



Monitoring and Observability on AWS

Sergey Kurson
Solutions Architect, AWS
kursonsk@amazon.com



Quickly about me



Sergey Kurson
Solutions Architect, AWS
kursonsk@amazon.com



Agenda

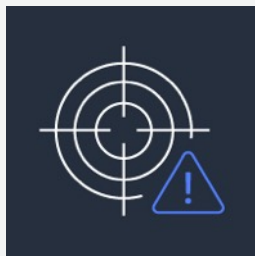
1. General concepts
2. Solution demo
3. What is Cloudwatch today?
4. Metrics and monitoring tools on AWS today with demo
5. What is observability
6. Observability tools on AWS today with demo



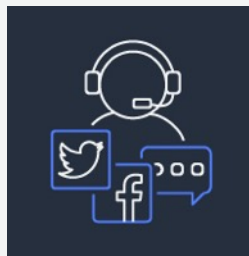
Observability matters because ...



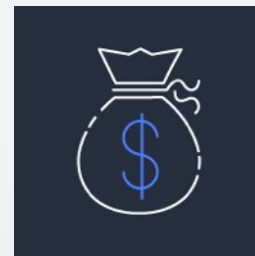
Visibility



Real-time
troubleshooting



#Customer
experience



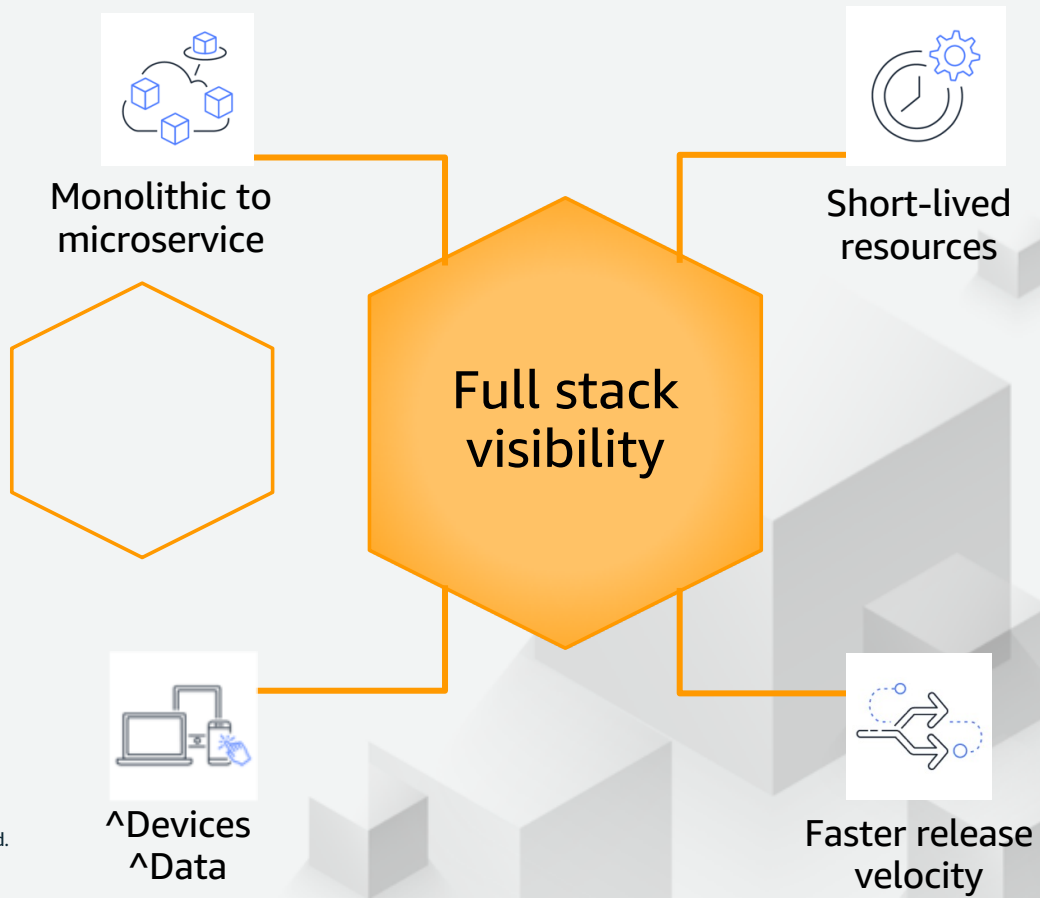
Applications
= \$\$

Operational

Business

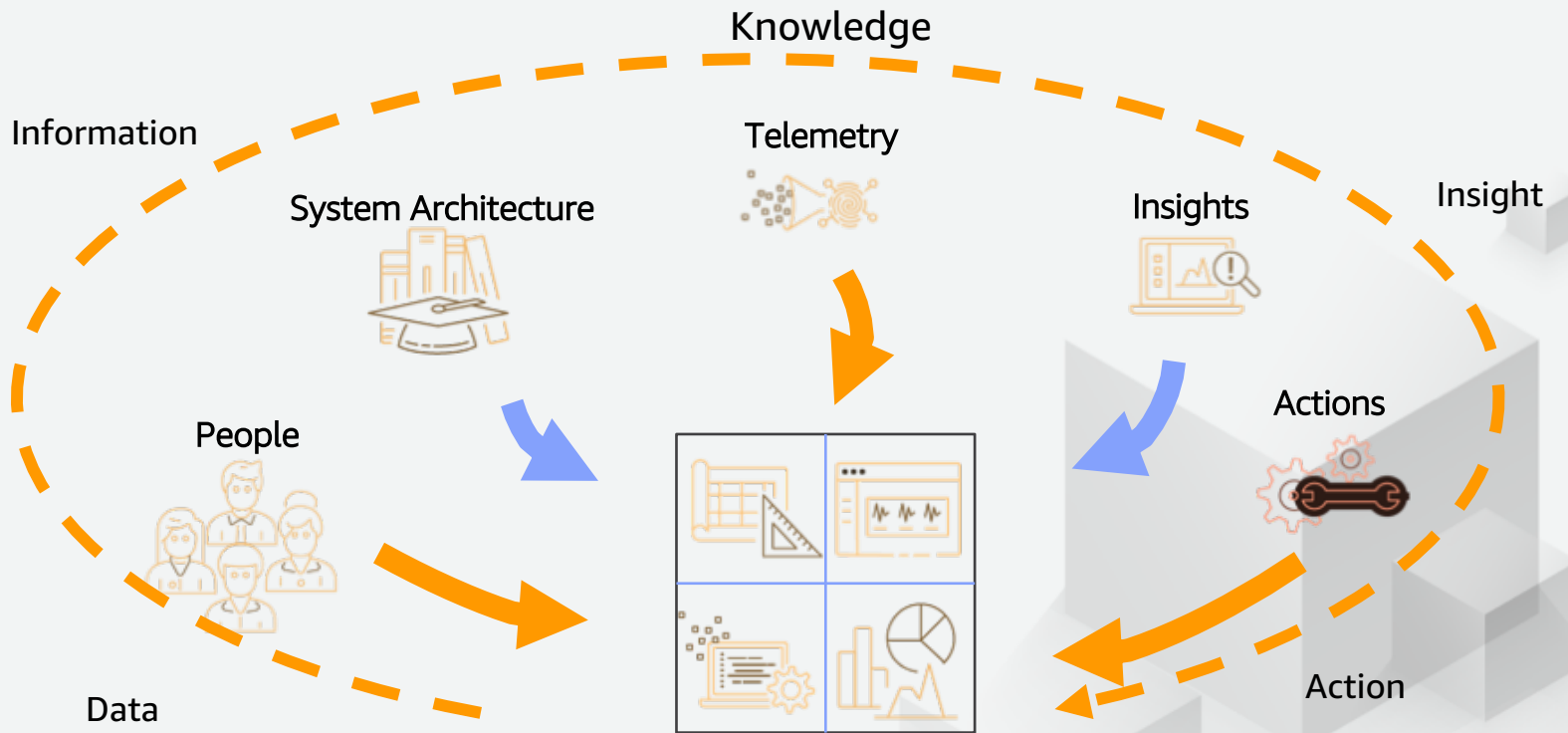


Monitoring must evolve



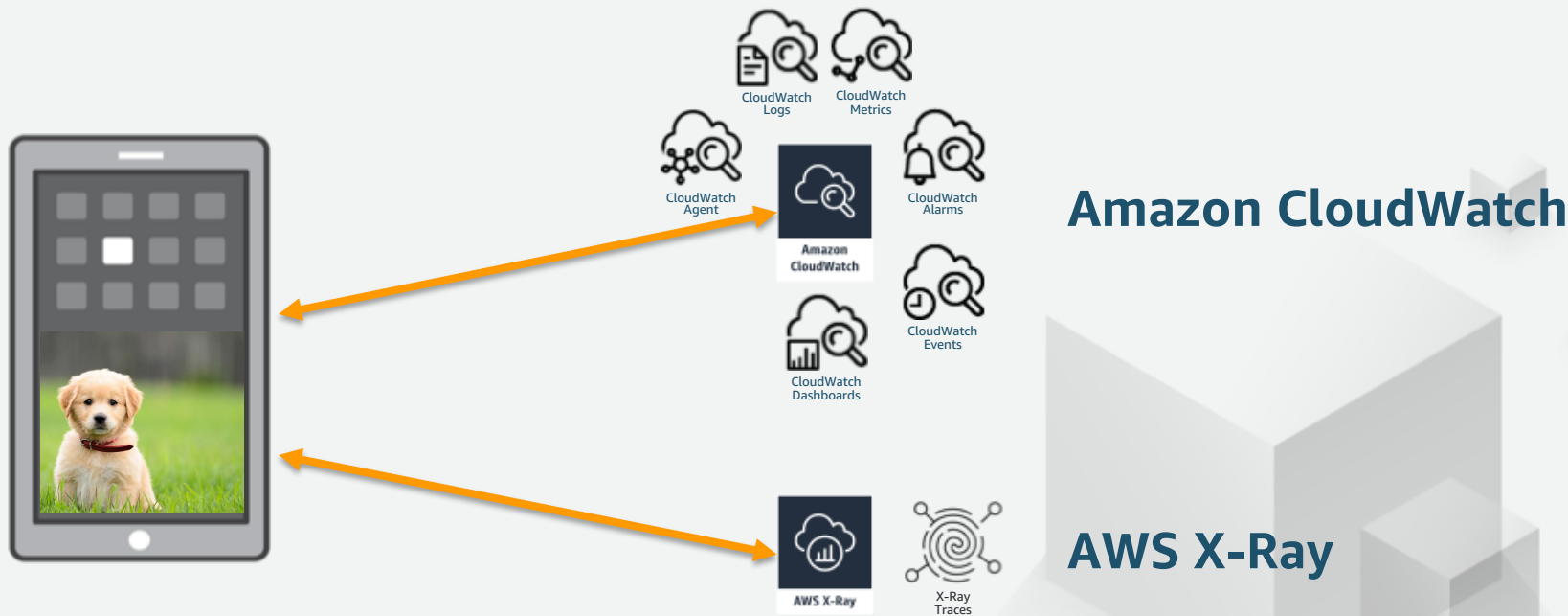


What Goes Into an Observability Plan?



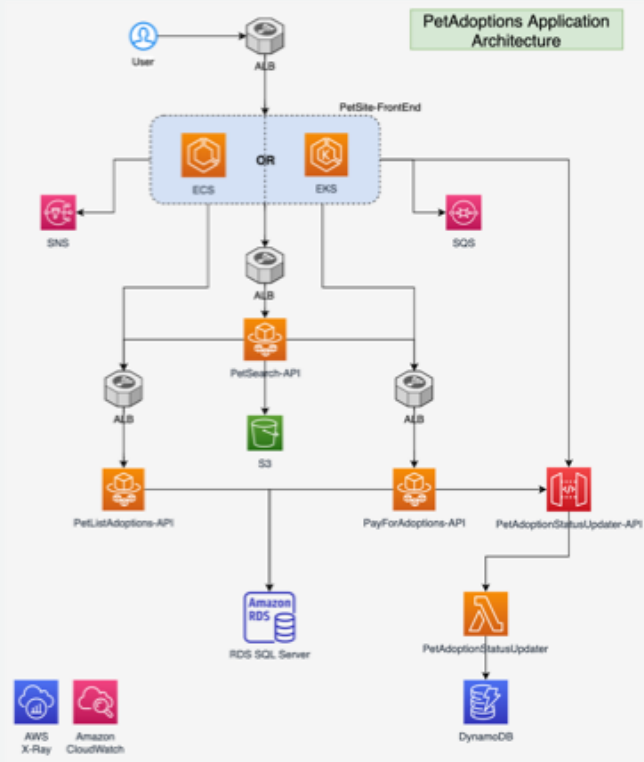


App example: Monitoring and Observability





App example architecture





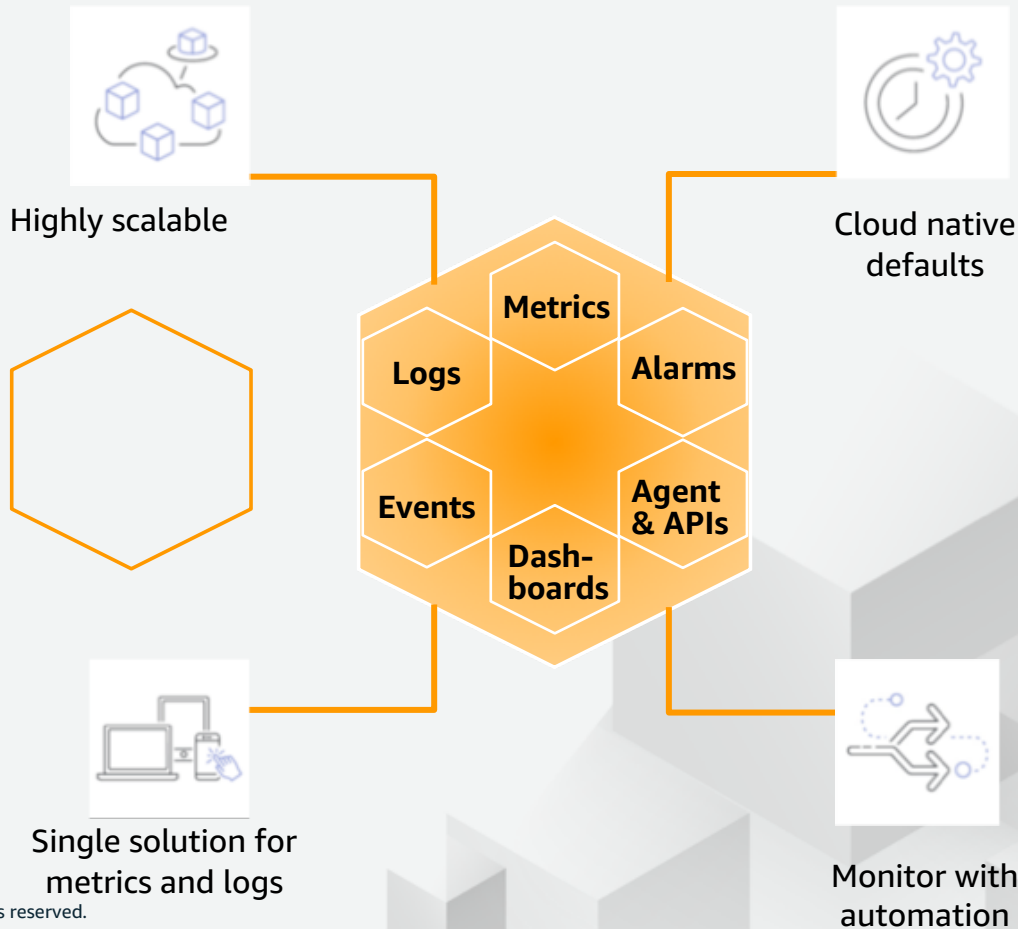
Solution demo



What is Cloudwatch today?

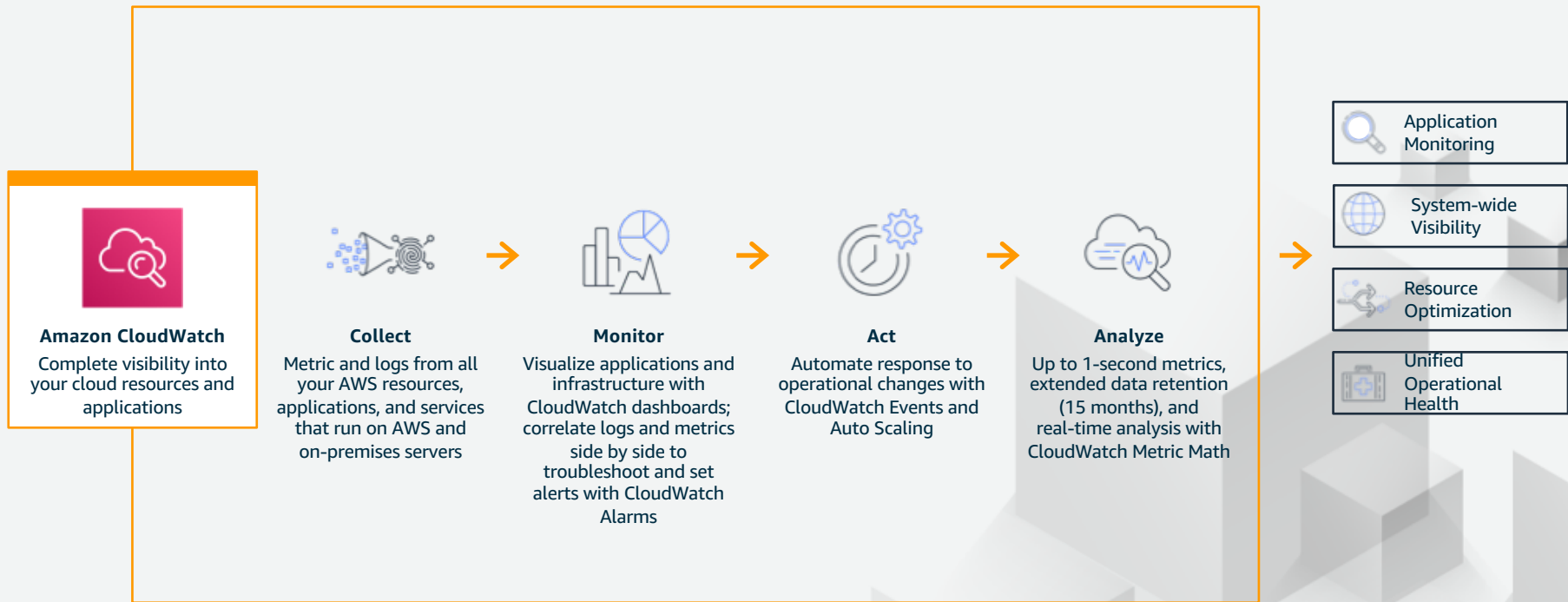


Amazon Cloudwatch





How it works





Interacting with Amazon Cloudwatch



AWS Management Console



AWS CLI



Amazon CloudWatch API



AWS SDKs

ISVs using CloudWatch



splunk >

dynatrace

solarwinds
papertrail

sumologic



pagerduty



Dramathaus



VictorOps



CloudHealth
Technologies



Collect



Collecting via Cloudwatch Agent



Collect EC2/ECS/EKS

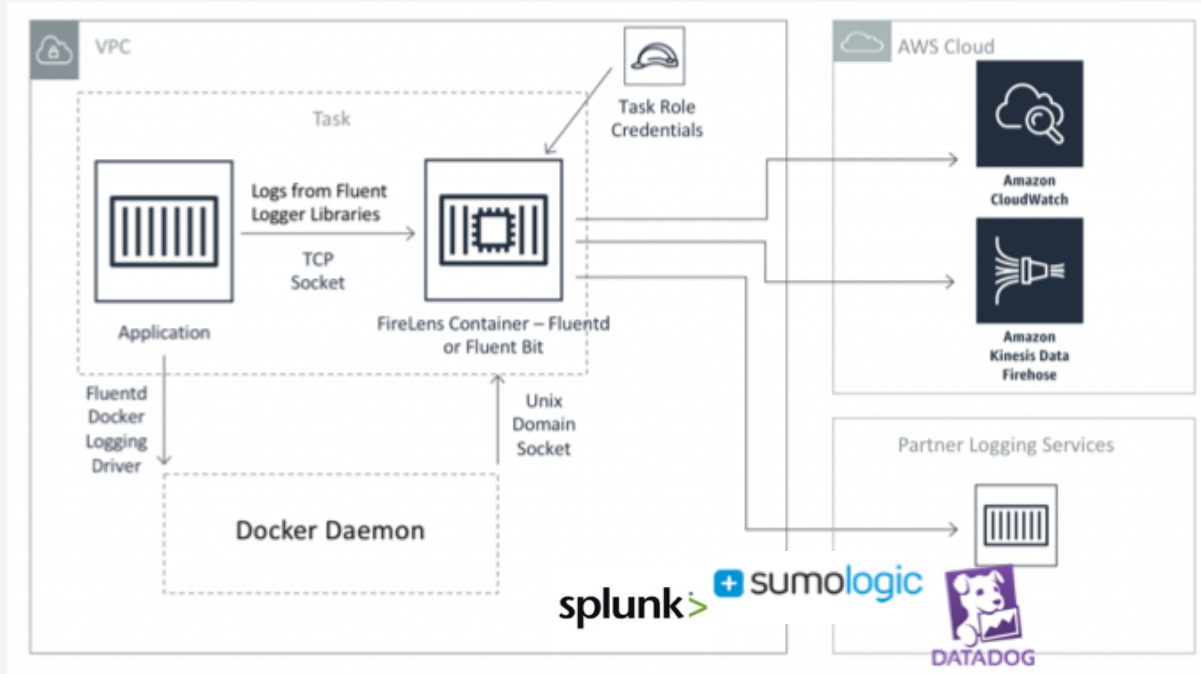
```
{
  "metrics": {
    "aggregation_dimensions": [ ["AutoScalingGroupName", "InstanceId"], ["AutoScalingGroupName"] ],
    "append_dimensions": { "InstanceId": "${aws:InstanceId}", "AutoScalingGroupName": "${aws:AutoScalingGroupName}" },
    "metrics_collected": {
      "mem": { "measurement": ["mem_used", "mem_cached", "mem_used_percent", "mem_available_percent"] },
      "processes": { "measurement": ["running", "sleeping", "dead"] },
      "disk": {"resources": ["/"], "measurement": ["free", "used_percent"] },
    },
    "namespace": "live-broadcast-red-button-linkmanager-api"
  },
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/var/log/broadcast-red-button-linkmanager-api/output.log",
            "log_group_name": "live-broadcast-red-button-linkmanager-api-infrastructure-ApplicationLog-J8FGOWKDFOE8",
            "log_stream_name": "${instance_id}-${ip_address}-output.log"
          }
        ]
      }
    },
    "log_stream_name": "${instance_id}-${hostname}"
  },
  "agent": { "logfile": "/var/log/amazon-cloudwatch-agent/amazon-cloudwatch-agent.log", "metrics_collection_interval": 60 }
}
```



Firelens



Collect ECS



Source: <https://aws.amazon.com/blogs/containers/under-the-hood-firelens-for-amazon-ecs-tasks/>



Firelens: Interface



Collect EC2/ECS/EKS

//App container

```
"logConfiguration": {  
  "logDriver": "awsfirelens",  
  "options": {  
    "Name": "datadog | sumologic | splunk | loggly | Kinesis Firehose | Kinesis Data Streams| CloudWatch",  
    "apiKey": "<API_KEY>",  
  }  
  "secretOptions": [{ "name": "apiKey",  
    "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-AbCdEf" }]  
}
```

//FluentBit sidecar

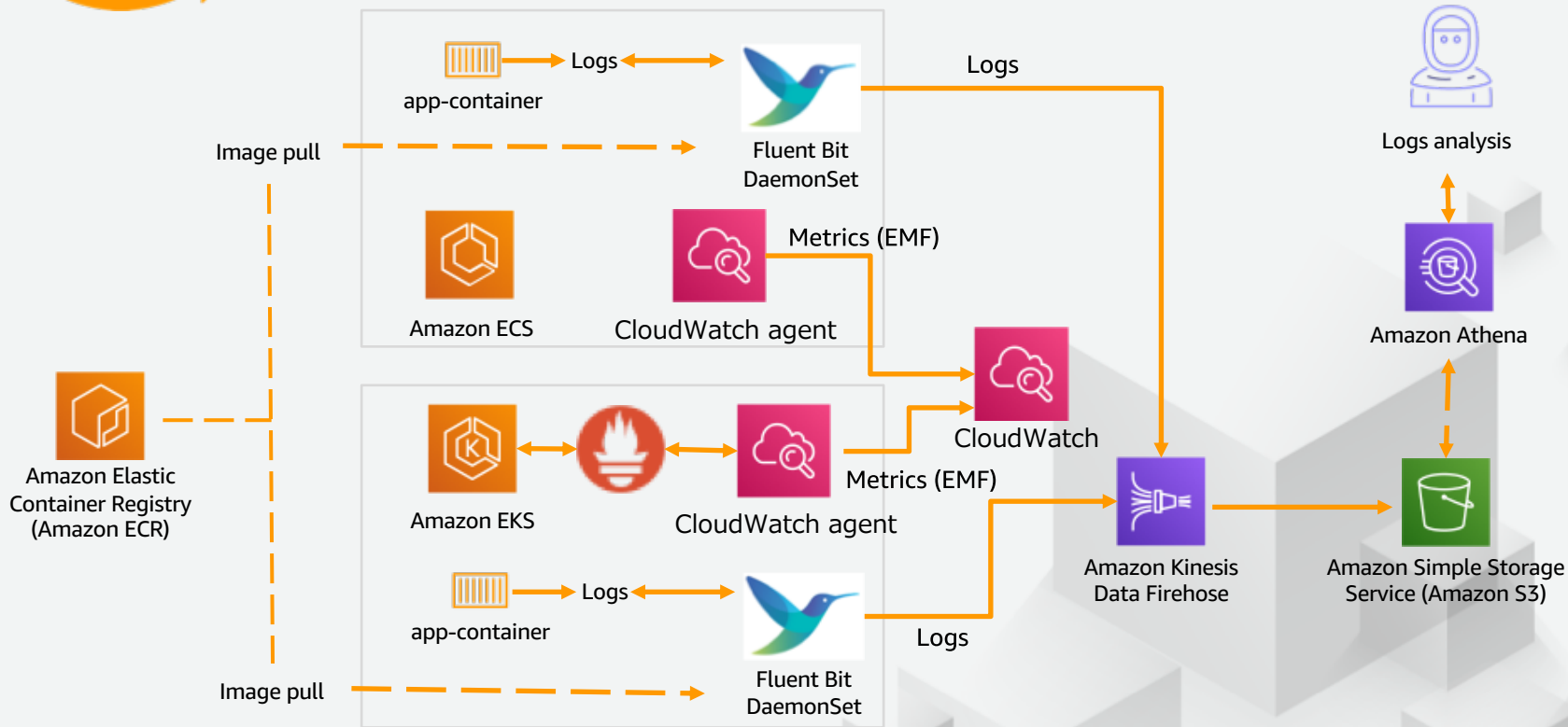
```
{  
  "essential": true,  
  "image": "amazon/aws-for-fluent-bit:latest",  
  "name": "log_router",  
  "firelensConfiguration": { "type": "fluentbit", "options": { "enable-ecs-log-metadata": "true" } }  
}
```



How to consolidate?



Collect ECS/EKS





Embedded Metric Format (EMF)



Collect EC2/ECS/EKS

Embedded Metric Format that can send properties with metric values is now available in CloudWatch Logs

When sending log data to CloudWatch Logs with EMF (JSON), custom metrics are issued and graphed

You can query collected logs with CloudWatch Logs Insights and graph and analyze the properties affecting them in Contributor Insights

Output EMF format log with your own code, or easily output with Python / Node.js EMF library or CloudWatch Agent

```
{
  "_aws": {
    "Timestamp": 1574109732004,
    "CloudWatchMetrics": [
      {
        "Namespace": "lambda-function-metrics",
        "Dimensions": [["functionVersion"]],
        "Metrics": [
          {
            "Name": "time",
            "Unit": "Milliseconds"
          }
        ]
      }
    ]
  },
  "functionVersion": "$LATEST",
  "time": 100,
  "requestId": "989ffbfb8-9ace-4817-a57c-e4dd734019ee"
}
```

EMF log example

Python logging example

```
@metric_scope
def my_handler(metrics):
    metrics.put_dimensions({"Foo": "Bar"})
    metrics.put_metric("ProcessingLatency", 100, "Milliseconds")
    metrics.set_property("AccountId", "123456789012")
    metrics.set_property("RequestId", "422b1569-16f6-4a03")
    metrics.set_property("DeviceId", "61270781-c6ac-46f1")



    return {"message": "Hello!"}
```



Collecting metrics from log extraction



Collect Cloudwatch logs



Filter Name: live-broadcast-monitoring-backend-metrics-FilterAbstractorInputProcessingTime-9VKBTZAGJ4UP [Create Alarm](#)  

Filter Pattern: { \$.times.absInPut = * }

Metric: [BBC/Red Button/Monitoring Backend/e2emon](#) / [live-E2emonAbstractorInputTime](#)

Metric Value: \$.times.absInPut

Default Value: none

Filter Name: live-broadcast-monitoring-backend-metrics-FilterDCABProcessingTime-LMT9LC2DMO6L [Create Alarm](#)  

Filter Pattern: { \$.times.Liberate = * }

Metric: [BBC/Red Button/Monitoring Backend/e2emon](#) / [live-E2emonDCABTime](#)

Metric Value: \$.times.Liberate

Default Value: none



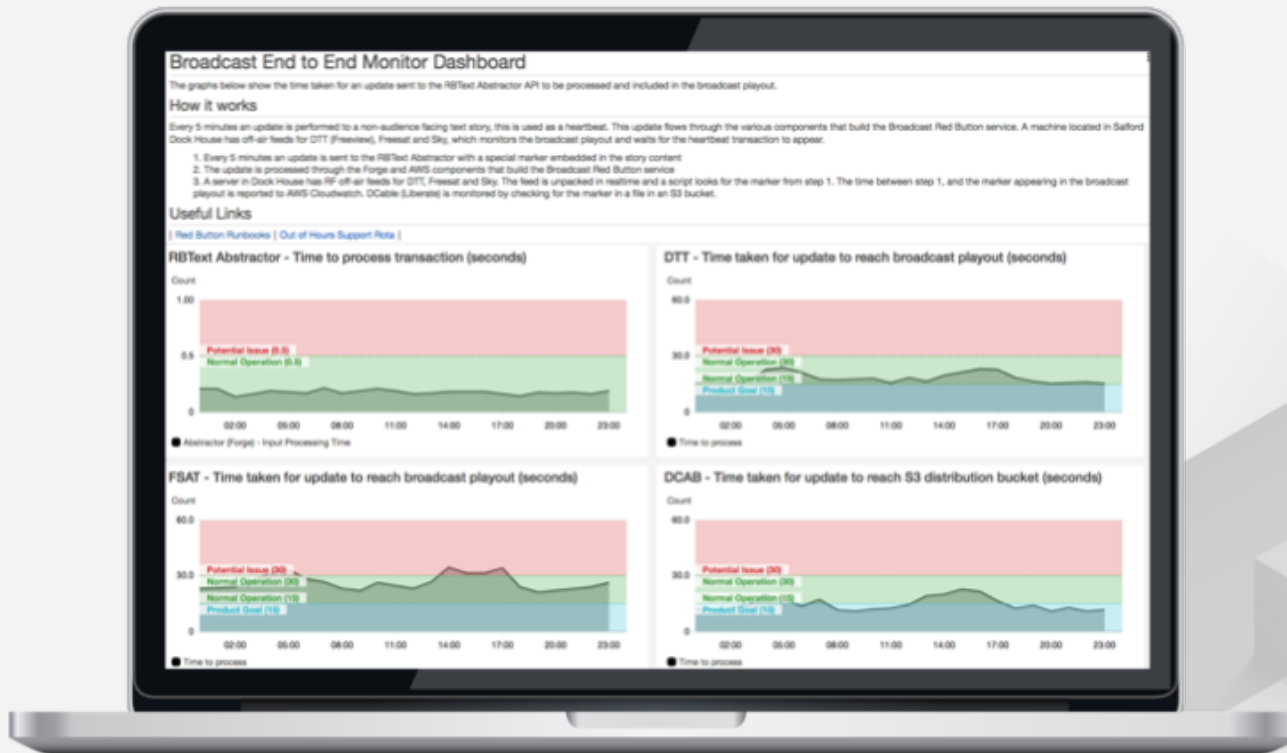
Monitor



Monitoring view – typical day



Monitor Dashboards

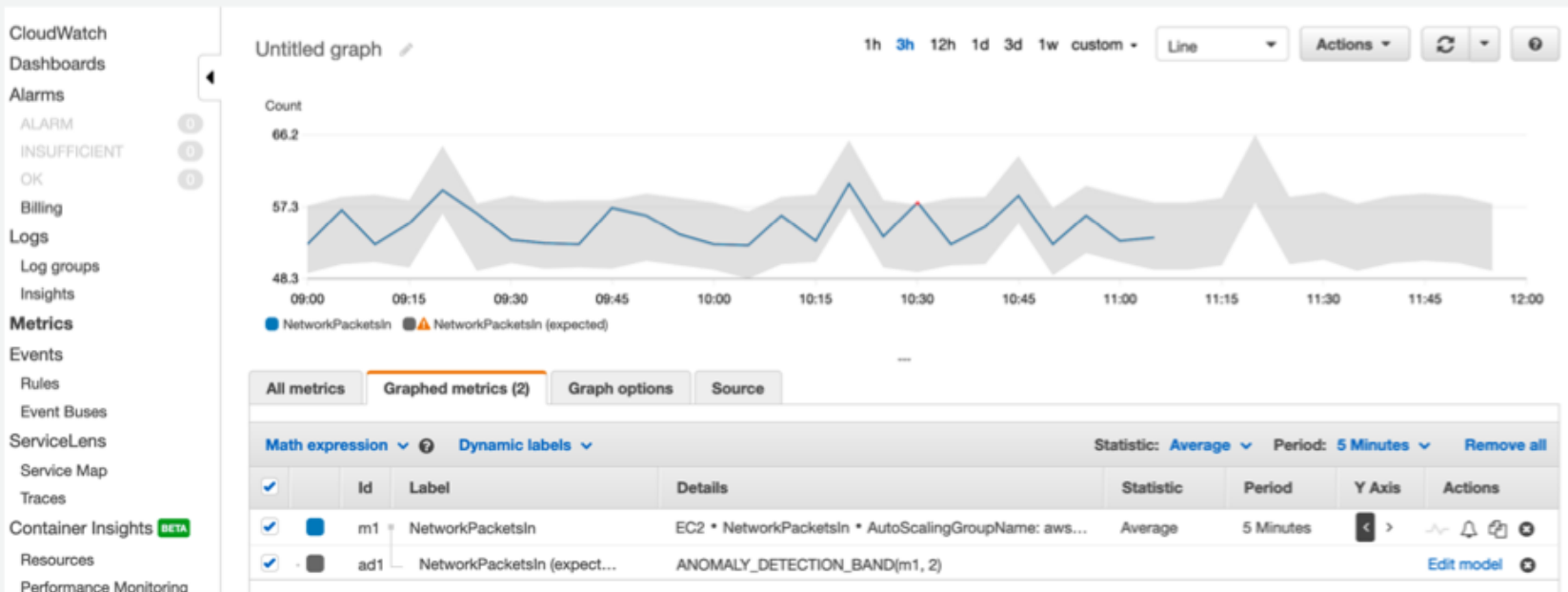




Anomaly detection



Monitor
Anomalies





Act



Automate response with CloudWatch Events



Act
Alerts

CloudWatch Events provides a near real-time stream of system events that describe changes to your AWS resources.

- Respond quickly
- Take corrective action

Write rules to indicate which events are of interest to your application and what automated actions to take when a rule matches an event.

- Invoke a Lambda Function
- Notify an SNS Topic
- Create an Ops Item in Systems Manager

The screenshot displays the AWS CloudWatch console interface for a specific event rule. At the top, the breadcrumb navigation shows 'Rules > ChangelnstanceSize' and an 'Actions' dropdown menu. Below this is a 'Summary' section containing the following information:

- ARN:** `arn:aws:events:eu-west-1:180304385487:rule/ChangelnstanceSize`
- Schedule:** Cron expression `* * * * *`
- Next 10 Trigger Date(s):**
 1. Fri, 28 Feb 2020 06:00:00 GMT
 2. Fri, 27 Mar 2020 06:00:00 GMT
 3. Fri, 24 Apr 2020 06:00:00 GMT
 4. Fri, 29 May 2020 06:00:00 GMT
 5. Fri, 26 Jun 2020 06:00:00 GMT
 6. Fri, 31 Jul 2020 06:00:00 GMT
 7. Fri, 28 Aug 2020 06:00:00 GMT
 8. Fri, 25 Sep 2020 06:00:00 GMT
 9. Fri, 30 Oct 2020 06:00:00 GMT
 10. Fri, 27 Nov 2020 06:00:00 GMT
- Status:** Enabled
- Description:** (empty)
- Monitoring:** [Show metrics for the rule](#)

Below the summary is a 'Targets' section with a 'Filter' input field. A table lists the configured targets:

Type	Name	Input
SSM Automation	ChangelnstanceSize (version \$DEFAULT)	Constant: {"InstanceId": "i-0cb0104ddf22z



Best practices

Log sampling

Frequent log rotation

Keep the application log free of spam

Rate-limit an application log's error spam



Solution Demo 2 - metrics



Key takeaways

Collect everything with ease (logs, containers, managed services)

Use embedded metric format

Use self-learning metric behavior

Automate monitoring



Break and Q&A



Observability on AWS



Observability

Monitoring

Anticipated

Anticipated,
Unexpected

Observability

Unexpected

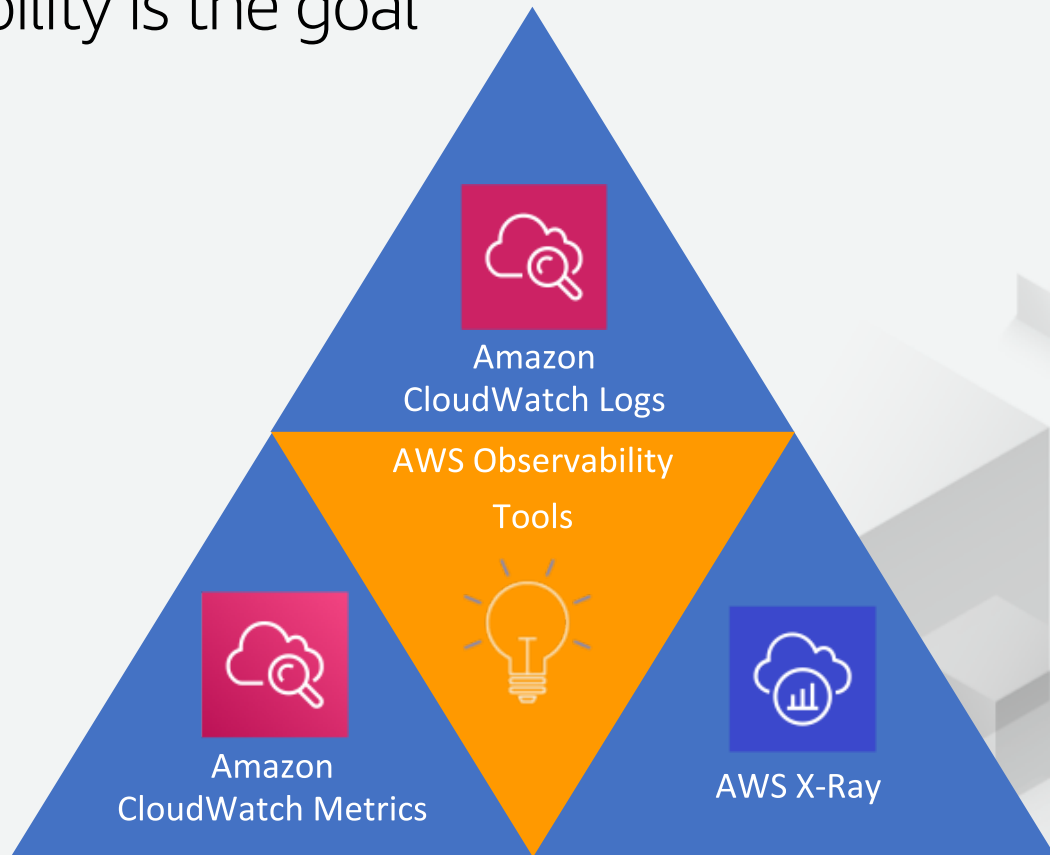


Observability is the goal





Observability is the goal



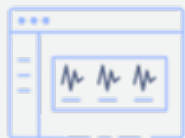


Breadth and depth observability tools



Collect

- Embedded Metric Format
- Metric Filters
- StatsD & CollectD
- AWS PrivateLink



Monitor

- Cross-Account, Cross-Region Dashboards
- Automatic Dashboards
- Metric Math
- SQS & SNS add support for X-Ray



Act

- Synthetics
- Anomaly Detection
- Metric Math Alarms
- Search Expressions



Analyze

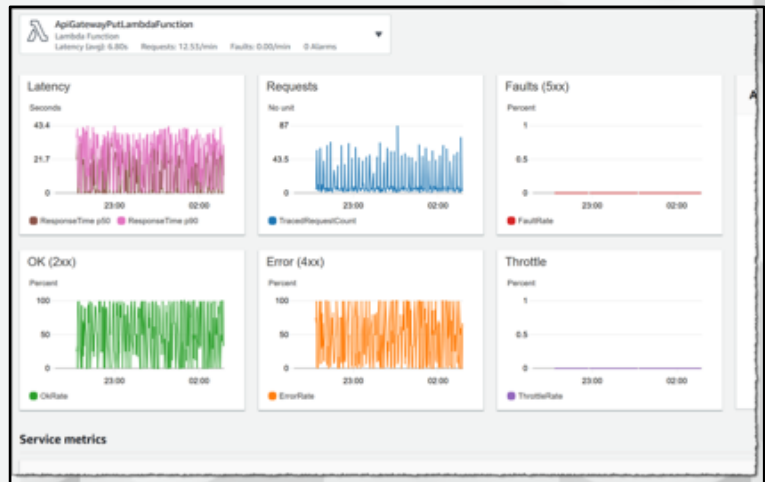
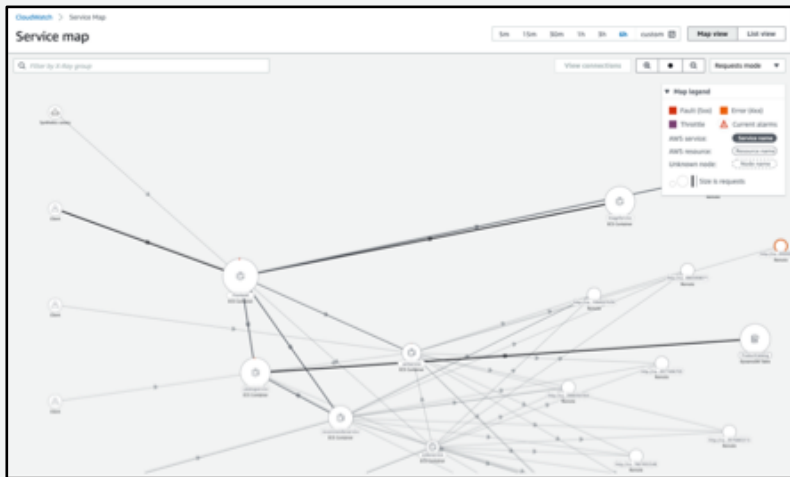
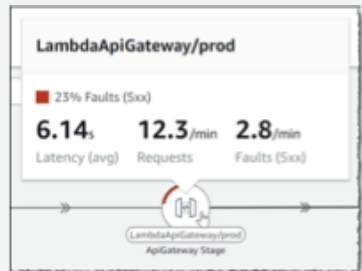
- ServiceLens
- Contributor Insights
- Container Insights
- Logs Insights
- X-Ray Analytics



ServiceLens



Fully managed solution that helps customers visualize and analyze the health, performance, and availability of their distributed applications in a single place.



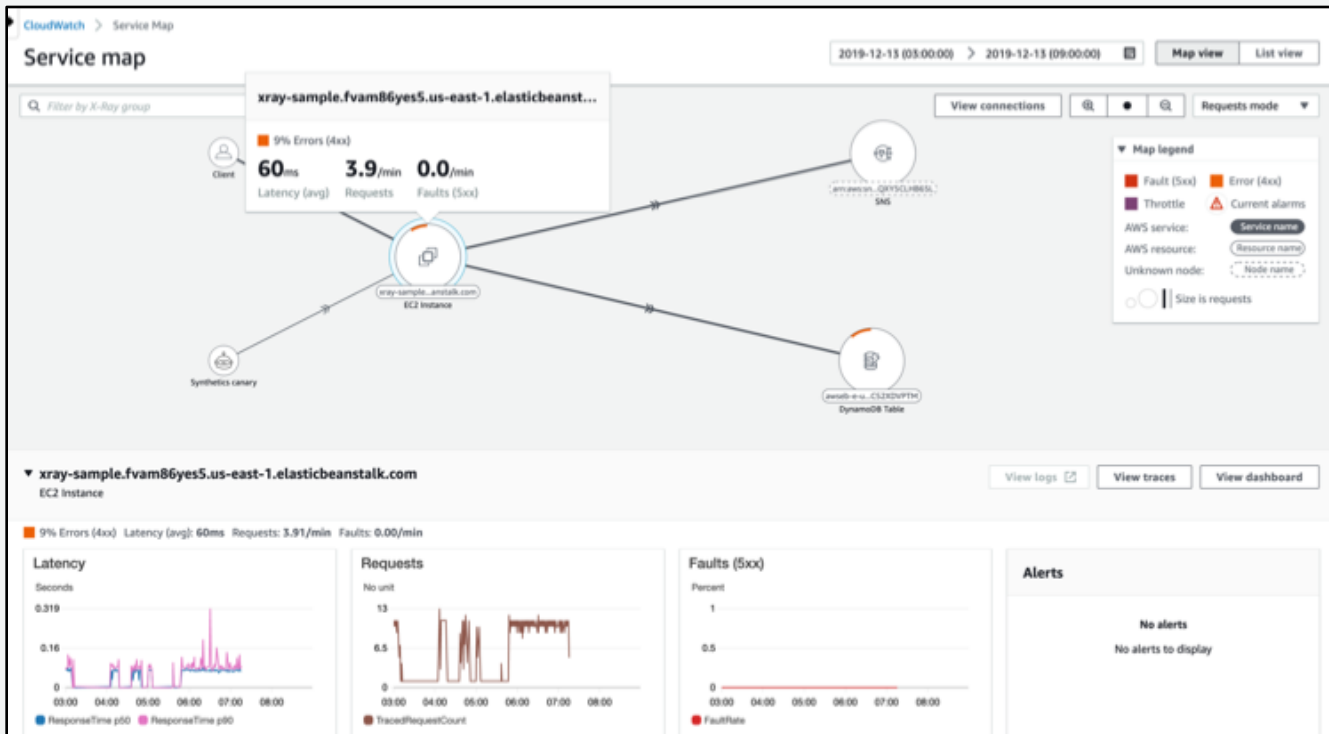


ServiceLens (1/2) Service Map



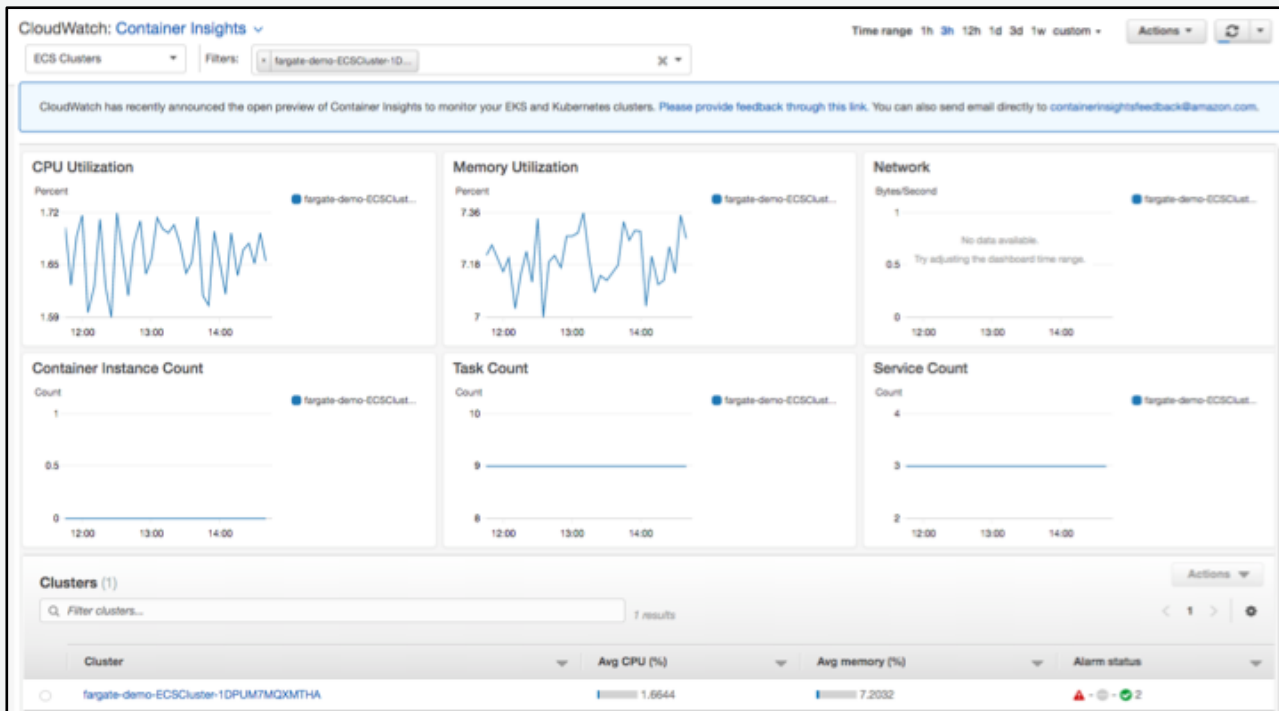


ServiceLens (2/2) Traces



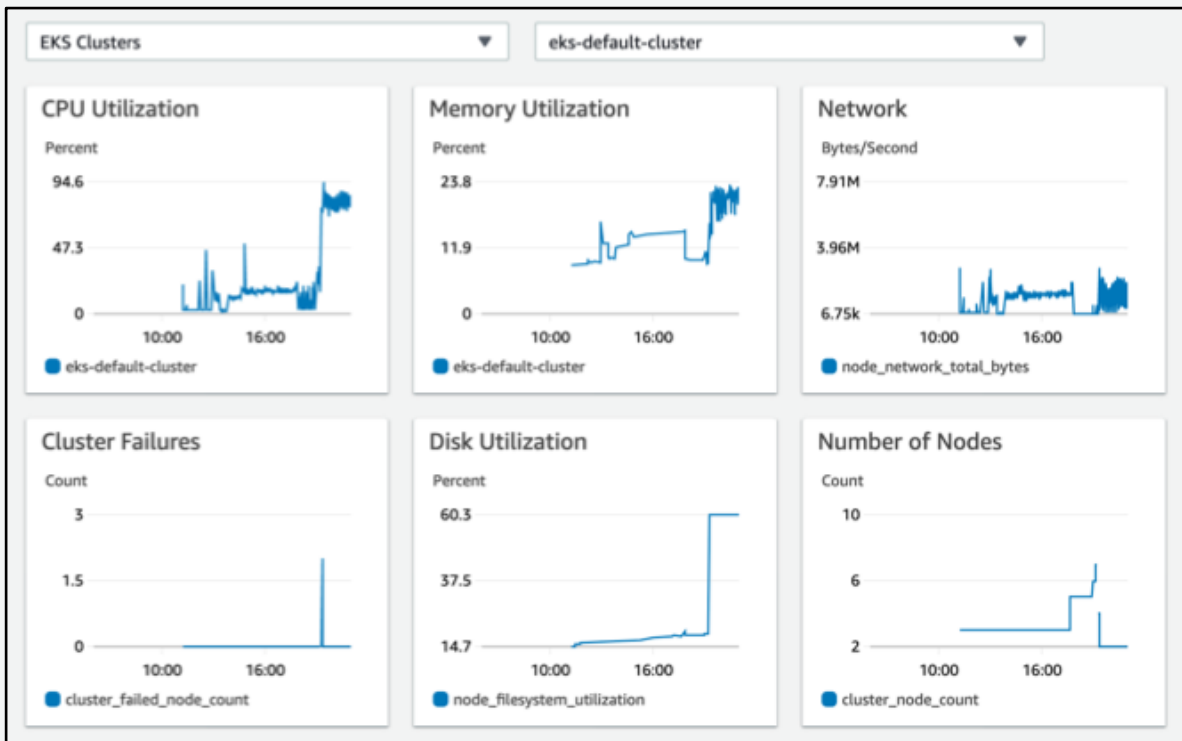


Container Insights (1/2)





Container Insights (2/2). Load test.





Cloudwatch logs insights

/aws/containerinsights/eks-default-cluster/application X Clear

```
fields @timestamp, @log, @logStream, @message
| filter ispresent(kubernetes.container_name) and kubernetes.container_name like /dns-tester/ and log like /DNS Timeout/
```

Run query Save History

Logs Visualization Export results Add to dashboard

Showing 1000 of 58,418 records matched ⓘ Hide histogram
68,548 records (66.2 MB) scanned in 4.5s @ 15,290 records/s (14.8 MB/s)

#	@timestamp	@log	@logStream
1	2020-05-08T13:02:29...	447948760339:/aws/containerinsights/eks-default-cluster/applicat...	dns-tester-6b676b4b9d-pq2c7_default_dns-tester-
2	2020-05-08T13:02:29...	447948760339:/aws/containerinsights/eks-default-cluster/applicat...	dns-tester-6b676b4b9d-zwkv2_default_dns-tester-
3	2020-05-08T13:02:29...	447948760339:/aws/containerinsights/eks-default-cluster/applicat...	dns-tester-6b676b4b9d-pgzfw_default_dns-tester-
4	2020-05-08T13:02:29...	447948760339:/aws/containerinsights/eks-default-cluster/applicat...	dns-tester-6b676b4b9d-gfxcw_default_dns-tester-
5	2020-05-08T13:02:29...	447948760339:/aws/containerinsights/eks-default-cluster/applicat...	dns-tester-6b676b4b9d-mdfhq_default_dns-tester-
6	2020-05-08T13:02:29...	447948760339:/aws/containerinsights/eks-default-cluster/applicat...	dns-tester-6b676b4b9d-qk9xt_default_dns-tester-



Contributor insights

Wizard Syntax

Rule name* VPCFlowLogsRule

Log group(s)* VPCFlowLogGroup

Select by prefix match

Log format*
 JSON
 CLF (Common Log Format)

Fields

5	destination address
Position	Alias
7	destination port
Position	Alias
9	packet count
Position	Alias
Position	Alias

Contribution*

destination address

Unique key

Unique key

packet count

Value* - optional

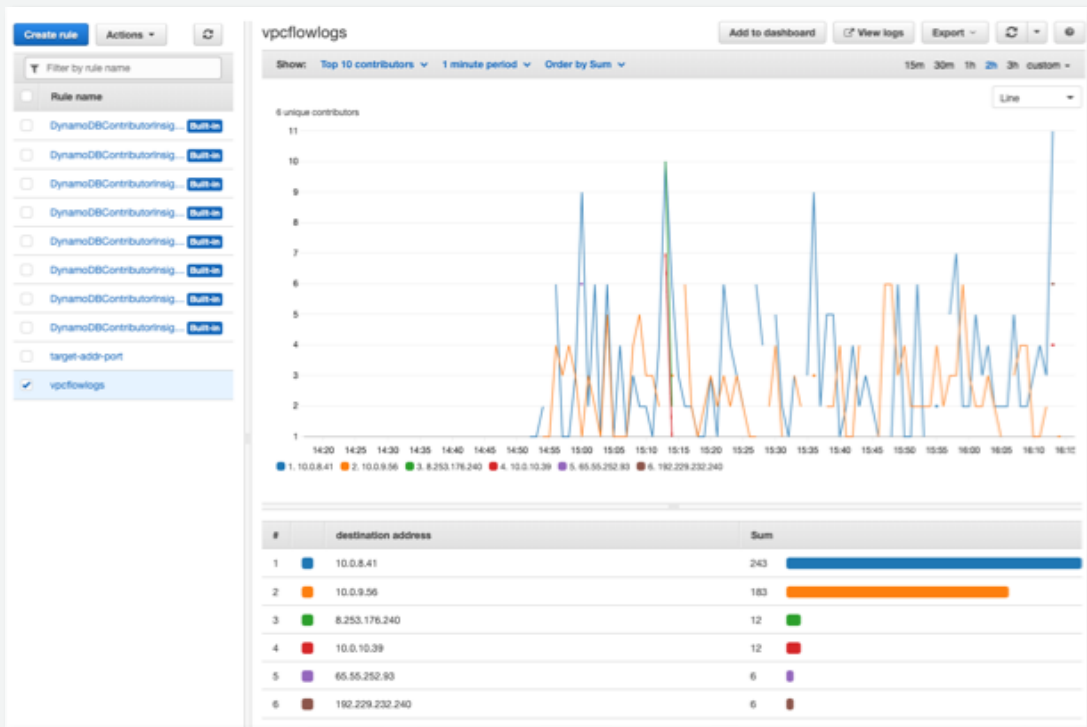
Filters

Use filters to process only specific log events. You can apply up to 4 filters per rule.

[Add filter - optional](#)

Aggregate on*
 COUNT
 SUM

Rule state
 Create rule in disabled state.
A disabled rule will not incur any costs, but will count towards your rule limit.





Solution Demo 3 - Observability



Best practices for observability

Log the availability and latency of all of dependencies.

Log a trace ID and propagate it in backend calls

Log different latency metrics depending on status code and size.

Log important metadata about the unit of work

Add an additional counter for every error reason



Key takeaways

Cloudwatch is evolving

Container insights

Application and microservice tracing

Correlate metrics, logs and traces

Top contributors





Q&A



Thank you!

Sergey Kurson
Solutions Architect, AWS
kursonsk@amazon.com

