Analyze Root Cause and End-User Impact using AWS X-Ray

Abhishek Singh
General Manager, AWS X-Ray

June 19th, 2019

aws

Agenda

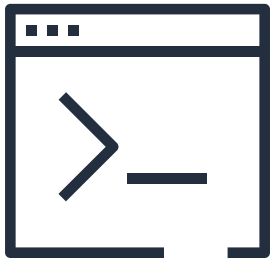
- Monitoring and debugging modern applications
- AWS X-Ray overview
- Root cause and end-user impact analysis using AWS X-Ray
- Demo
- Q&A

Monitoring and Debugging Modern Applications

Monitoring and Debugging Modern Applications

Built on Containers and Serverless

Distributed Microservices Architecture

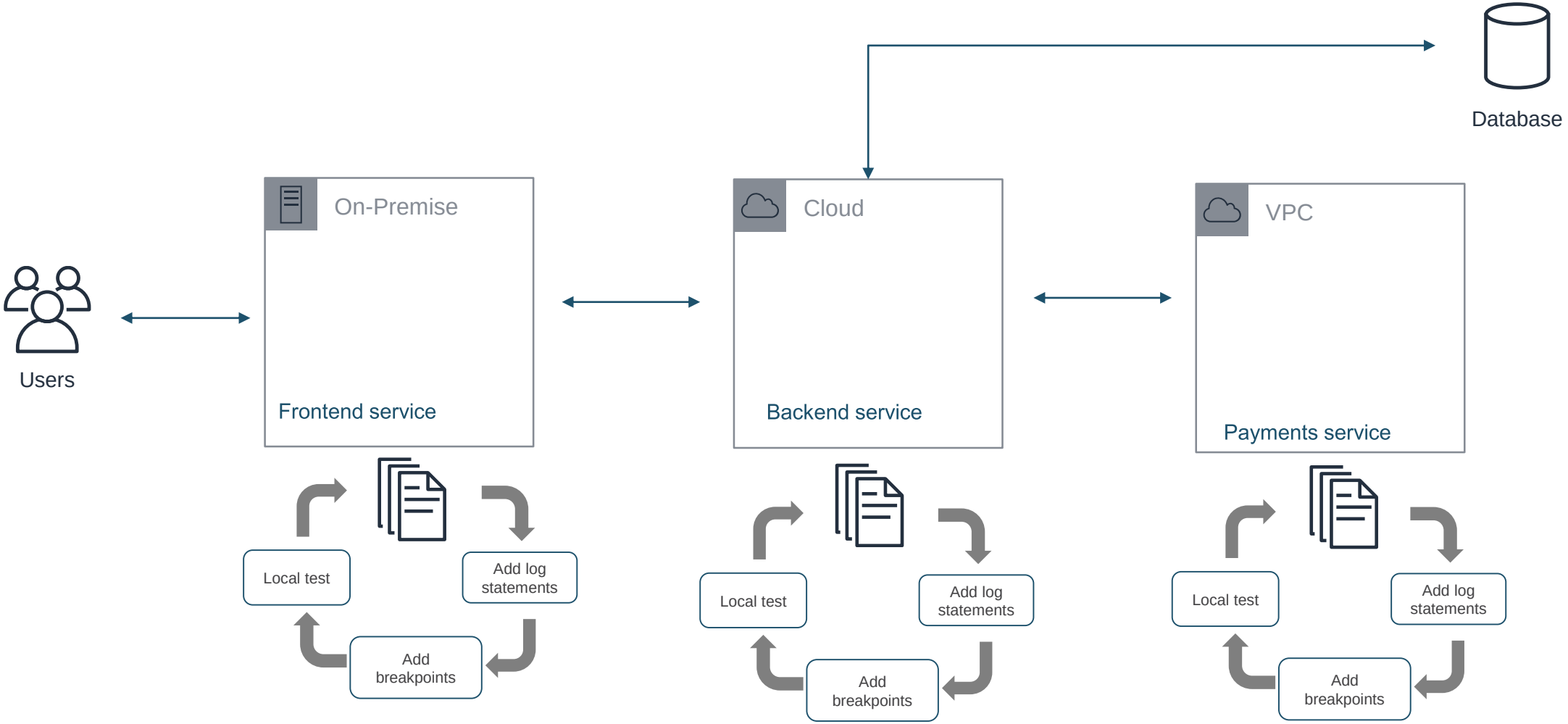


Autonomous and Independent Components

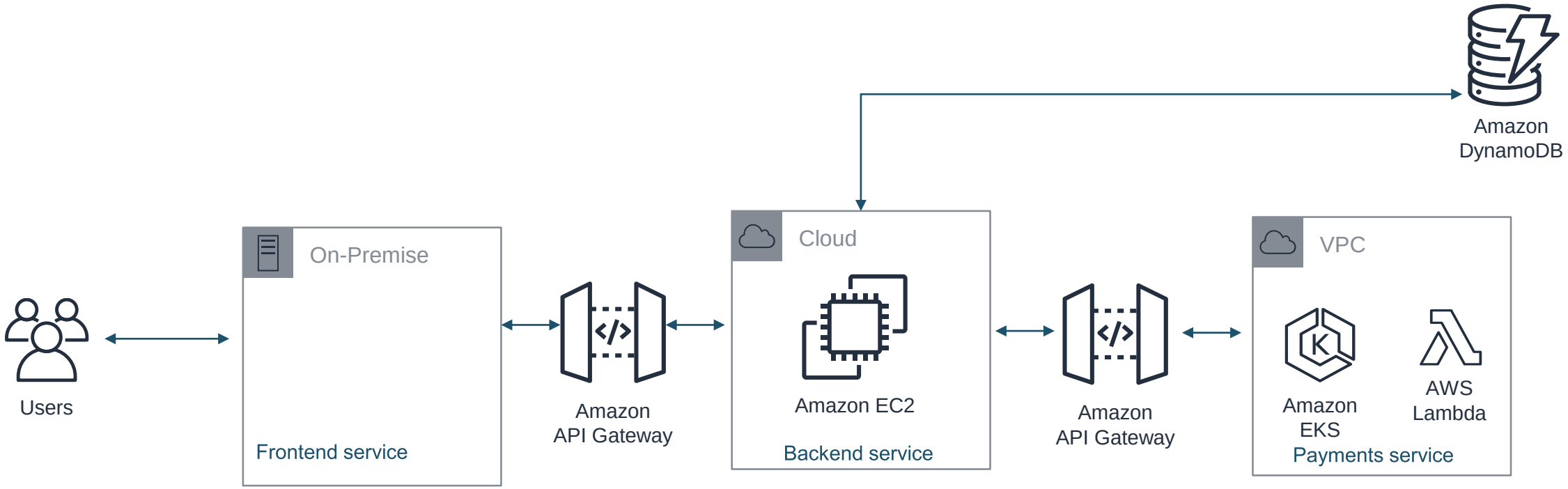
Analyze Impact of Isolated Errors

aws

Traditional Debugging Does Not Scale



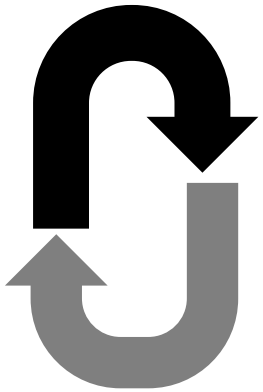
Tracing For Your Modern Applications



AWS X-Ray Overview

AWS X-Ray Overview

Analyze and Debug Issues Quickly



End-to-End View of Individual Services

Identify Customer Impact

Cloud Agnostic

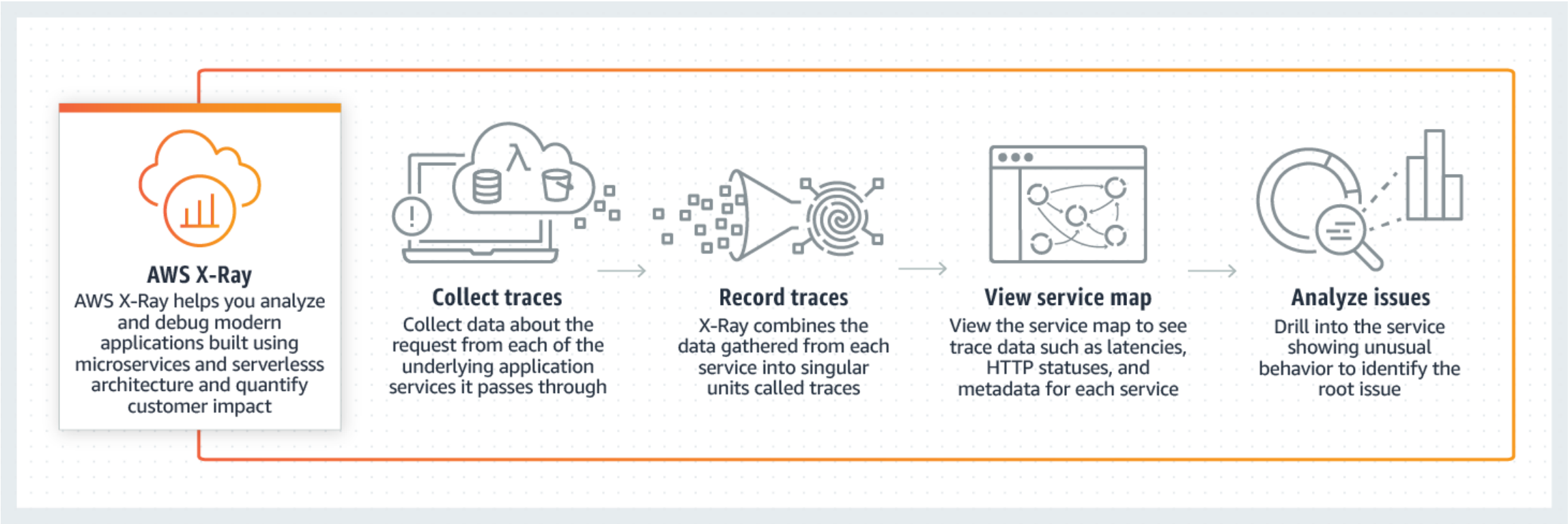
aws

AWS X-Ray Overview

AWS X-Ray helps developers analyze and debug applications built using microservices architecture to identify the root cause and end-user impact.

aws

How AWS X-Ray Works



AWS X-Ray Instrumentation

Strong Open Source Community

.NET
(with .NET Core 2.0)

Go (beta)

Java

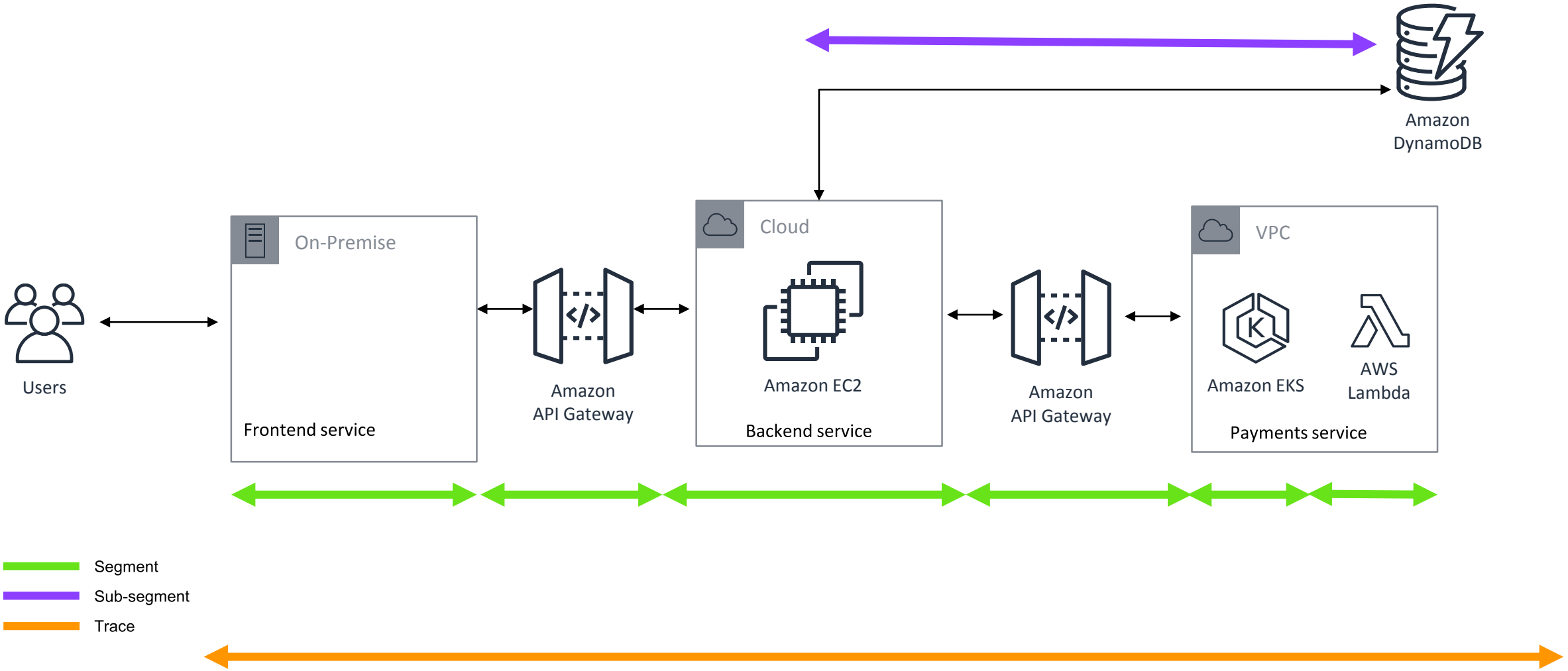
Python

Ruby (beta)

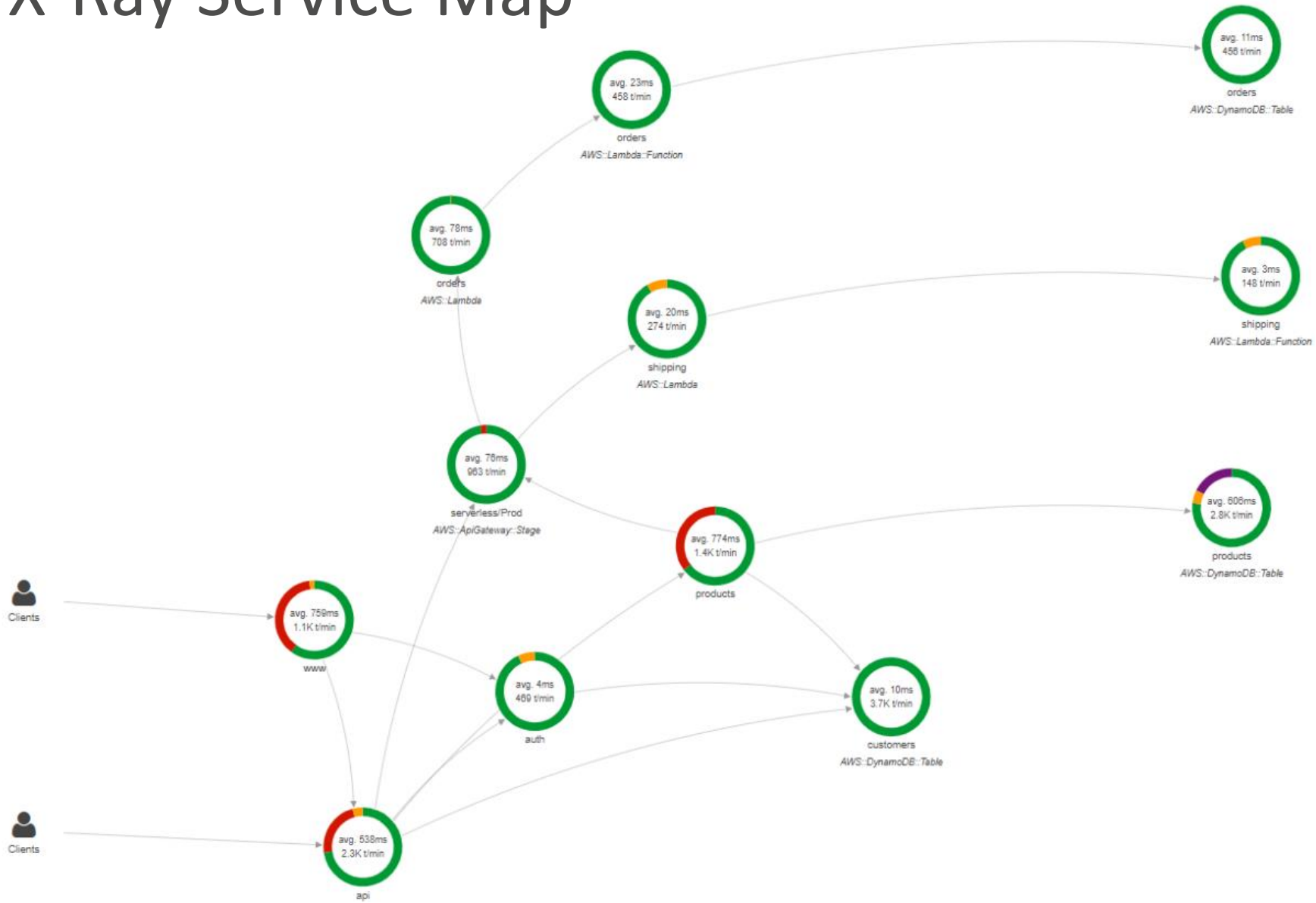
Node.js

aws

AWS X-Ray for your Modern Applications

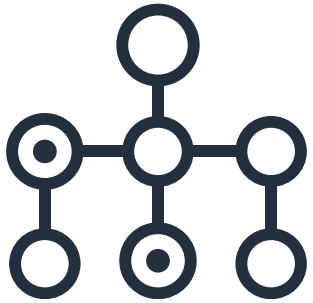


AWS X-Ray Service Map

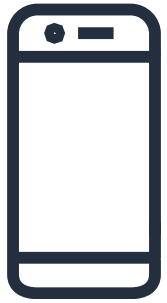


Root Cause and End-User Impact Analysis using AWS X-Ray

AWS X-Ray Groups



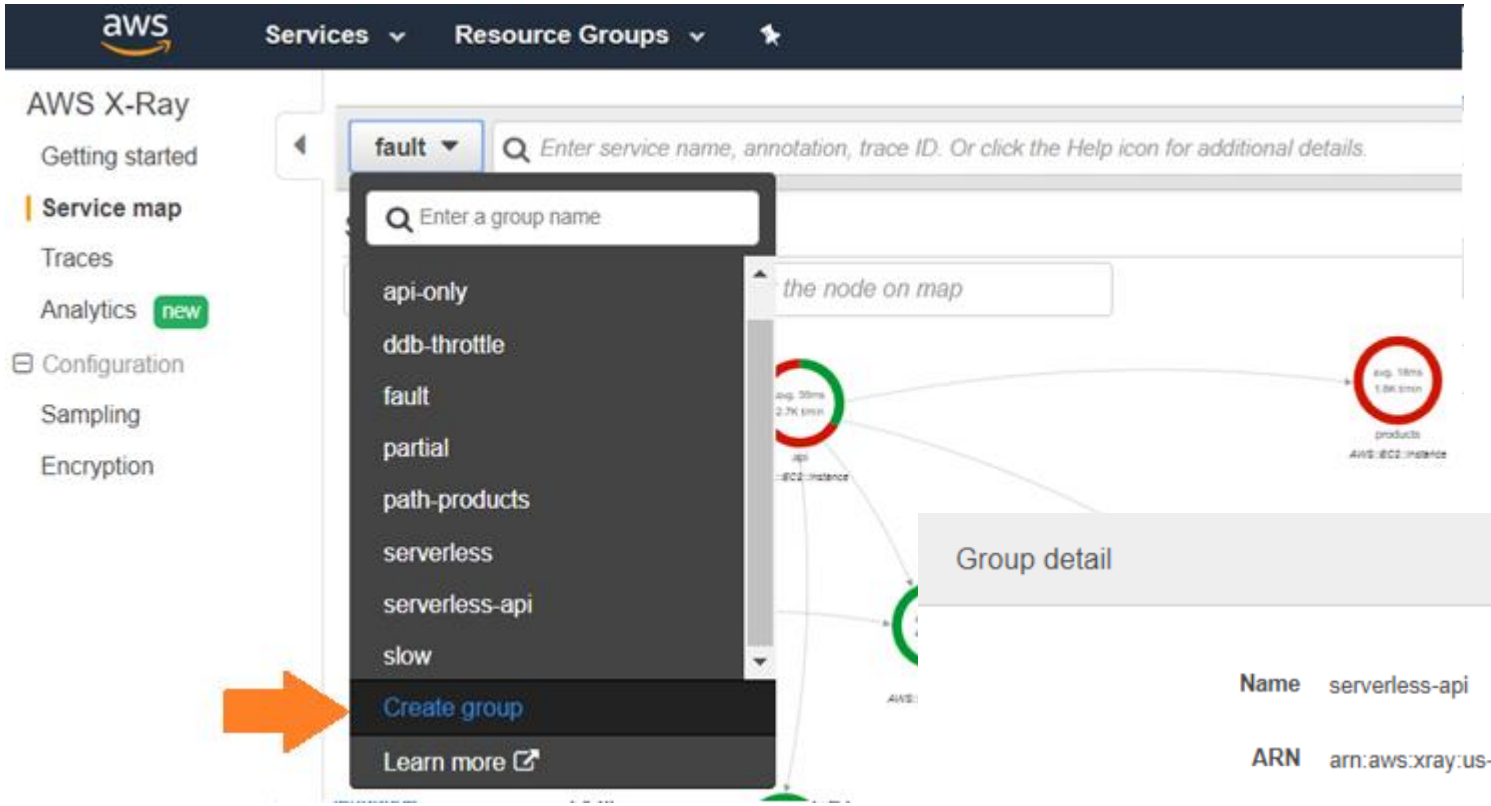
Focus on selected workflows

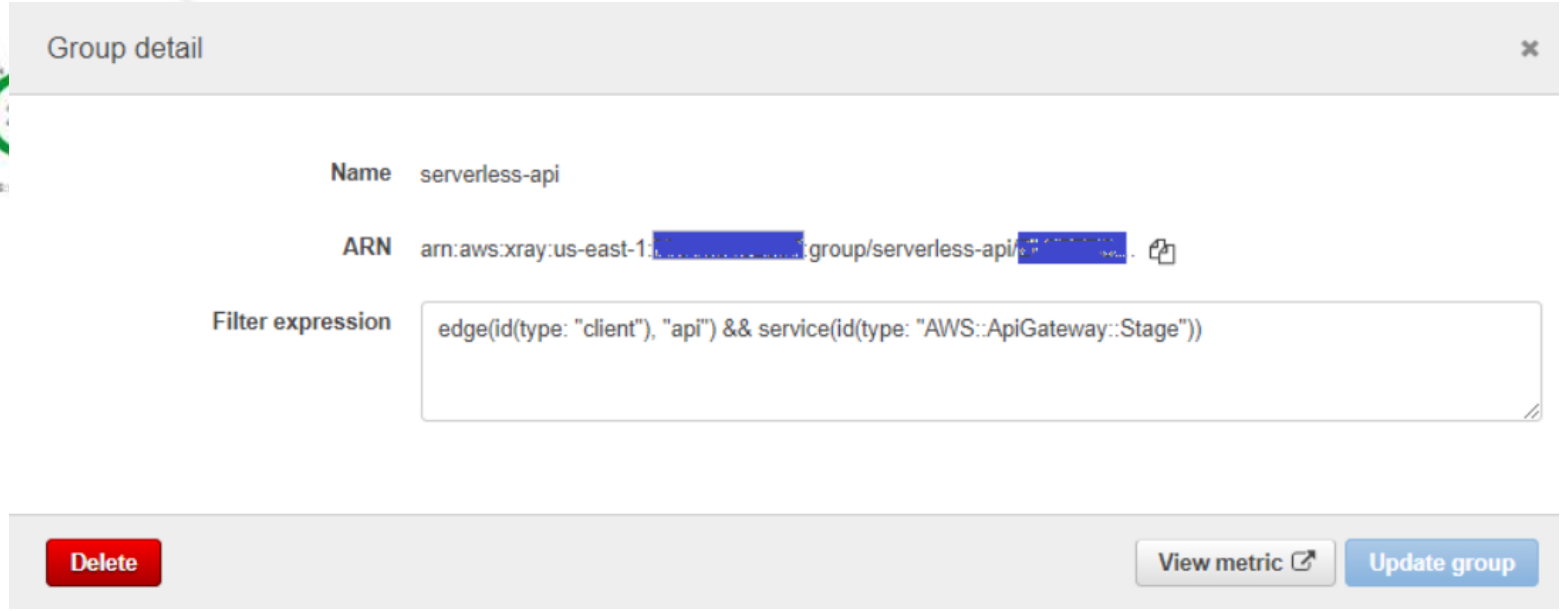


Get alerted on increased error/fault rates

Increase visibility on selected services/end-users

AWS X-Ray Groups



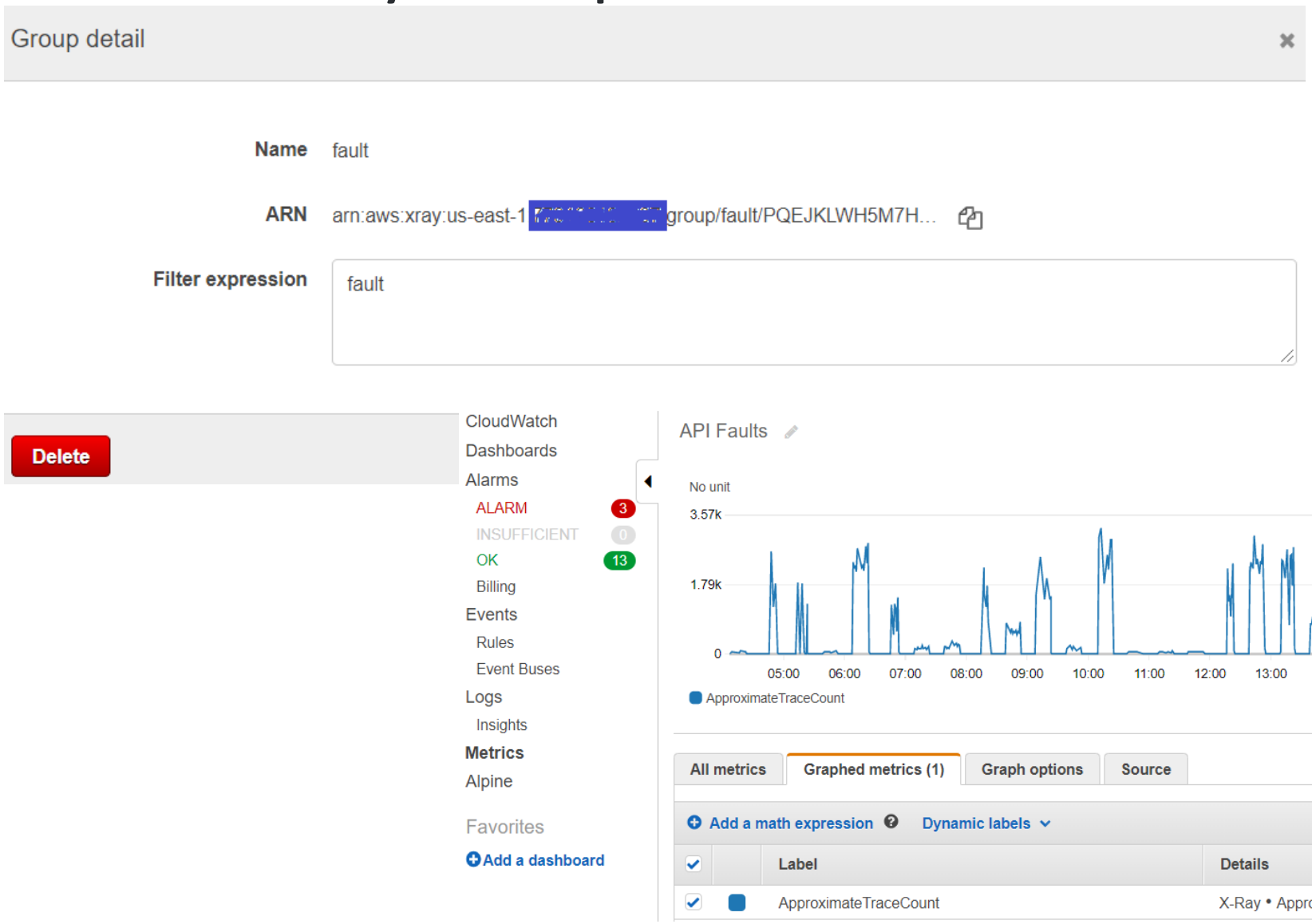


Focus on selected workflows

AWS X-Ray Groups

Focus on selected workflows

AWS X-Ray Groups



Get alerted on increased error/fault rates

AWS X-Ray Groups

Increase visibility on selected services/end-user

Group detail

Name slow

ARN arn:aws:xray:us-east-1:779168132807:group/slow/W3ULUGNWUD4N...

Filter expr

Delete

AWS X-Ray

- Getting started
- Service map
- Traces
- Analytics new
- Configuration
- Sampling
- Encryption

slow | service("serverless/Prod") { responsetime > 1.5 } | 2019/06/04 01:23 - 201

Trace overview

Group by: URL

Done 10

URL	Avg response time	% of Traces	Response
http://storefront.us-east-1.elasticbeanstalk.com/	1.7 sec	50.00%	2 OK, 0 Throttled, 0 Errors, 0 Faul
http://storefront.us-east-1.elasticbeanstalk.com/error/	3.0 sec	25.00%	0 OK, 0 Throttled, 0 Errors, 1 Faul
http://storefront.us-east-1.elasticbeanstalk.com/slow/	3.0 sec	25.00%	0 OK, 0 Throttled, 0 Errors, 1 Faul

Trace list

ID	Age	Method	Response	Response time	URL	Client IP
...f357dc91	2.4 hr	GET	200	1.6 sec	http://storefront.us-east-1.elasticbeanst...	18.213.102.12
...f1c1c029	2.5 hr	GET	200	1.8 sec	http://storefront.us-east-1.elasticbeanst...	18.213.102.12
...f6addd96	2.8 hr	GET	500	3.0 sec	http://storefront.us-east-1.elasticbeanst...	18.213.102.12
...108fbe28	2.8 hr	GET	500	3.0 sec	http://storefront.us-east-1.elasticbeanst...	18.213.102.12

aws

AWS X-Ray Analytics

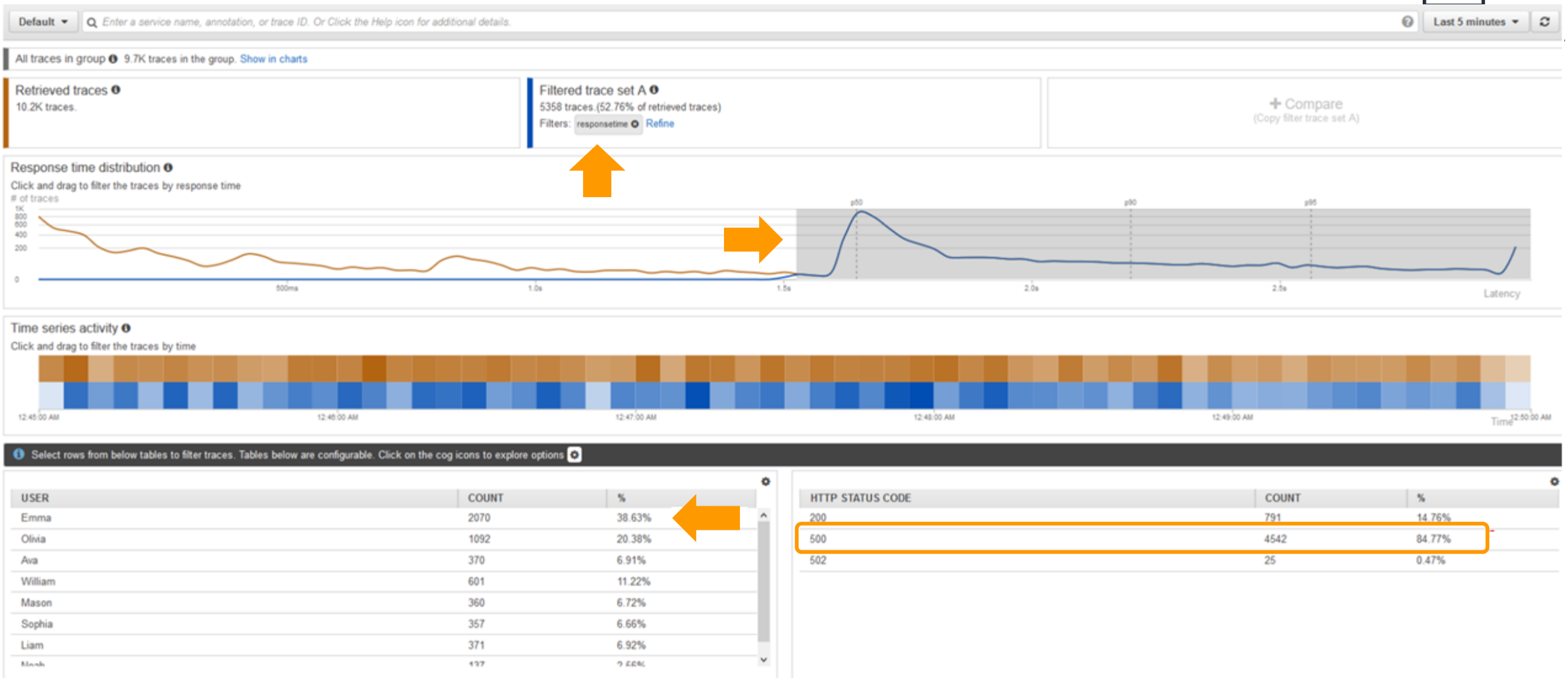
Understand End-User Impact

Analyze Root Cause

Compare Trends

aws

AWS X-Ray Analytics



AWS X-Ray Analytics

Analyze Root Cause

RESPONSE TIME ROOT CAUSE	COUNT	%
api → forward → products.us-east-1.elasticbeanstalk.com ⇒ products → fetch → DynamoDB → wait	1673	36.83%
www → products → store-api.us-east-1.elasticbeanstalk.com ⇒ api → forward → products.us-east-1.elasticbeanstalk.com ⇒ products → fetch → DynamoDB → wait	2566	56.49%
api → forward → products.us-east-1.elasticbeanstalk.com	75	1.65%
www → products → store-api.us-east-1.elasticbeanstalk.com	108	2.38%
www → products → store-api.us-east-1.elasticbeanstalk.com → response	41	0.90%
www → products → store-api.us-east-1.elasticbeanstalk.com ⇒ api → forward → products.us-east-1.elasticbeanstalk.com	2	0.04%
www → products → store-api.us-east-1.elasticbeanstalk.com ⇒ api → forward → products.us-east-1.elasticbeanstalk.com ⇒ products → fetch → DynamoDB	7	0.15%

Trace List

URL	USERS	RESPONSE	RESPONSE TIME
http://store-api.us-east-1.elasticbeanstalk.com/products/	Emma	500	1.729
http://store-api.us-east-1.elasticbeanstalk.com/products/	William	500	1.69
http://storefront.us-east-1.elasticbeanstalk.com/products/	Mason	500	1.757
http://storefront.us-east-1.elasticbeanstalk.com/	Olivia	500	1.665
http://store-api.us-east-1.elasticbeanstalk.com/products/	William	500	1.706

FAULT ROOT CAUSE	COUNT
api → error ⇒ products → error → error → error → error ⇒ products (AWS::DynamoDB::Table)	1297
api → error ⇒ products → error → error → error → error → error ⇒ products (AWS::DynamoDB::Table)	225
www → error → error ⇒ api → error ⇒ products → error → error → error → error ⇒ products (AWS::DynamoDB::Table)	1996

Response Time Root Cause
Error Root Cause
Fault Root Cause

FAULT ROOT CAUSE MESSAGE	COUNT	%
ProvisionedThroughputExceededException: The level of configured provisioned throughput for the table was exceeded. Consider increasing your provisioning level with the UpdateTable API. status code: 4	4437	97.69%
http request to products failed with status code: 500	37	0.81%
http request to api failed with status code: 502	14	0.31%

aws

AWS X-Ray Analytics

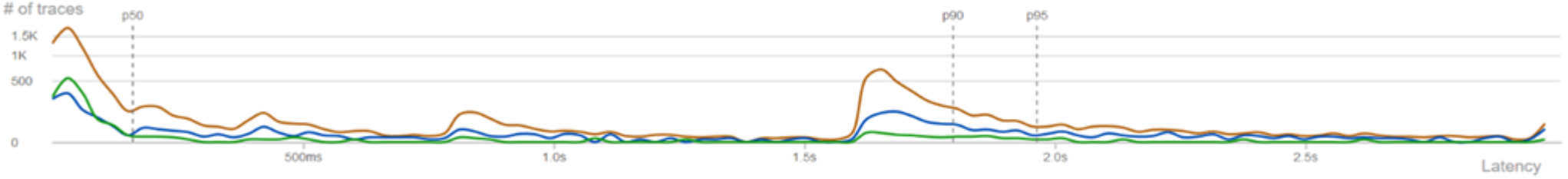
Compare Trends

Retrieved traces
10.3K traces.

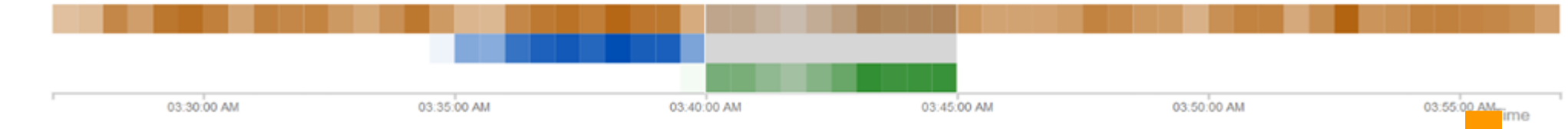
Filtered trace set A
1899 traces.(18.45% of retrieved traces)
Filters: timerange

Filtered trace set B
1354 traces.(13.15% of retrieved traces)
Filters: timerange

Response time distribution
Click and drag to filter the traces by response time



Time series activity
Click and drag to filter the traces by time



USER	COUNT	%	Δ
William	189	13.96%	0.00%
Olivia	279	20.61%	0.17%
Sophia	88	6.50%	0.23%
Mason	98	7.24%	0.76%
Emma	433	31.98%	-0.41%
Noah	74	5.47%	0.62%

HTTP STATUS CODE	COUNT	%	Δ
200	1218	89.96%	32.87%
400	16	1.18%	0.39%
401	22	1.62%	-0.22%
500	97	7.16%	-32.01%
-	1	0.07%	-0.19%

aws

Demo

Thank you!