AWS Well-Architected

Monitoring with CloudWatch Dashboards

AWS Well-Architected

Eric Pullen
Performance Efficiency Lead

Pillars of AWS Well-Architected

Operational Excellence

Security

Reliability

Performance efficiency

Cost optimization

What is Performance Efficiency?

The efficient use of computing resources to meet requirements and how to maintain that efficiency as demand changes and technologies evolve.

Why is it important?

Always having the best resources will give you the greatest outcomes and help increase your innovation and business success.

What should you avoid?

- Not factoring in operational and opportunity costs
- Separation between the business and technology

What are your peers doing?

- Implementing on a continual cycle
- Removing the burden of technology

Poll #1: When new instance types are released, how quickly do you integrate them into your workloads?

a. Immediately

b. 1-3 Months

c. 3-6 Months

d. 6-12 Months

Performance Efficiency Design Principles

- Democratize advanced technologies
- Go global in minutes
- Use serverless architectures
- Experiment more often
- Mechanical sympathy

Deming Cycle (PDCA)

(start)

Plan – what achieve?	Reduce response time
Do – execute the plan	Update CloudFormation
Check – study the results	Was it a positive improvement?
Act – make improvements	Merge into mainline

(repeat)

Poll #2: Which area do you perform the most monitoring?

a. CPU/Memory/Network

b. Technical KPI’s

c. Business KPI’s

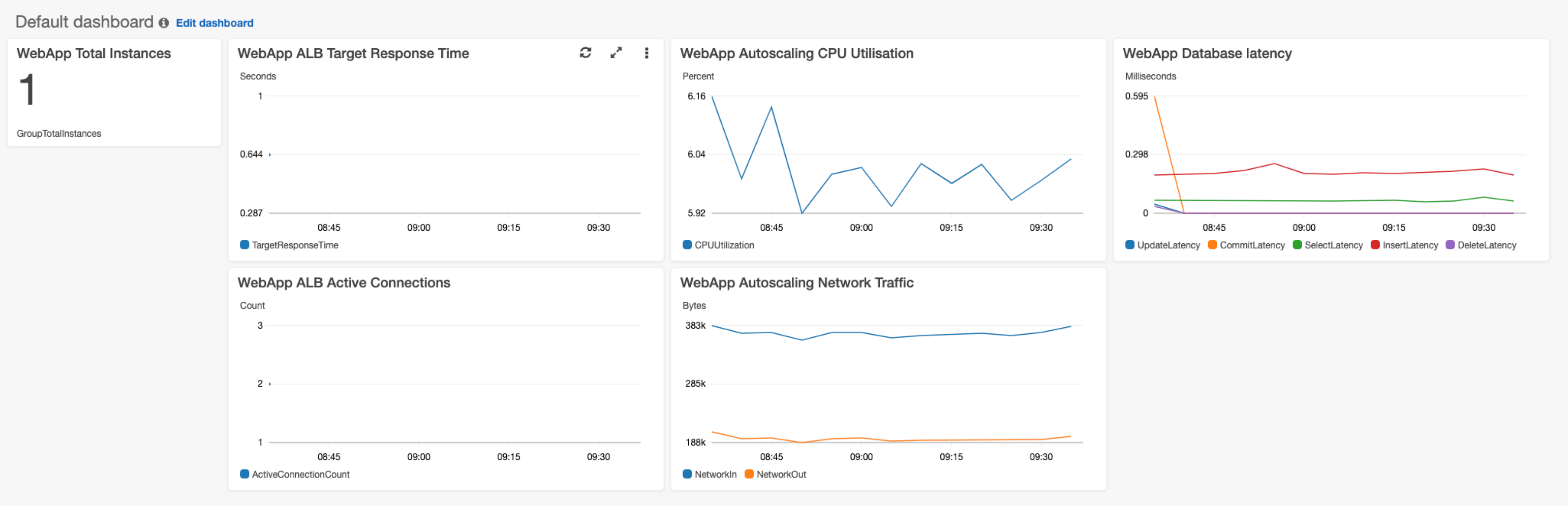
d. Customer Experience Monitoring

Hands on Lab

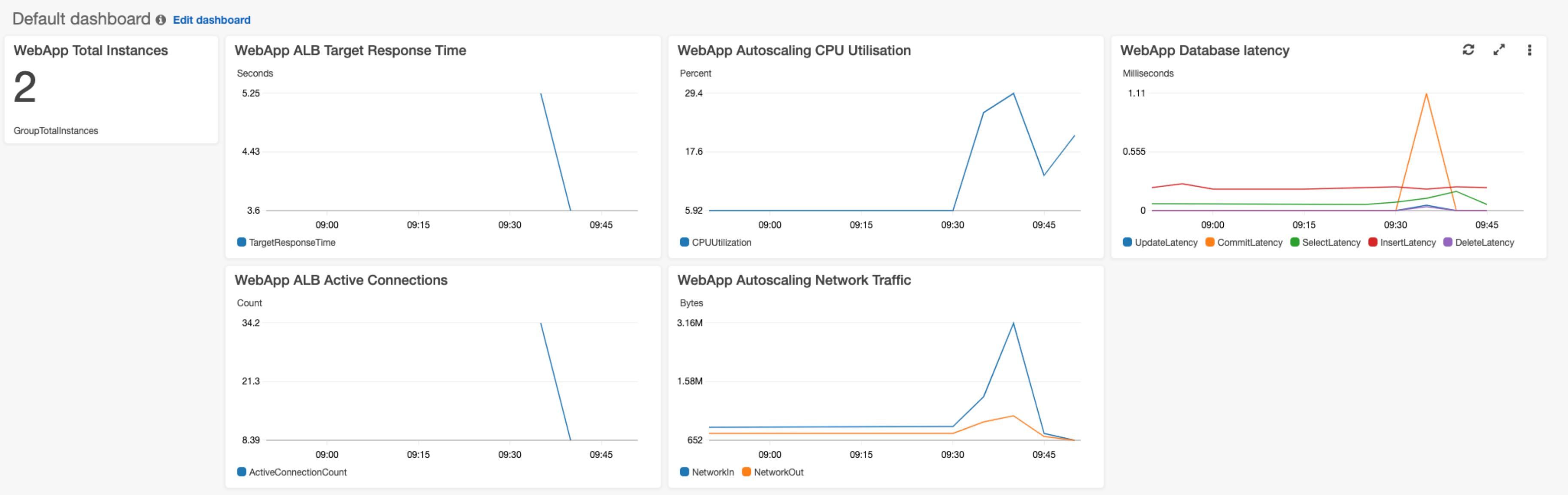
Monitoring with CloudWatch Dashboards

- https://bit.ly/2GMDVkN

Hands On Lab


Default dashboard
Edit dashboard
WebApp Total Instances
1
GroupTotalInstances
WebApp ALB Target Response Time
Seconds
1
0.644
0.287
08:45
09:00
09:15
09:30
TargetResponseTime
WebApp Autoscaling CPU Utilisation
Percent
6.16
6.04
5.92
08:45
09:00
09:15
09:30
CPUUtilization
WebApp Database latency
Milliseconds
0.595
0.298
0
08:45
09:00
09:15
09:30
UpdateLatency
CommitLatency
SelectLatency
InsertLatency
DeleteLatency
WebApp ALB Active Connections
Count
3
2
1
08:45
09:00
09:15
09:30
ActiveConnectionCount
WebApp Autoscaling Network Traffic
Bytes
383k
285k
188k
08:45
09:00
09:15
09:30
NetworkIn
NetworkOut

Hands On Lab


Default dashboard
Edit dashboard
WebApp Total Instances
2
GroupTotalInstances
WebApp ALB Target Response Time
Seconds
5.25
4.43
3.6
09:00
09:15
09:30
09:45
TargetResponseTime
WebApp Autoscaling CPU Utilisation
Percent
29.4
17.6
5.92
09:00
09:15
09:30
09:45
CPUUtilization
WebApp Database latency
Milliseconds
1.11
0.555
0
09:00
09:15
09:30
09:45
UpdateLatency
CommitLatency
SelectLatency
InsertLatency
DeleteLatency
WebApp ALB Active Connections
Count
34.2
21.3
8.39
09:00
09:15
09:30
09:45
ActiveConnectionCount
WebApp Autoscaling Network Traffic
Bytes
3.16M
1.58M
652
09:00
09:15
09:30
09:45
NetworkIn
NetworkOut

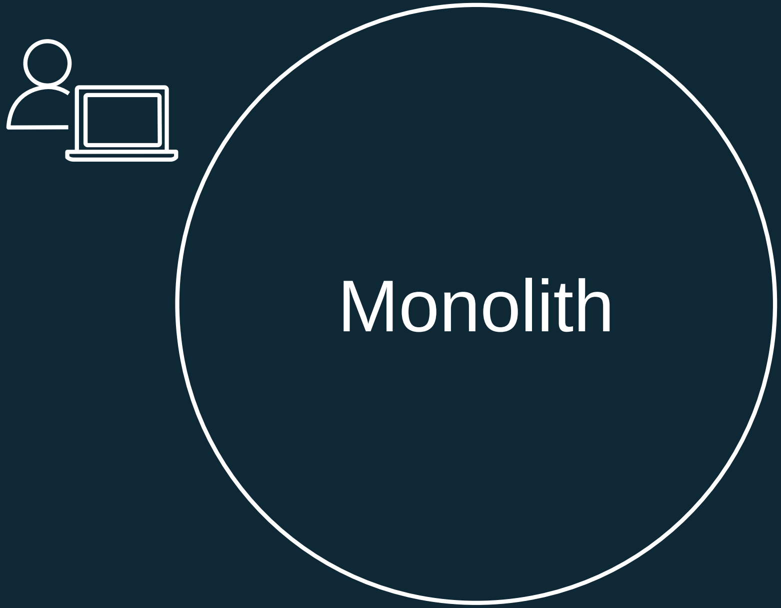
AWS Well-Architected

Observability

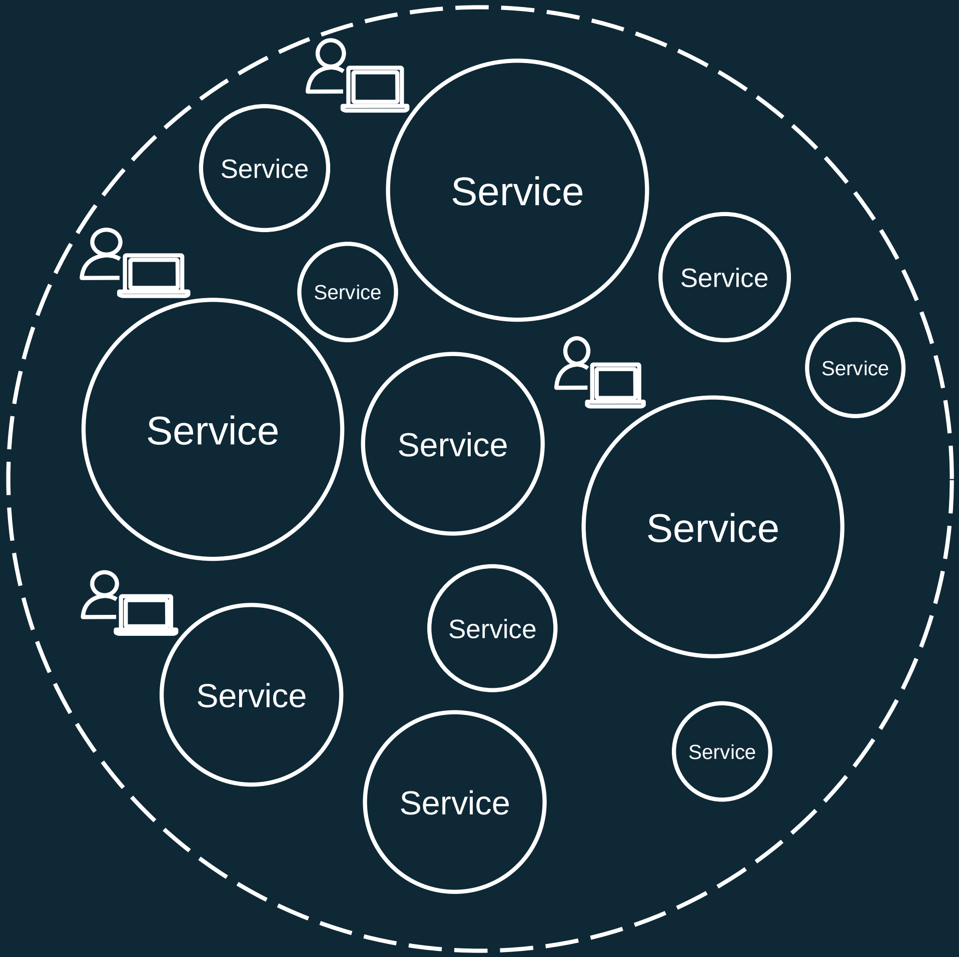
Microservices increase release agility

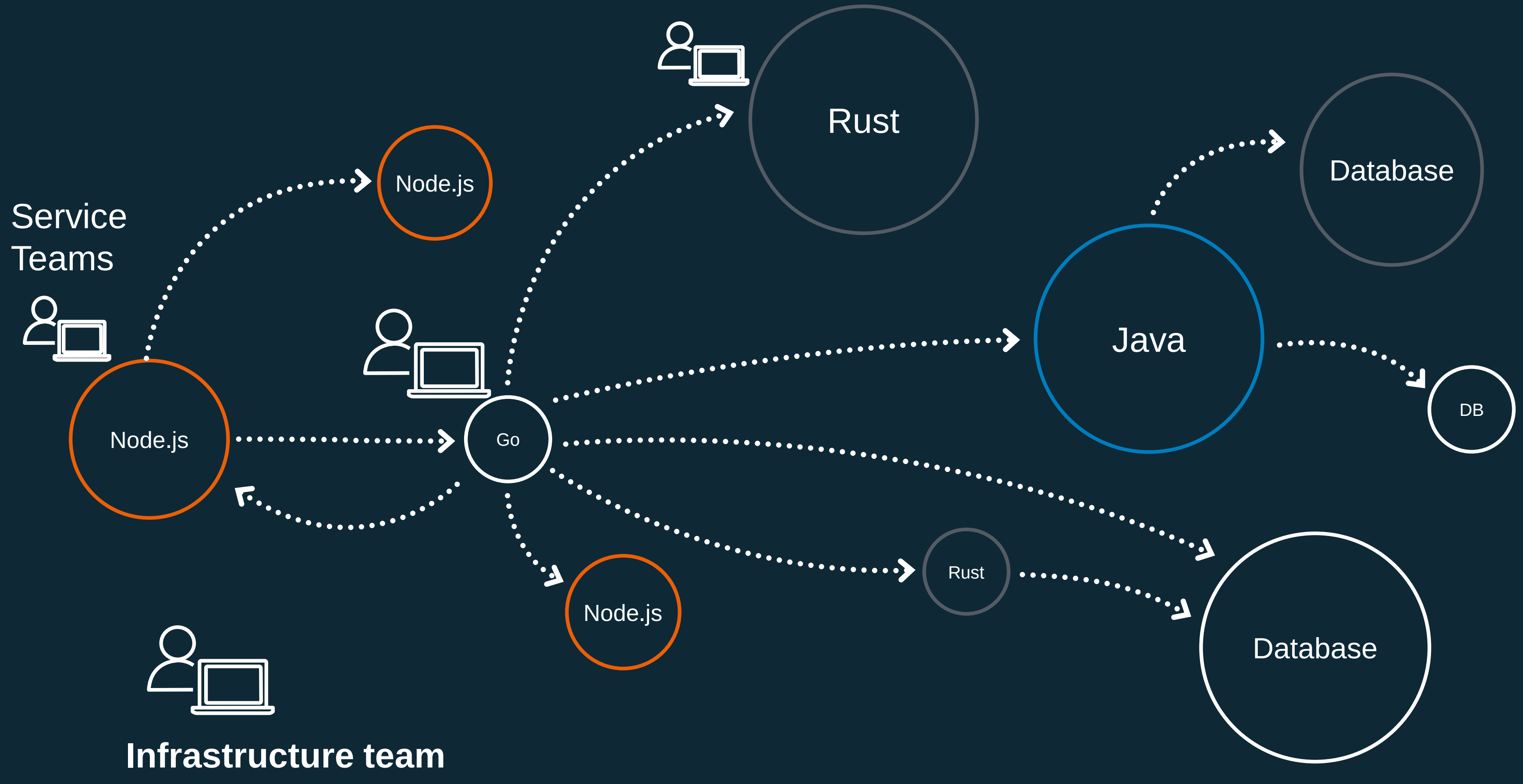
Monolithic application

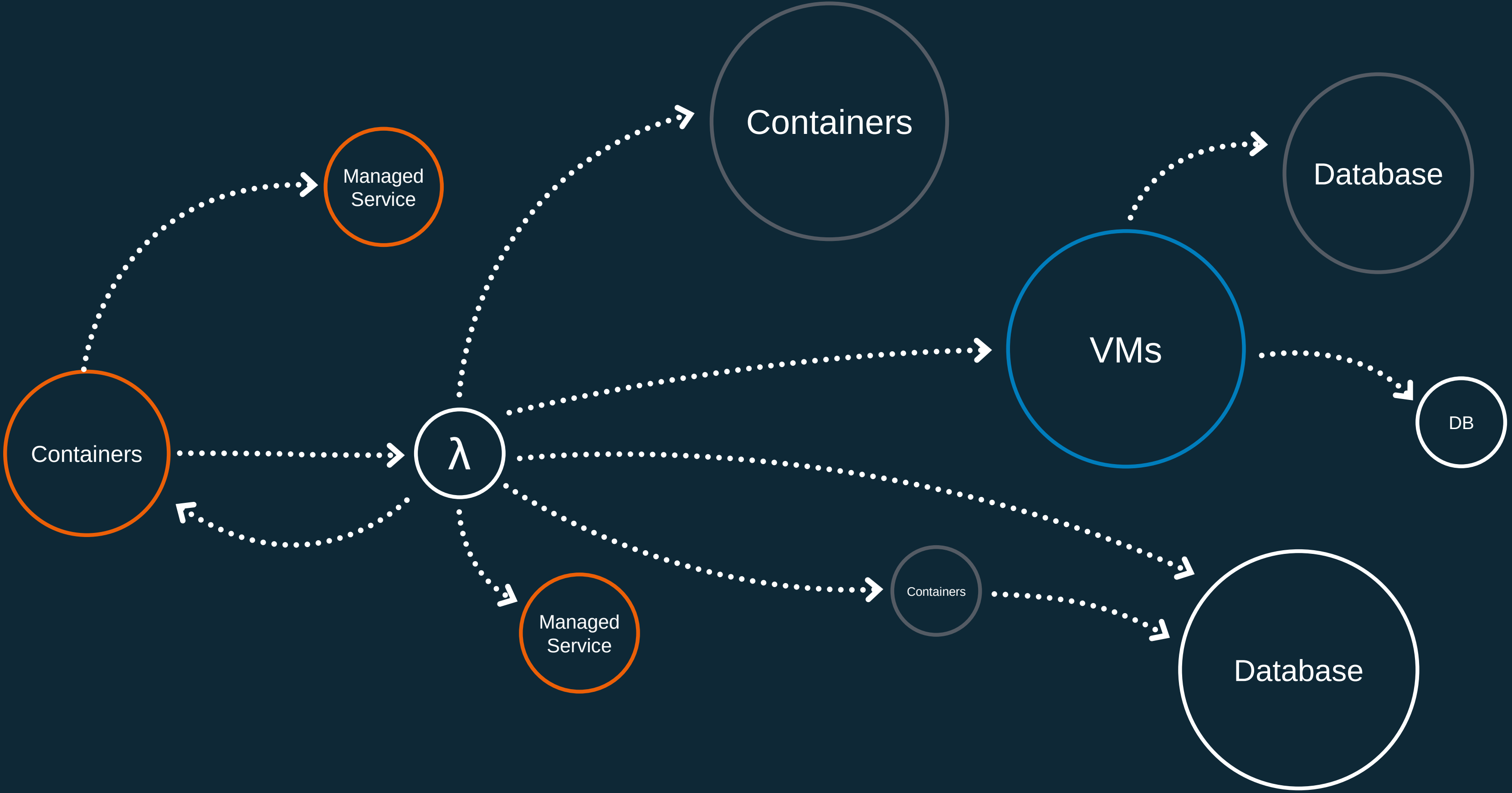
Microservices



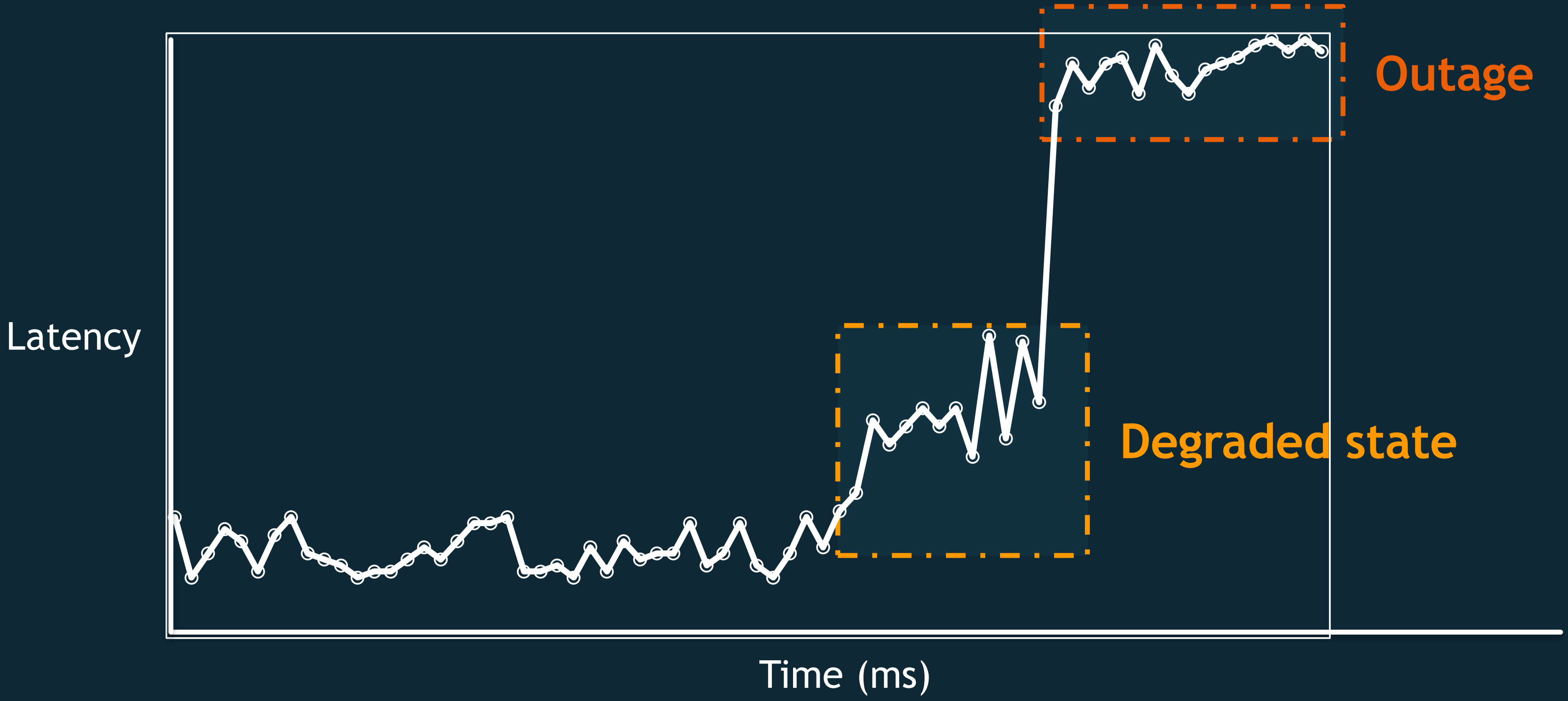
Monolith


Service
Service
Service
Service
Service
Service
Service
Service
Service
Service
Service
Service
Service
Service

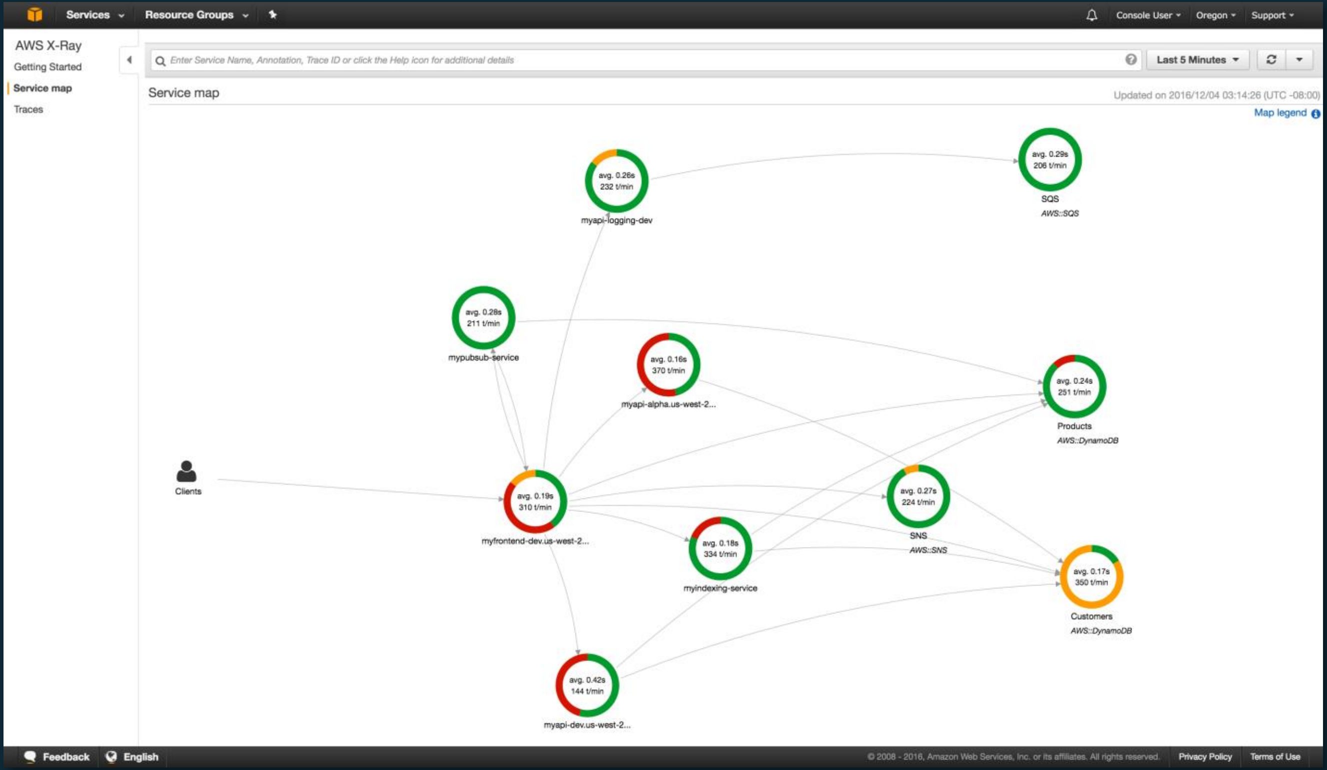

Service
Teams
Node.js
Node.js
Go
Node.js
Rust
Rust
Java
Database
DB
Database
Infrastructure team


Managed Service
Containers
Database
VMs
DB
Containers
λ
Managed Service
Containers
Database

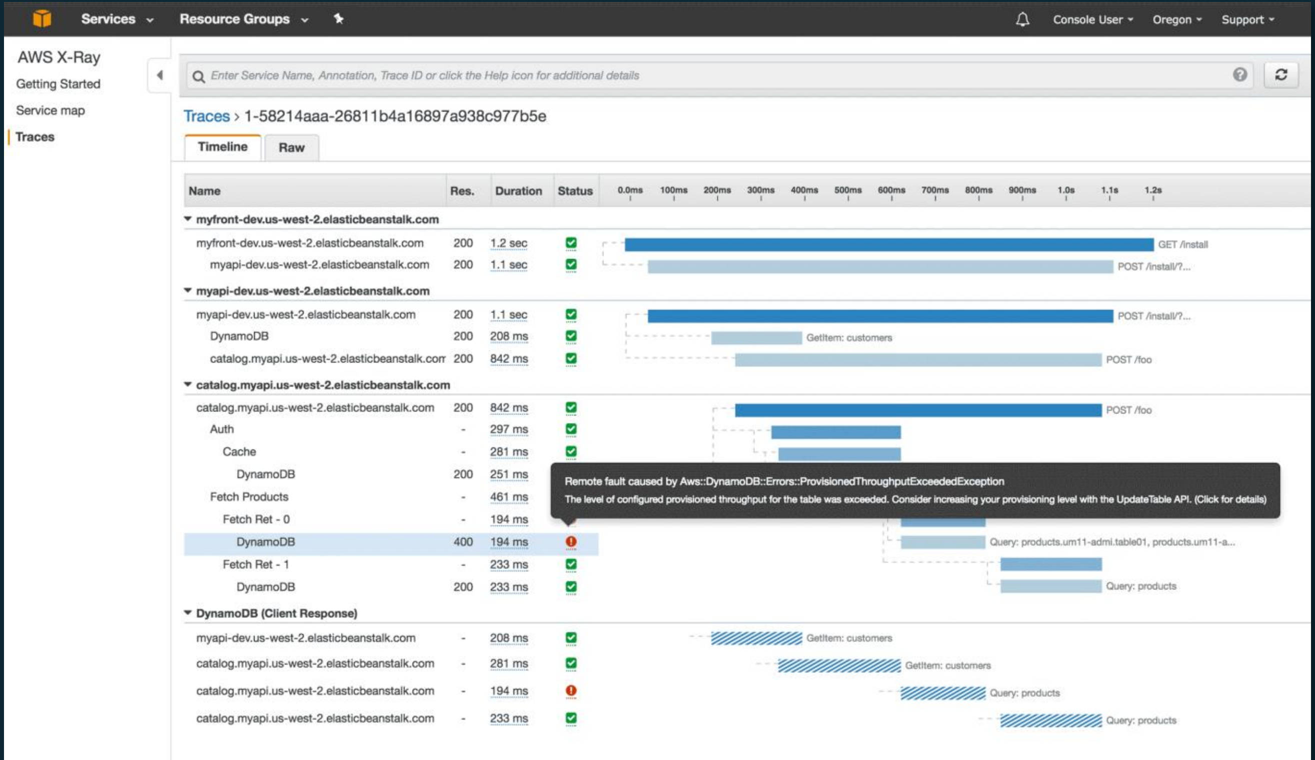
Proactive operations helps mitigate issues



End-to-end tracing – AWS X-Ray Service Map



End-to-end tracing – AWS X-Ray Traces



Traces › 1-58214aaa-26811b4a16897a938c977b5e

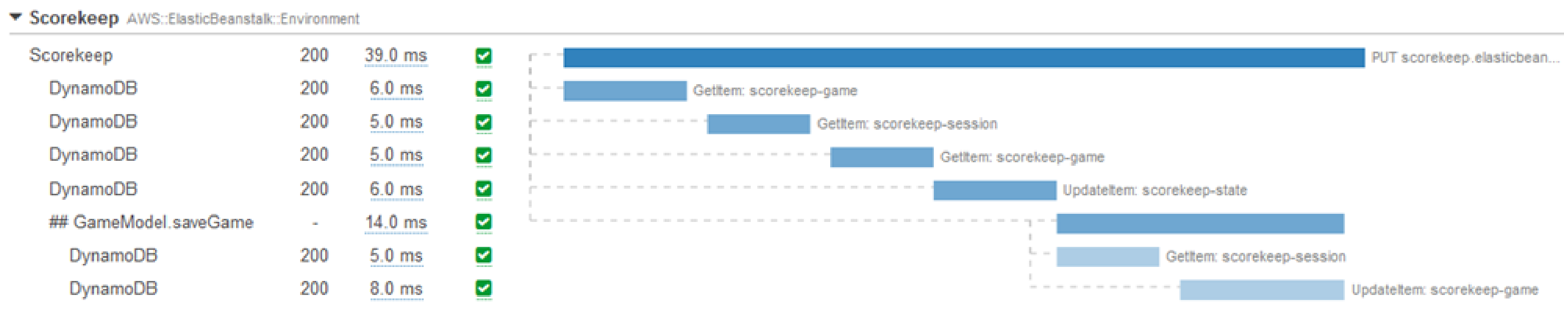
Name	Res.	Duration	Status	
myfront-dev.us-west-2.elasticbeanstalk.com				
myfront-dev.us-west-2.elasticbeanstalk.com	200	1.2 sec	✓	GET /install
myapi-dev.us-west-2.elasticbeanstalk.com	200	1.1 sec	✓	POST /install/?...
myapi-dev.us-west-2.elasticbeanstalk.com				
myapi-dev.us-west-2.elasticbeanstalk.com	200	1.1 sec	✓	POST /install/?...
DynamoDB	200	208 ms	✓	GetItem: customers
catalog.myapi.us-west-2.elasticbeanstalk.com	200	842 ms	✓	POST /foo
catalog.myapi.us-west-2.elasticbeanstalk.com				
catalog.myapi.us-west-2.elasticbeanstalk.com	200	842 ms	✓	POST /foo
Auth	-	297 ms	✓	
Cache	-	281 ms	✓	
DynamoDB	200	251 ms		
Fetch Products	-	461 ms		
Fetch Ret - 0	-	194 ms		
DynamoDB	400	194 ms	!	Query: products.um11-admi.table01, products.um11-a...
Fetch Ret - 1	-	233 ms	✓	
DynamoDB	200	233 ms	✓	Query: products
DynamoDB (Client Response)				
myapi-dev.us-west-2.elasticbeanstalk.com	-	208 ms	✓	GetItem: customers
catalog.myapi.us-west-2.elasticbeanstalk.com	-	281 ms	✓	GetItem: customers
catalog.myapi.us-west-2.elasticbeanstalk.com	-	194 ms	!	Query: products
catalog.myapi.us-west-2.elasticbeanstalk.com	-	233 ms	✓	Query: products

Remote fault caused by Aws::DynamoDB::Errors::ProvisionedThroughputExceededException

The level of configured provisioned throughput for the table was exceeded. Consider increasing your provisioning level with the UpdateTable API. (Click for details)

AWS X-Ray Key Concepts

Segments



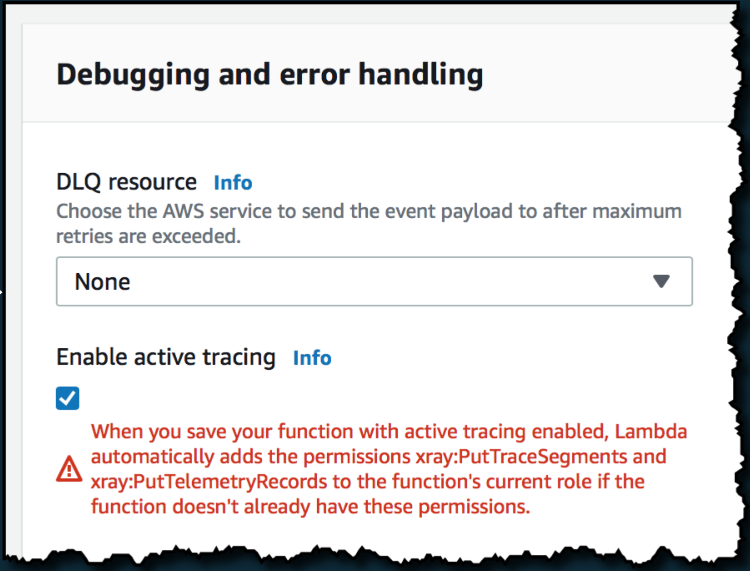
▼ **Scorekeep** AWS::ElasticBeanstalk::Environment

Scorekeep	200	39.0 ms	✓	PUT scorekeep.elasticbean...
DynamoDB	200	6.0 ms	✓	GetItem: scorekeep-game
DynamoDB	200	5.0 ms	✓	GetItem: scorekeep-session
DynamoDB	200	5.0 ms	✓	GetItem: scorekeep-game
DynamoDB	200	6.0 ms	✓	UpdateItem: scorekeep-state
## GameModel.saveGame	-	14.0 ms	✓	
DynamoDB	200	5.0 ms	✓	GetItem: scorekeep-session
DynamoDB	200	8.0 ms	✓	UpdateItem: scorekeep-game

Subsegments

Enabling X-Ray tracing

AWS Lambda Console



Debugging and error handling

DLQ resource **Info**

Choose the AWS service to send the event payload to after maximum retries are exceeded.

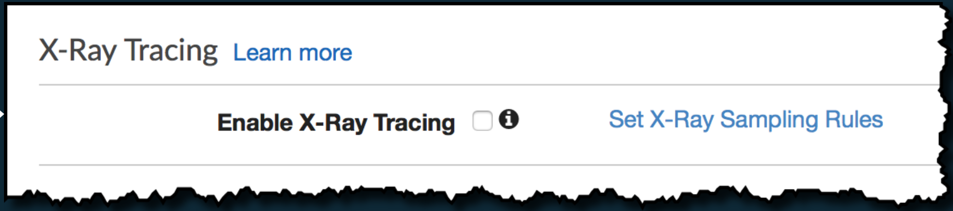
None ▼

Enable active tracing **Info**

☑

⚠ When you save your function with active tracing enabled, Lambda automatically adds the permissions xray:PutTraceSegments and xray:PutTelemetryRecords to the function's current role if the function doesn't already have these permissions.

Amazon API Gateway Console



X-Ray Tracing Learn more

Enable X-Ray Tracing □ ℹ Set X-Ray Sampling Rules

Enabling X-Ray tracing in your code

```
const AWS = require('aws-sdk');
```

↓

```
const AWSXRay = require('aws-xray-sdk');
const AWS = AWSXRay.captureAWS(require('aws-sdk'));
```

AWS X-Ray

Lab 2: http://bit.ly/EBSampleApp

Poll #3: Are you currently monitoring end-to-end requests via a tracing system?

a. Yes

b. No

c. Unsure

AWS Well-Architected

Performance Efficiency Qs

1. How do you select the best performing architecture?

- Understand the available services and resources
- Define a process for architectural choices
- Factor cost or budget into decisions
- Use policies or reference architectures
- Use guidance from AWS or an APN Partner
- Benchmark existing workloads
- Load test your workload

2. How do you select your compute solution?

- Evaluate the available compute options
- Understand the available compute configuration options
- Collect compute-related metrics
- Determine the required configuration by right-sizing
- Use the available elasticity of resources
- Re-evaluate compute needs based on metrics

3. How do you select your storage solution?

- Understand storage characteristics and requirements
- Evaluate available configuration options
- Make decisions based on access patterns and metrics

4. How do you select your database solution?

- Understand data characteristics
- Evaluate the available options
- Collect and record database performance metrics
- Choose data storage based on access patterns
- Optimize data storage based on access patterns and metrics

5. How do you configure your networking solution?

- Understand how networking impacts performance
- Understand available product options
- Evaluate available networking features
- Use minimal network ACLs
- Leverage encryption offloading and load-balancing
- Choose network protocols to improve performance
- Choose location based on network requirements
- Optimize network configuration based on metrics

6. How do you evolve your workload to take advantage of new releases?

- Keep up-to date on new resources and services
- Define a process to improve workload performance
- Evolve workload performance over time

7. How do you monitor your resources to ensure they are performing as expected?

- Record performance-related metrics
- Analyze metrics when events or incidents occur
- Establish KPIs to measure workload performance
- Use monitoring to generate alarm-based notifications
- Review metrics at regular intervals
- Monitor and alarm proactively

8. How do you use tradeoffs to improve performance?

- Understand the areas where performance is most critical
- Learn about design patterns and services
- Identify how tradeoffs impact customers and efficiency
- Measure the impact of performance improvements
- Use various performance-related strategies

Poll #4: What area of performance efficiency are you most interested in a deeper dive?

a. Selection

b. Review

c. Monitoring

d. Trade-offs

Thank you!

https://aws.amazon.com/architecture/well-architected/