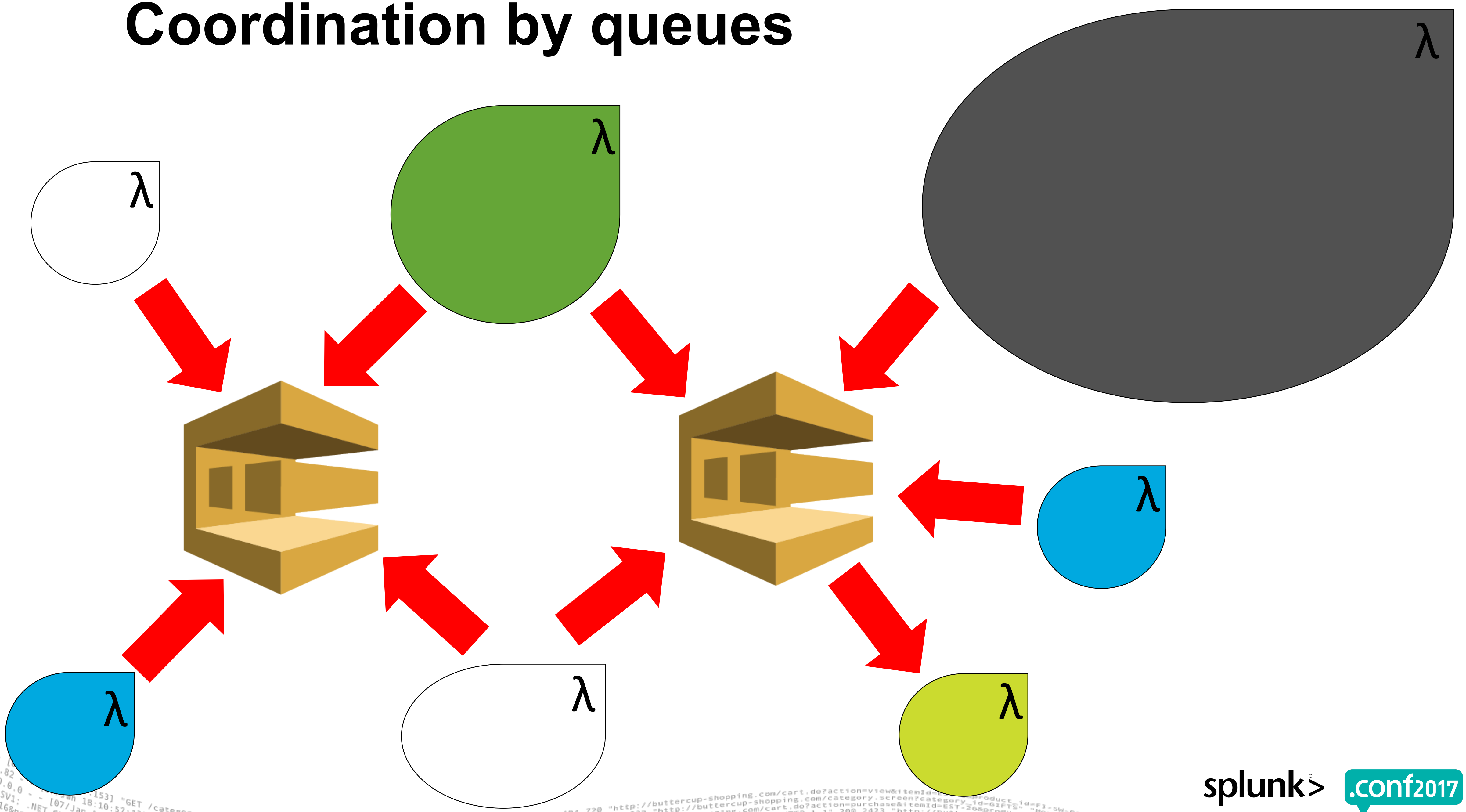
Coordination by function chaining

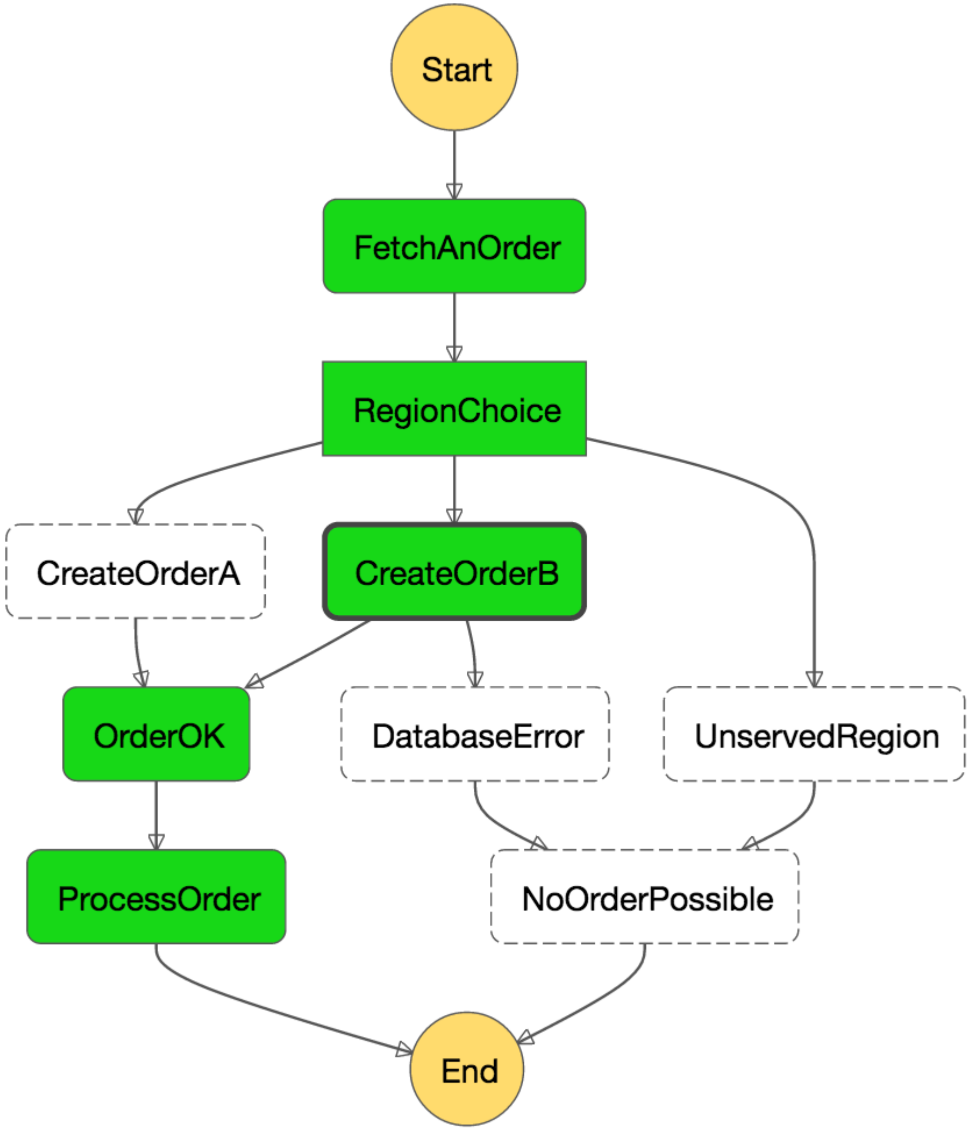


Coordination by database



Coordination by queues





AWS Step Functions

Benefits of AWS Step Functions

Productivity

Easy to connect and coordinate distributed components and microservices to quickly create apps

Agility

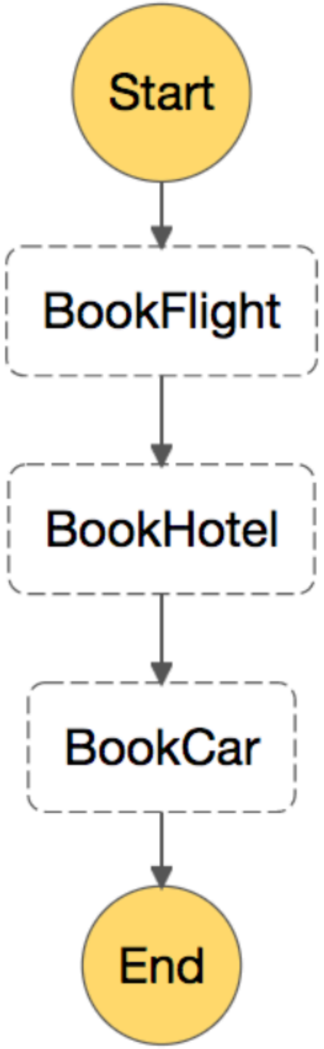
Diagnose and debug problems faster

Adapt to change

Resilience

Manages the operations and infrastructure of service coordination to ensure availability at scale, and under failure

Run Each Step in Sequence

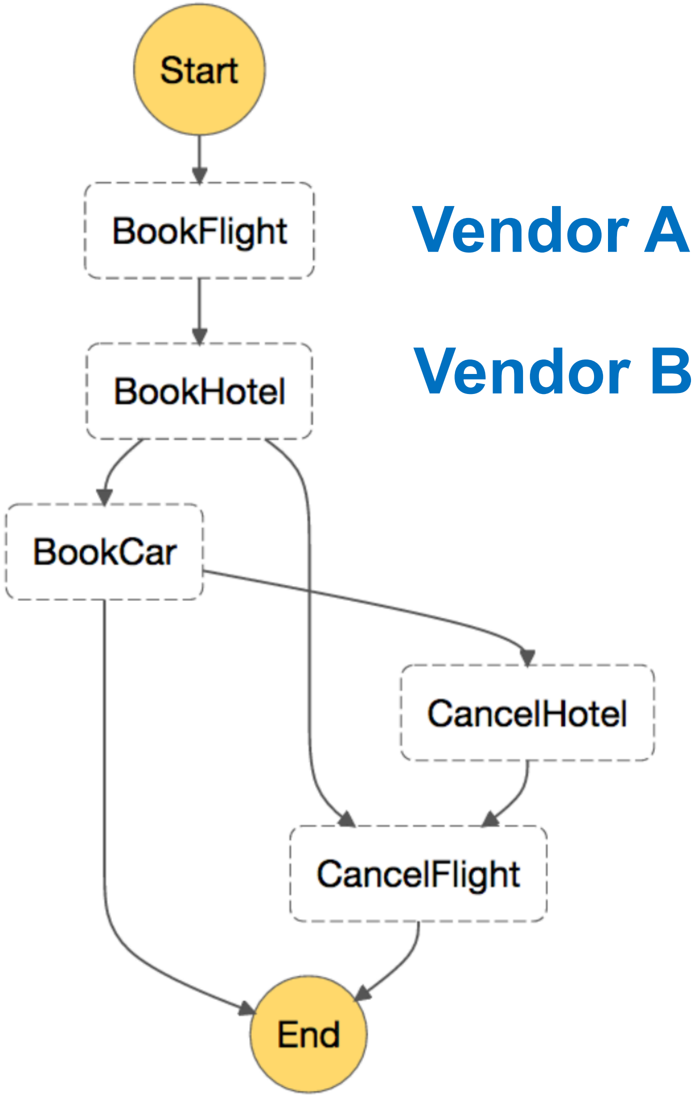


Vendor A

Vendor B

Vendor C

…and Unwind if My Plans Fail at Any Point



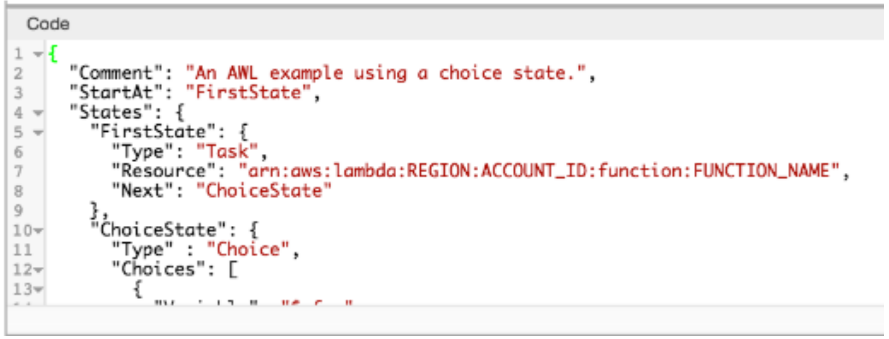
Vendor A

Vendor B

Vendor C

Application Lifecycle in AWS Step Functions

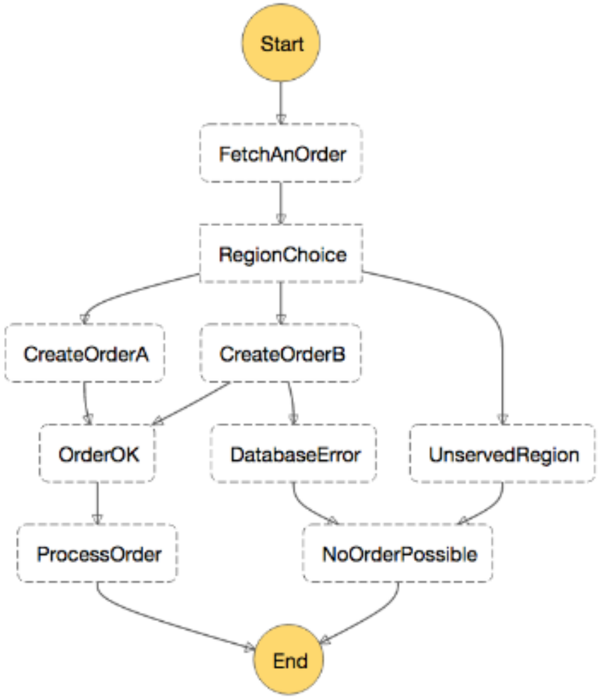
Define in JSON



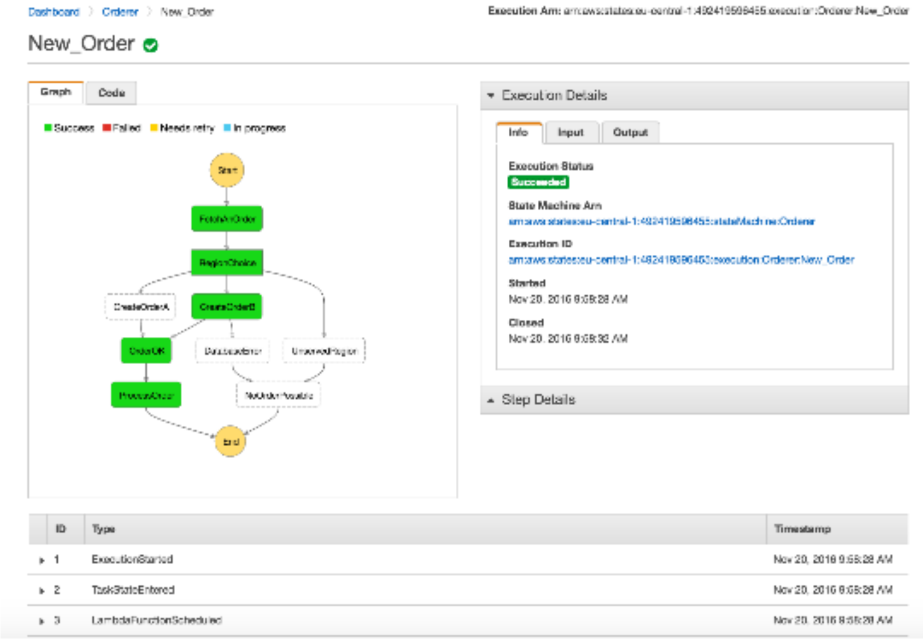
Code

```
{
  "Comment": "An AWL example using a choice state.",
  "StartAt": "FirstState",
  "States": {
    "FirstState": {
      "Type": "Task",
      "Resource": "arn:aws:lambda:REGION:ACCOUNT_ID:function:FUNCTION_NAME",
      "Next": "ChoiceState"
    },
    "ChoiceState": {
      "Type" : "Choice",
      "Choices": [
        {
```

Visualize in the Console



Monitor Executions



Execution Arn: arn:aws:states:eu-central-1:492419598455:execution:Orderer:New_Order

New_Order ✓

Execution Status
Succeeded

State Machine Arn
arn:aws:states:eu-central-1:492419598455:stateMachine:Orderer

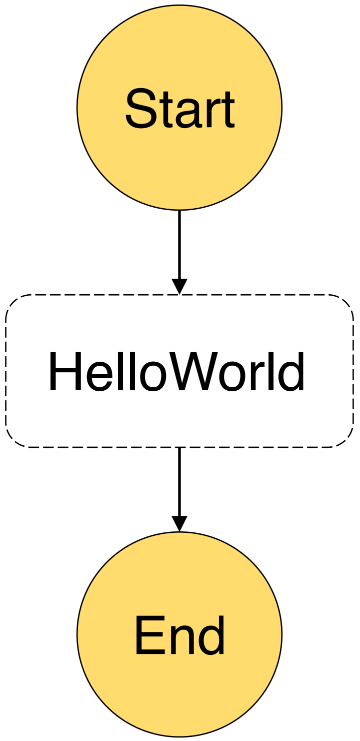
Execution ID
arn:aws:states:eu-central-1:492419598455:execution:Orderer:New_Order

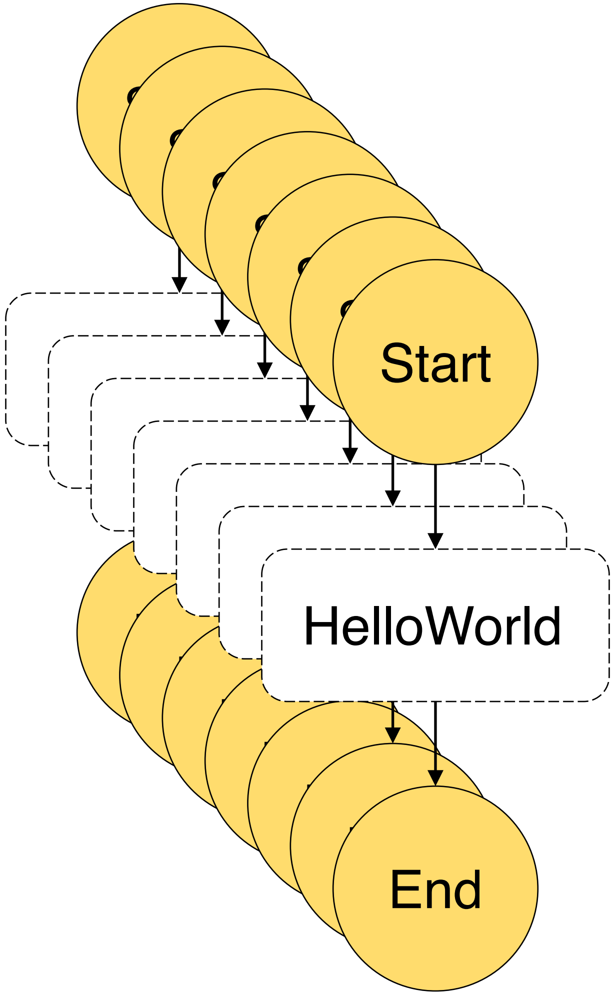
Started
Nov 20, 2016 9:58:28 AM

Closed
Nov 20, 2016 9:58:32 AM

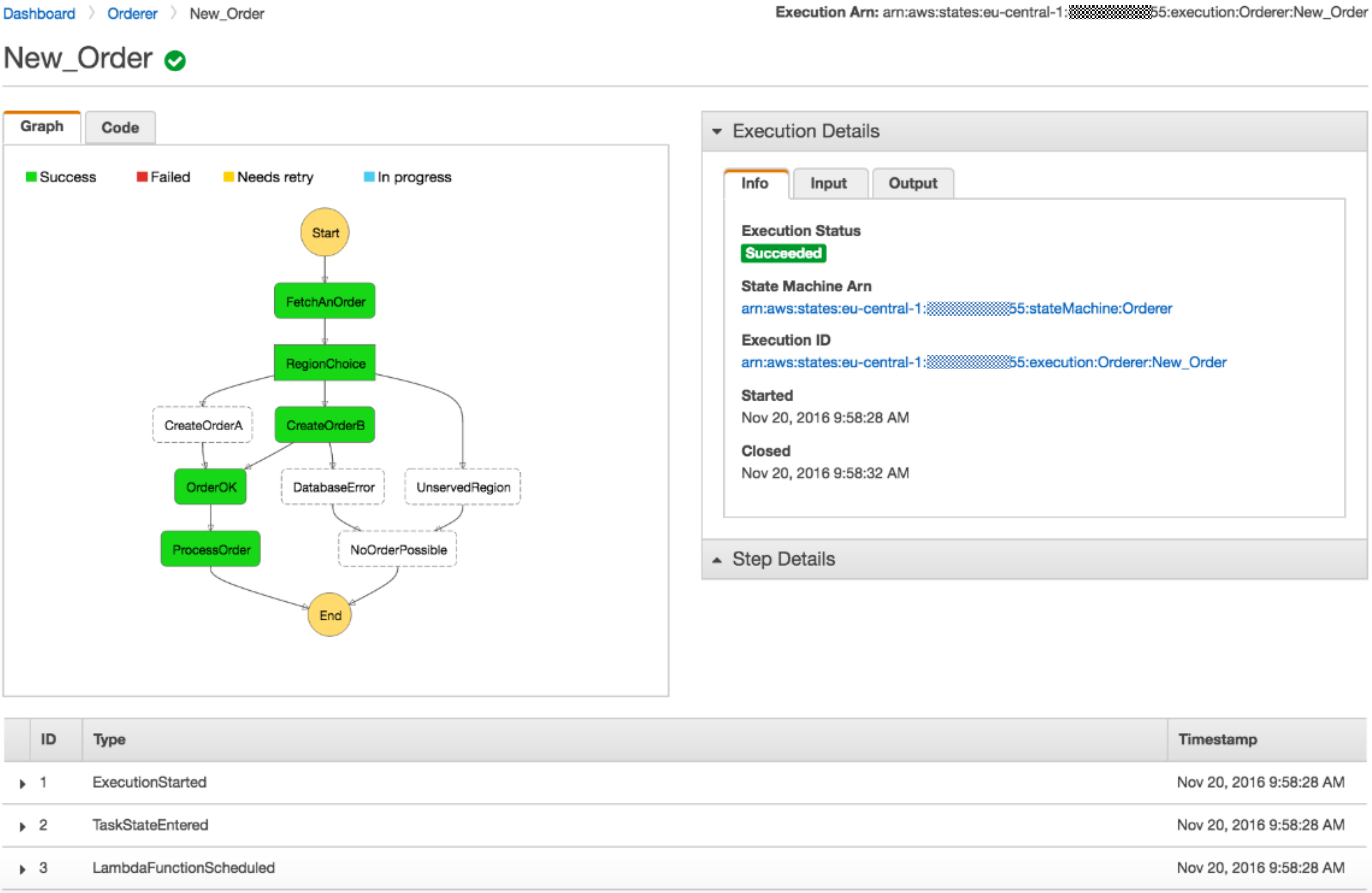
ID	Type	Timestamp
▸ 1	ExecutionStarted	Nov 20, 2016 9:58:28 AM
▸ 2	TaskStateEntered	Nov 20, 2016 9:58:28 AM
▸ 3	LambdaFunctionScheduled	Nov 20, 2016 9:58:28 AM

Execute One or One Million





Monitor Executions from the Console



Execution Arn: arn:aws:states:eu-central-1:[redacted]55:execution:Orderer:New_Order

New_Order ✓

▾ Execution Details

Execution Status
Succeeded

State Machine Arn
arn:aws:states:eu-central-1:[redacted]55:stateMachine:Orderer

Execution ID
arn:aws:states:eu-central-1:[redacted]55:execution:Orderer:New_Order

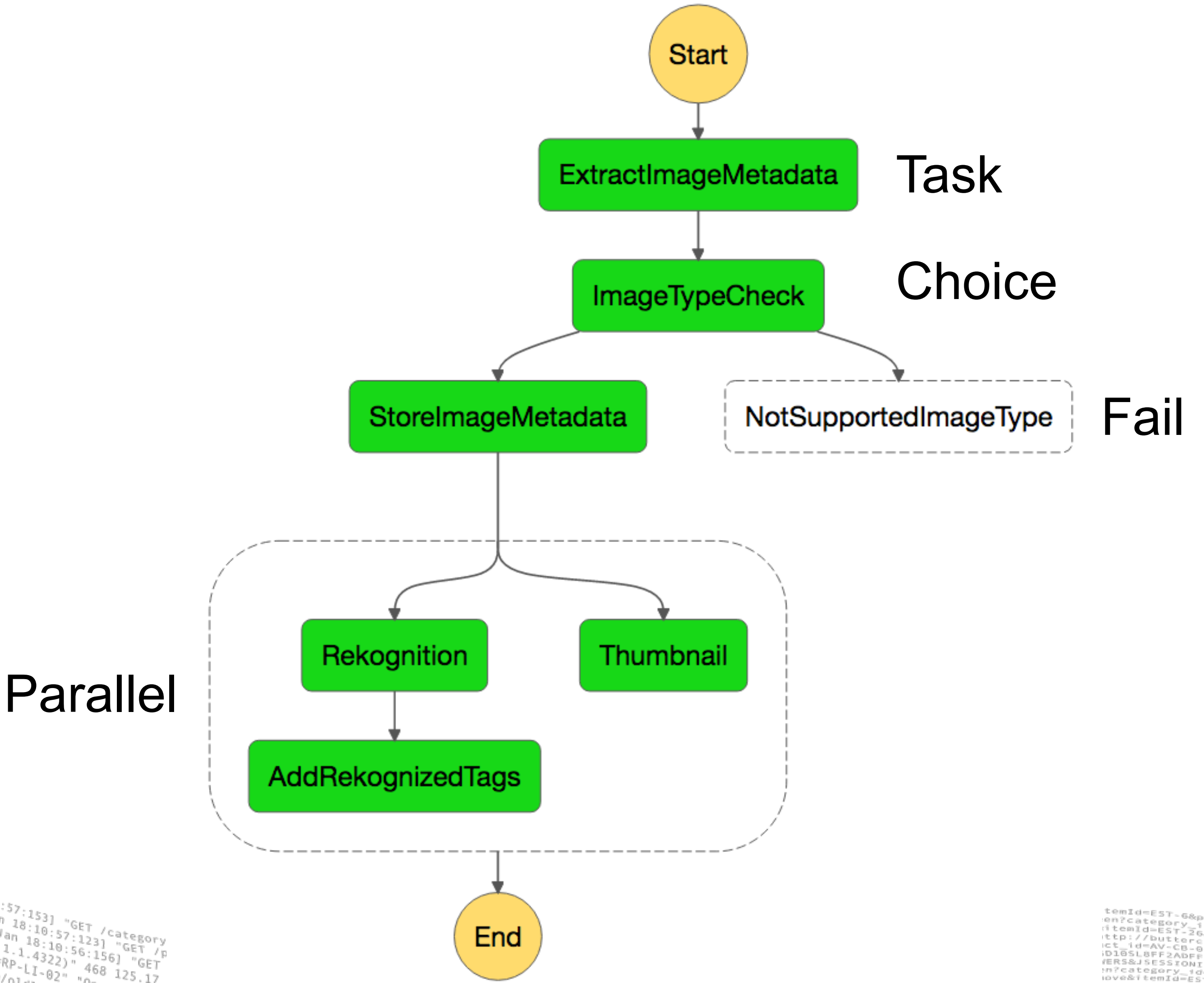
Started
Nov 20, 2016 9:58:28 AM

Closed
Nov 20, 2016 9:58:32 AM

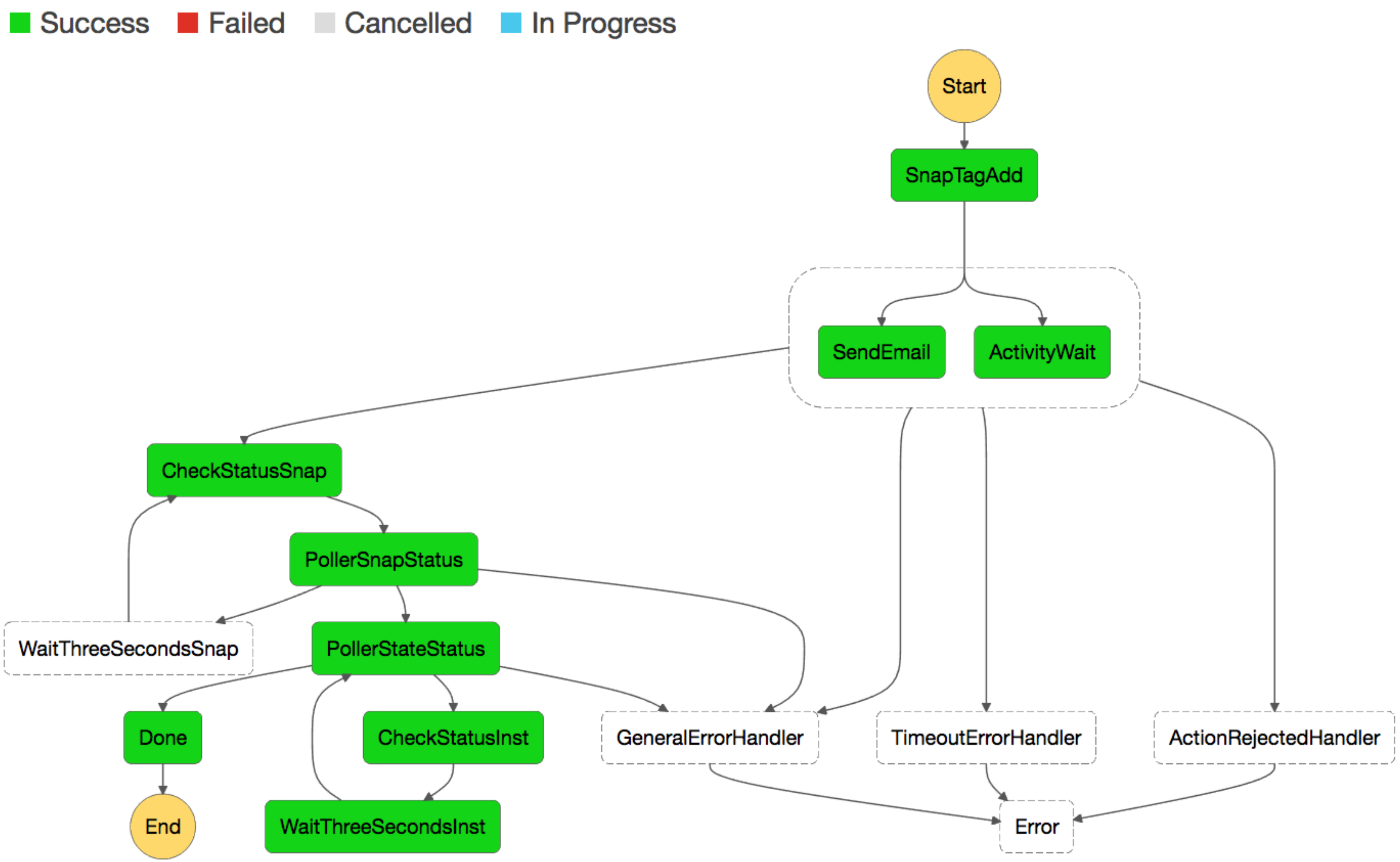
▴ Step Details

ID	Type	Timestamp
▸ 1	ExecutionStarted	Nov 20, 2016 9:58:28 AM
▸ 2	TaskStateEntered	Nov 20, 2016 9:58:28 AM
▸ 3	LambdaFunctionScheduled	Nov 20, 2016 9:58:28 AM

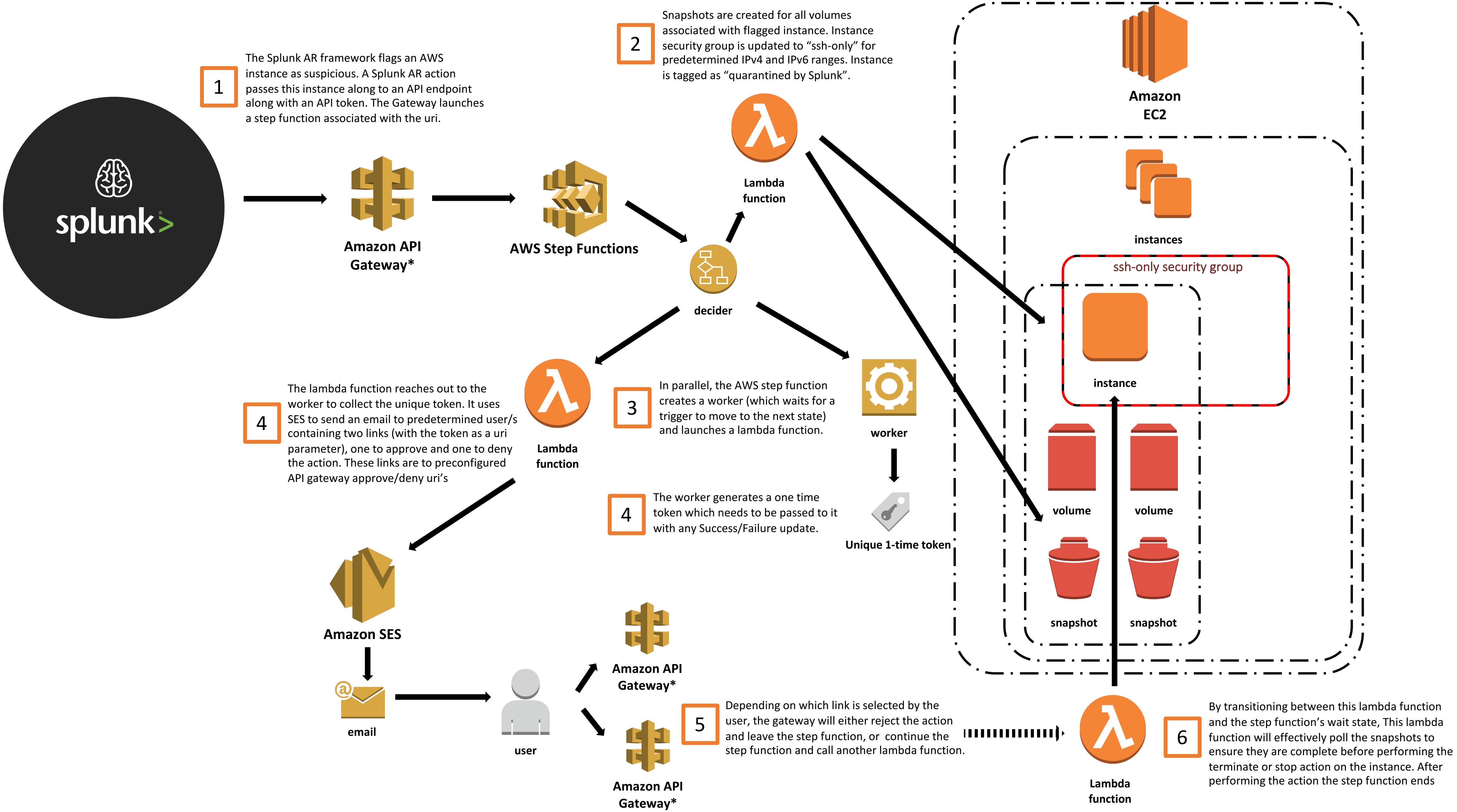
Build Visual Workflows from State Types



Our Step Function



Architecture



1 The Splunk AR framework flags an AWS instance as suspicious. A Splunk AR action passes this instance along to an API endpoint along with an API token. The Gateway launches a step function associated with the uri.

2 Snapshots are created for all volumes associated with flagged instance. Instance security group is updated to “ssh-only” for predetermined IPv4 and IPv6 ranges. Instance is tagged as “quarantined by Splunk”.

3 In parallel, the AWS step function creates a worker (which waits for a trigger to move to the next state) and launches a lambda function.

4 The lambda function reaches out to the worker to collect the unique token. It uses SES to send an email to predetermined user/s containing two links (with the token as a uri parameter), one to approve and one to deny the action. These links are to preconfigured API gateway approve/deny uri’s

4 The worker generates a one time token which needs to be passed to it with any Success/Failure update.

5 Depending on which link is selected by the user, the gateway will either reject the action and leave the step function, or continue the step function and call another lambda function.

6 By transitioning between this lambda function and the step function’s wait state, This lambda function will effectively poll the snapshots to ensure they are complete before performing the terminate or stop action on the instance. After performing the action the step function ends