KodeKloud

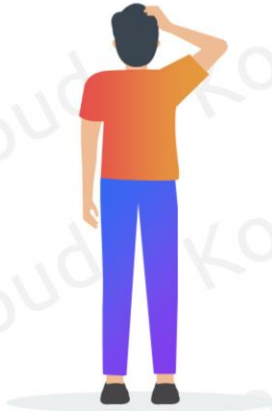
Follow us on https://kodekloud.com/ to learn more.

Observability in AWS – Introduction

What is Observability?

What is Observability?

- Is my system down?
- What is the root cause of this incident?
- How do I configure the alarm and monitoring?
- Is my application slow? If so, why is it happening?

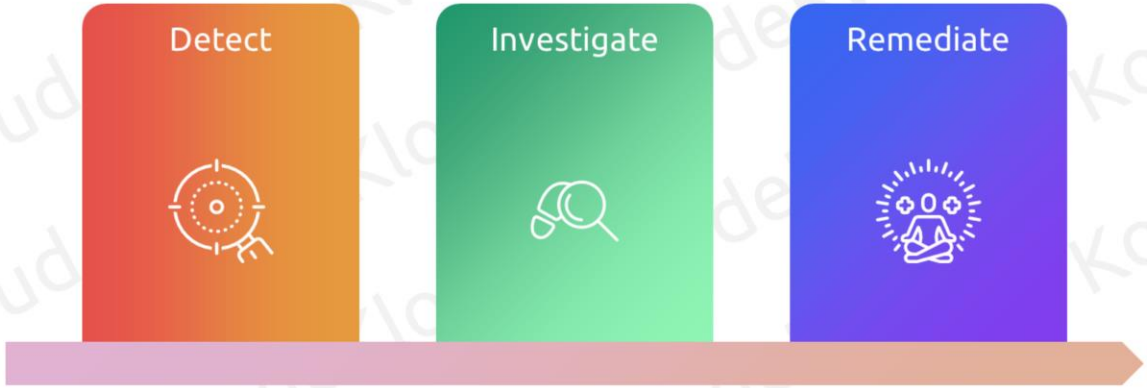


What is Observability?

"Observability" describes how well you can understand what is happening in a system.

Foundations of Observability

Observability action plan



Proactive Problem Detection

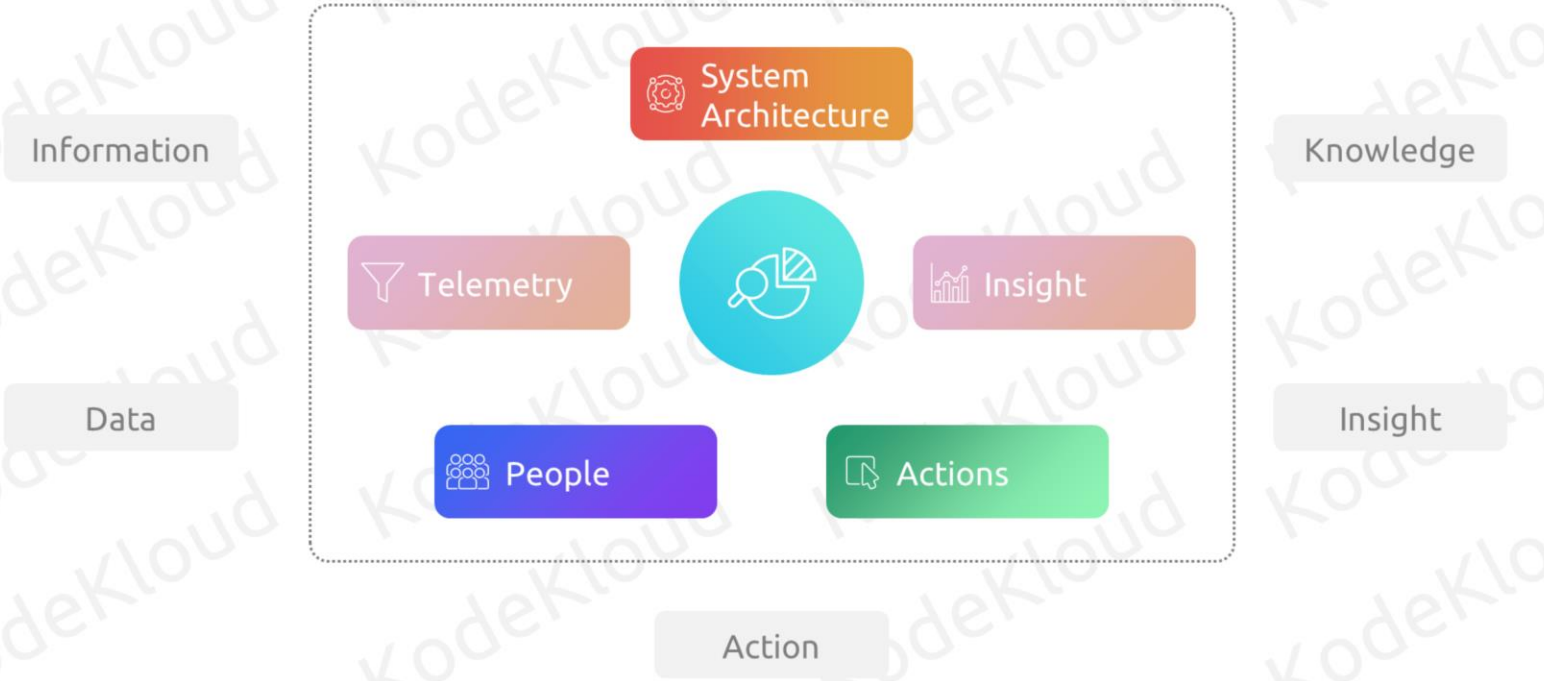
Why?

Identifying and addressing potential issues before they escalate

Result: Ensuring software reliability, performance, and user satisfaction

Monitoring Strategy and Categories of Insights

Monitoring Strategy – Components



Categories of Insights

- **Fault management**
- Configuration management
- Accounting management
- Performance management
- Security management

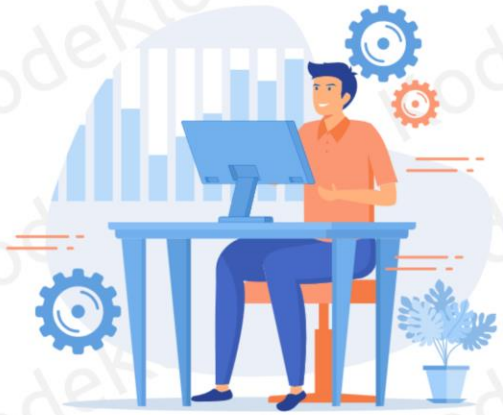
FCAPS Model

Faults: Monitoring to avert or respond to incidents

Categories of Insights

- Fault management
- Configuration management
- Accounting management
- Performance management
- Security management

FCAPS Model

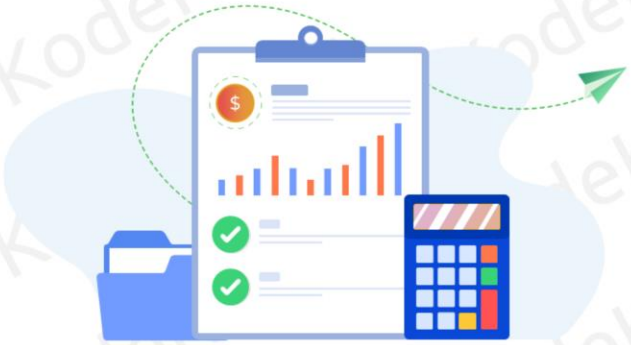


Configuration: Monitoring and tracking configurations and changes

Categories of Insights

- Fault management
- Configuration management
- **Accounting management**
- Performance management
- Security management

FCAPS Model



Accounting: Monitoring utilization and enabling attribution

Categories of Insights

- Fault management
- Configuration management
- Accounting management
- **Performance management**
- Security management

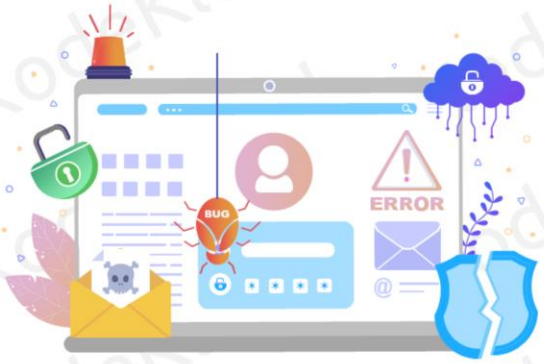
FCAPS Model

Performance: Monitoring for constrained components and the impact of changes

Categories of Insights

- Fault management
- Configuration management
- Accounting management
- Performance management
- **Security management**

FCAPS Model

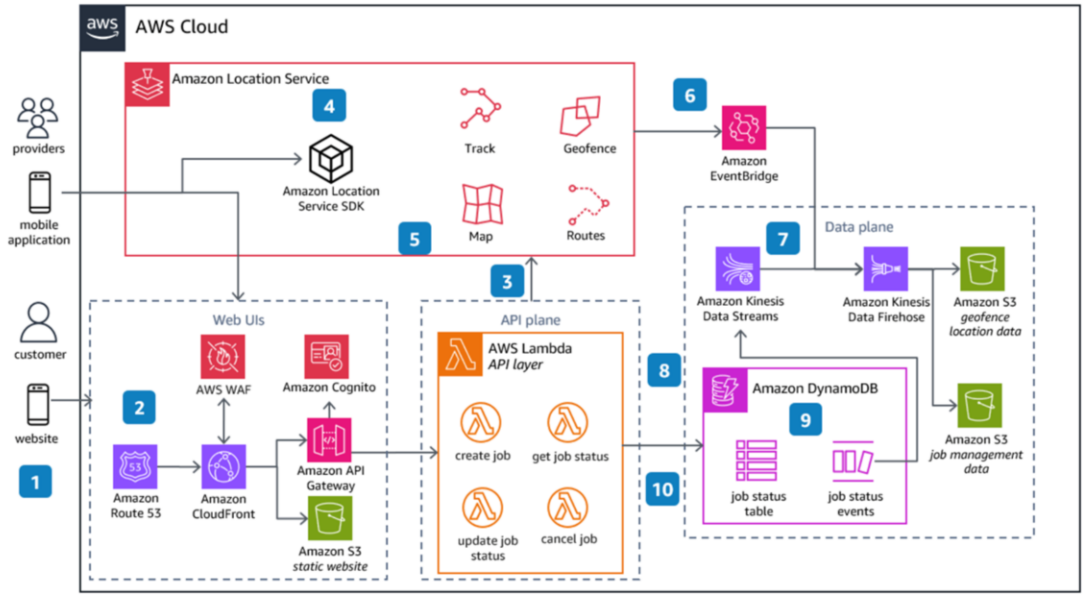


Security: Monitoring access, security controls, and identifying inappropriate or malicious activity

CloudWatch – Use Cases

You have been tasked to build centralized

- Alarms,
- Notifications,
- Logging and
- Observability System in AWS



Ref: AWS Sample Design

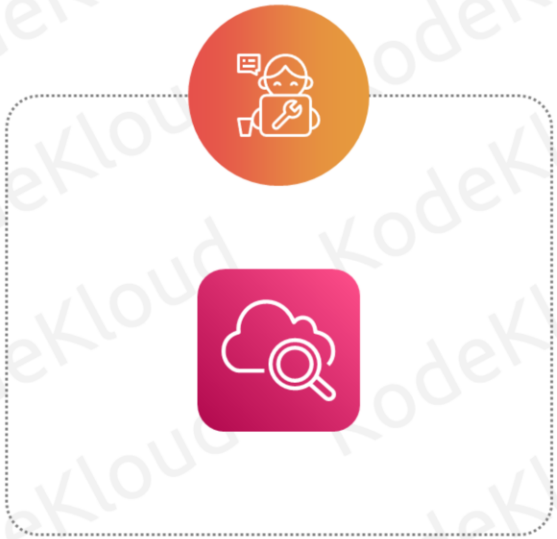
This is the beginning of our journey into understanding how CloudWatch provides the tools and capabilities to monitor your applications and systems, ensuring that they are running smoothly and efficiently.

AWS CloudWatch – Introduction and Key Features

Monitoring Service

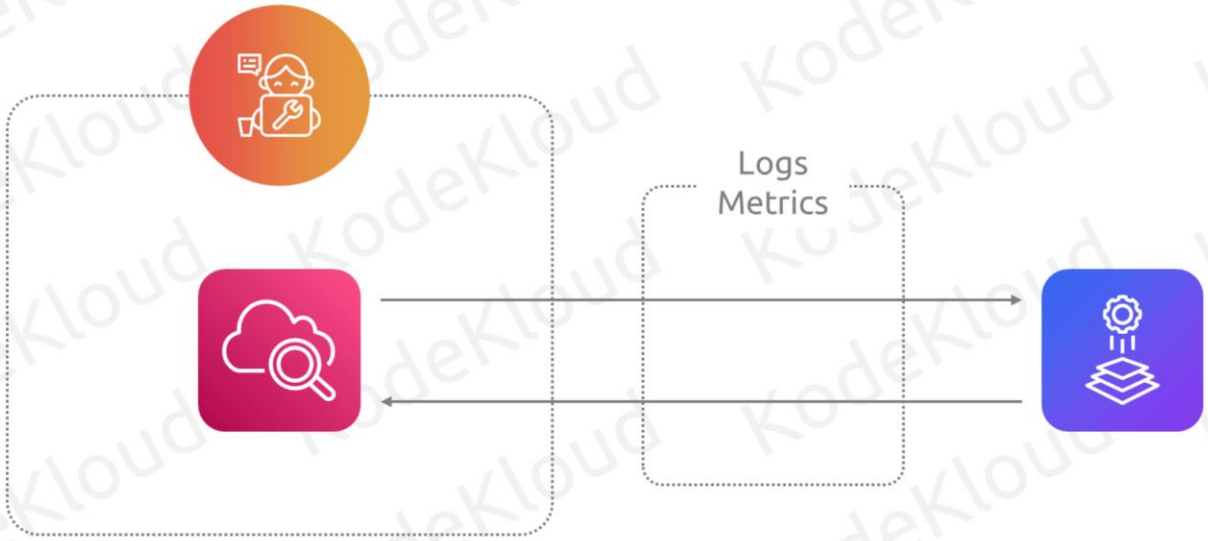
Cloudwatch is more than just a monitoring service, it is an integral component of any DevOps professional. It allows you to collect and access all your performance and operational data in the form of logs metrics from a single platform. This capability is crucial for maintaining the health of your application and quickly diagnosing any issues that arise. First, let's talk about the full stack.

AWS CloudWatch – Introduction and Key Features



Cloudwatch is more than just a monitoring service, it is an integral component of any DevOps professional. It allows you to collect and access all your performance and operational data in the form of logs metrics from a single platform. This capability is crucial for maintaining the health of your application and quickly diagnosing any issues that arise. First, let's talk about the full stack.

AWS CloudWatch – Introduction and Key Features



Cloudwatch is more than just a monitoring service, it is an integral component of any DevOps professional. It allows you to collect and access all your performance and operational data in the form of logs metrics from a single platform. This capability is crucial for maintaining the health of your application and quickly diagnosing any issues that arise. First, let's talk about the full stack.

AWS CloudWatch – Introduction and Key Features

Cloudwatch is more than just a monitoring service, it is an integral component of any DevOps professional. It allows you to collect and access all your performance and operational data in the form of logs metrics from a single platform. This capability is crucial for maintaining the health of your application and quickly diagnosing any issues that arise. First, let's talk about the full stack.

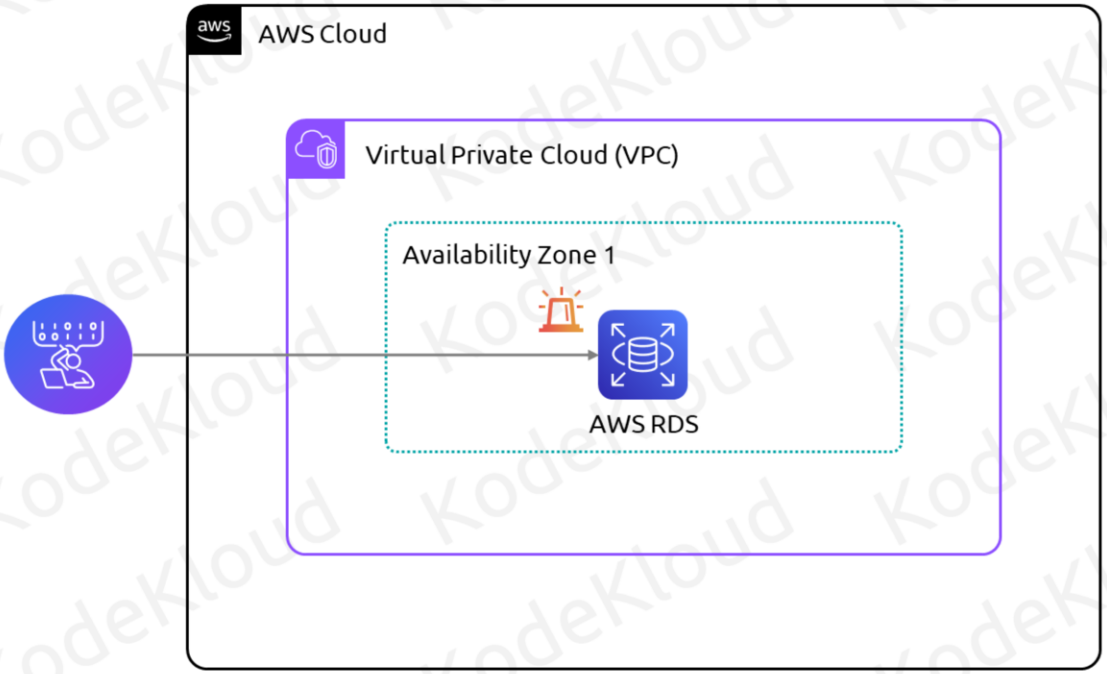
AWS CloudWatch – Introduction and Key Features

Full-Stack Observability

Anatomy of Alarms

Metrics and Metric Dimensions

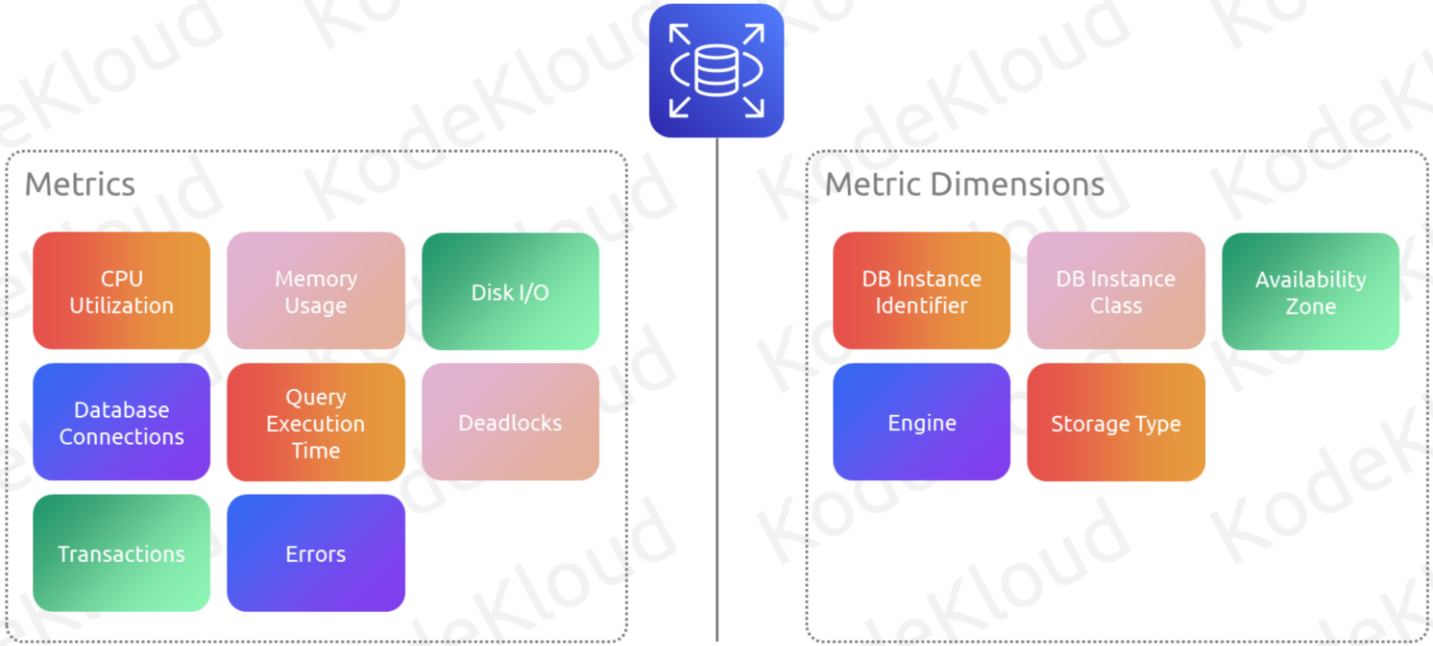
Metrics and Metric Dimensions



What should I monitor?

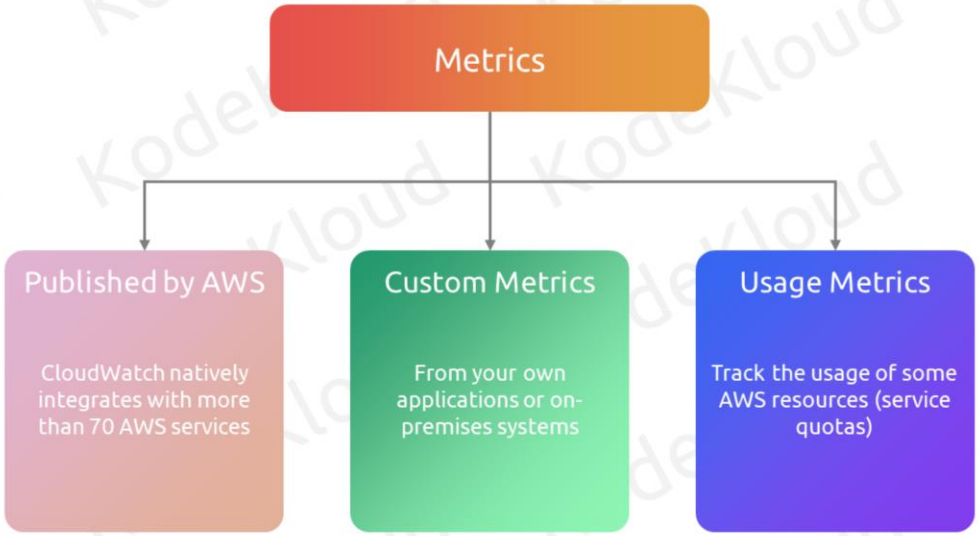
How should I monitor?

Metrics and Metric Dimensions

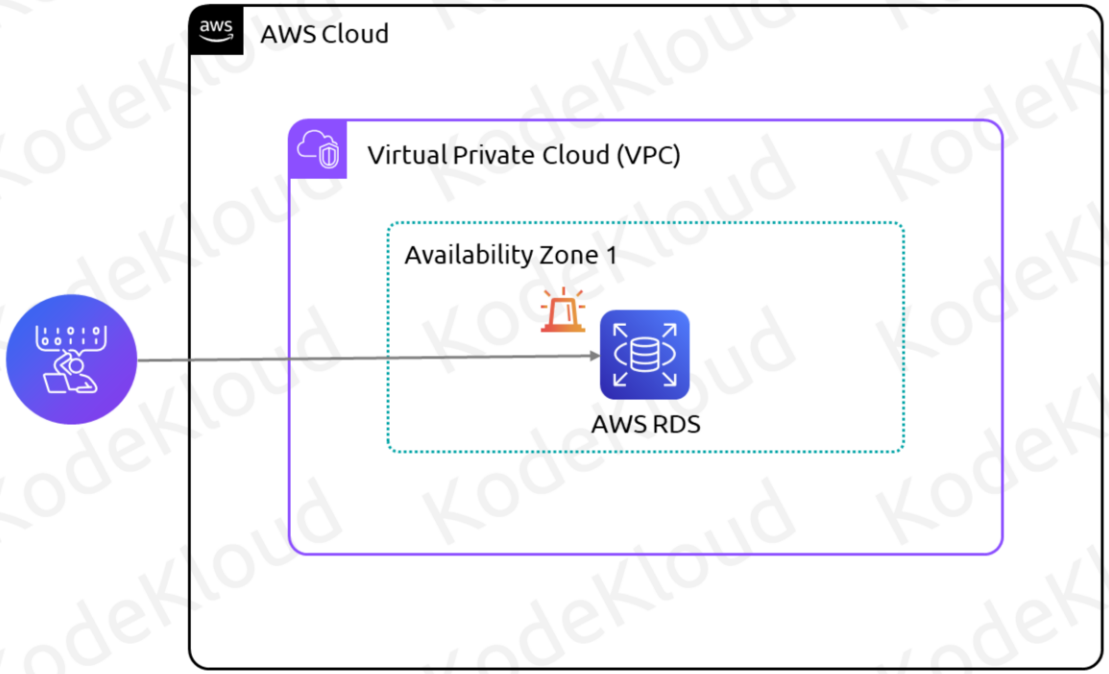


Built-in and Custom Metrics

Built-in and Custom Metrics



Built-in and Custom Metrics



What should I monitor?

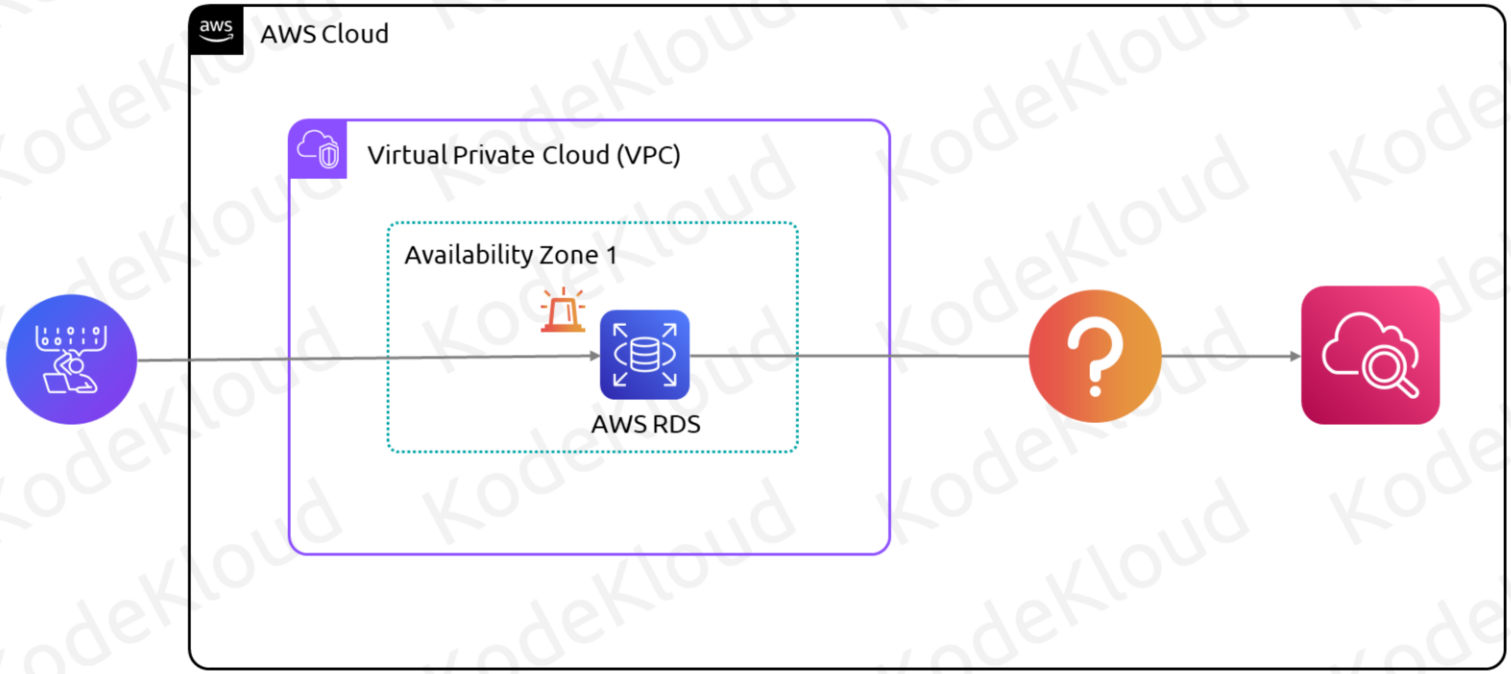
- CPU utilization
- Memory usage
- Transactions

How should I monitor as custom metrics?

- Number of successful logins
- Average query execution time

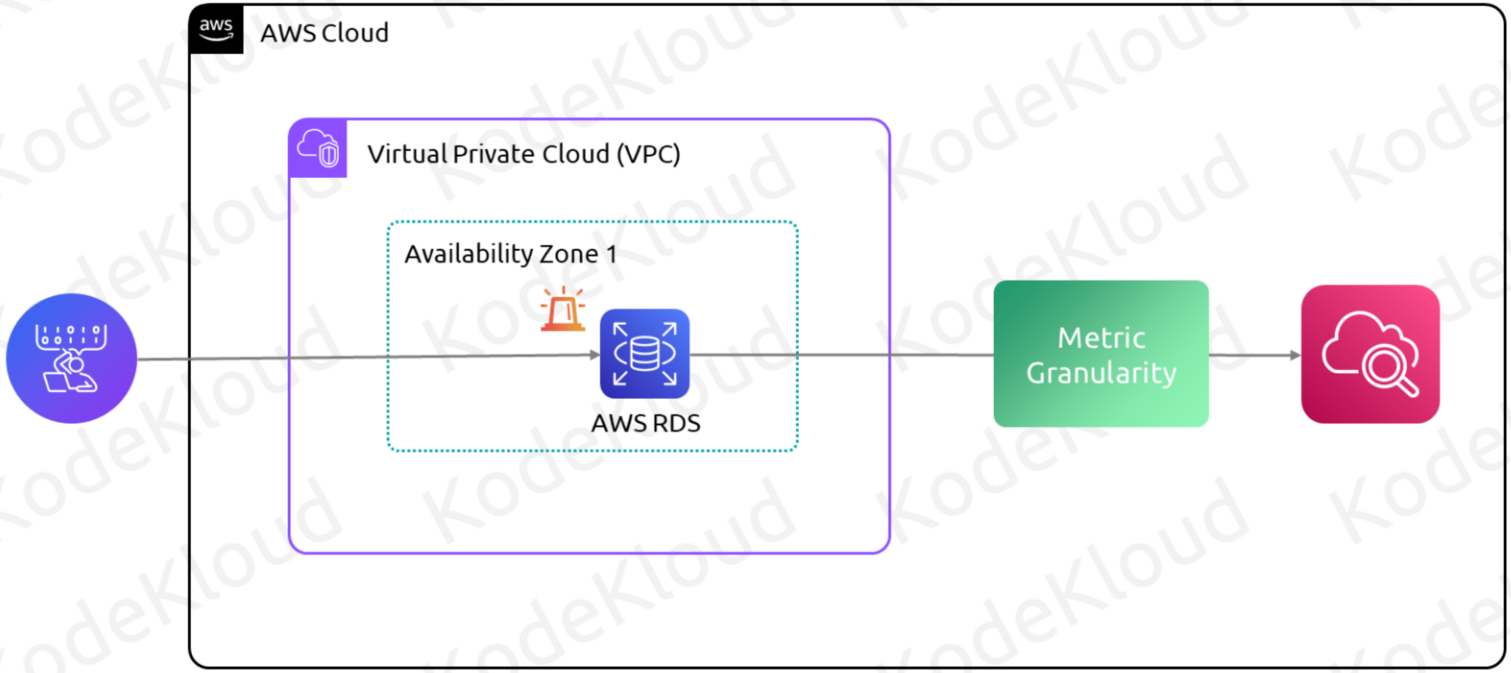
Metric Granularity and Aggregation

Metric Granularity and Aggregation



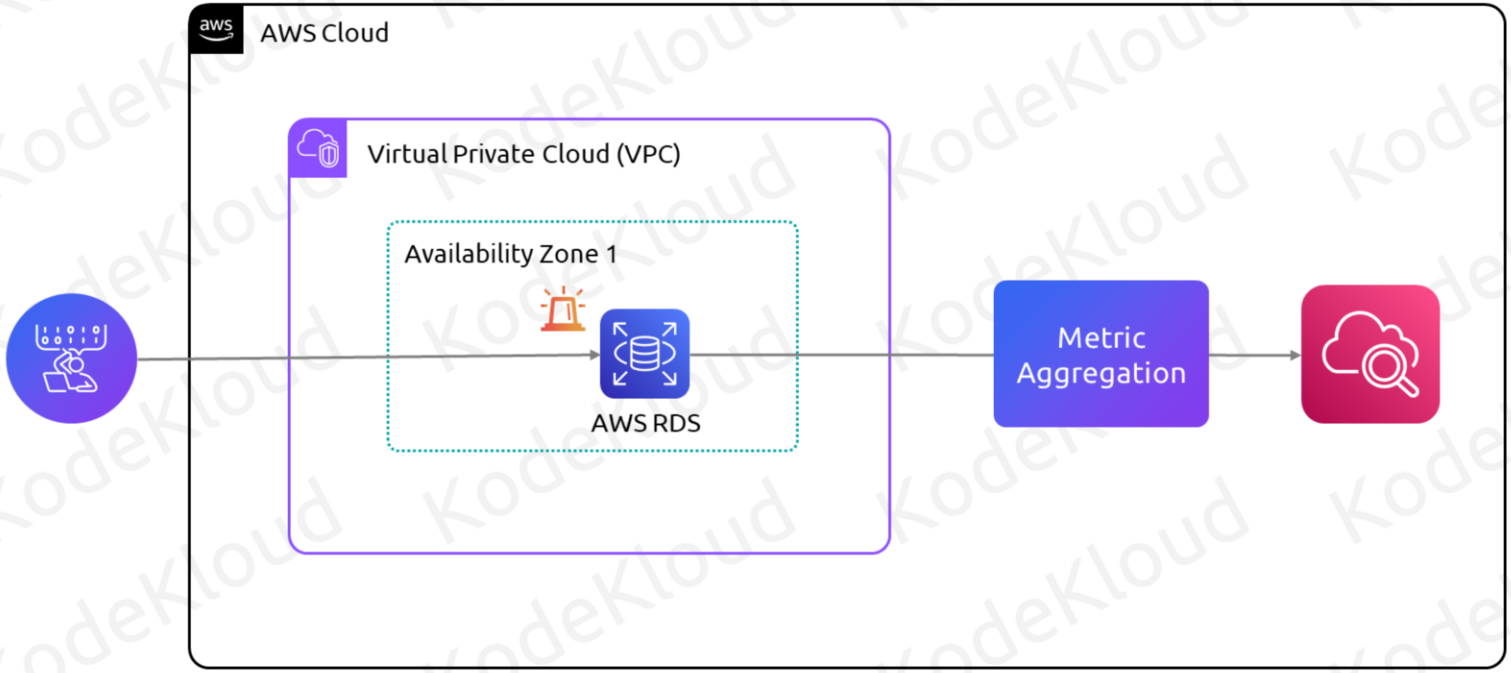
Metric Granularity

The frequency at which Amazon CloudWatch collects and stores metric data



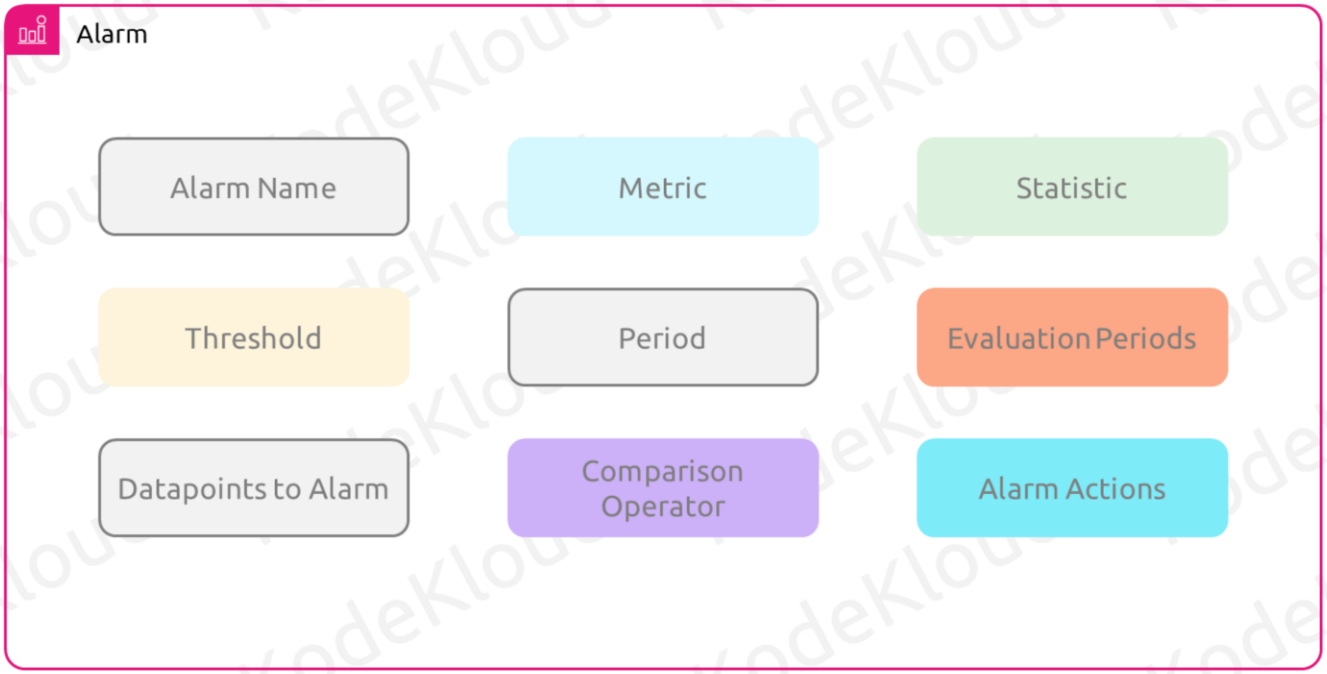
Metric Aggregation

The process of combining multiple metric data points into a single data point focused on retrieving statistics when required

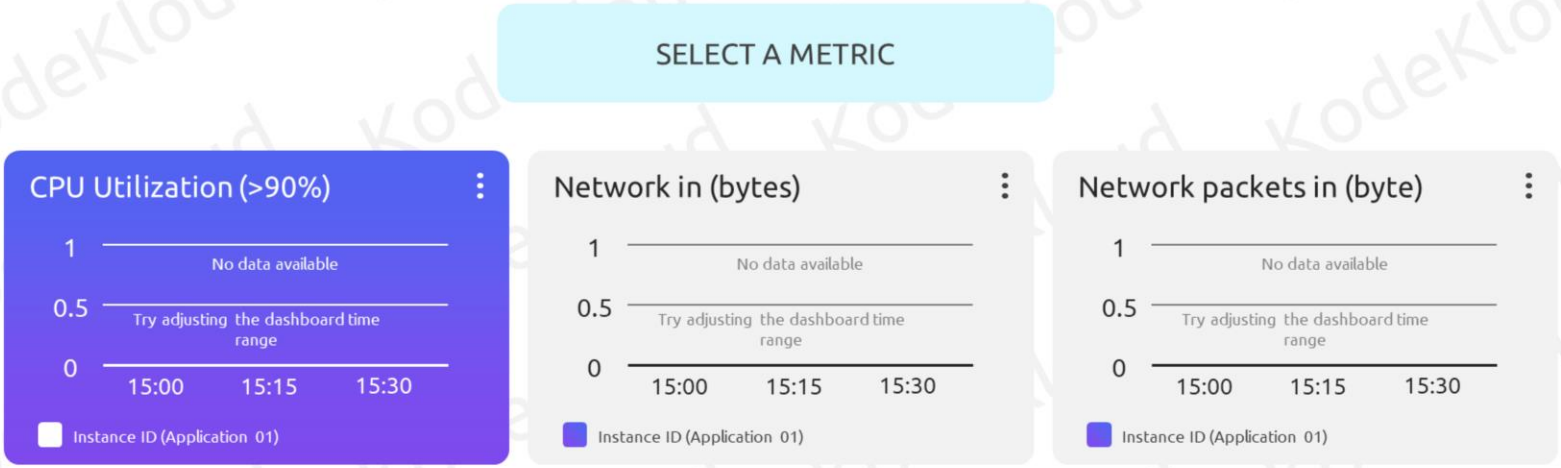


Alarm Anatomy

Alarm Anatomy in AWS CloudWatch



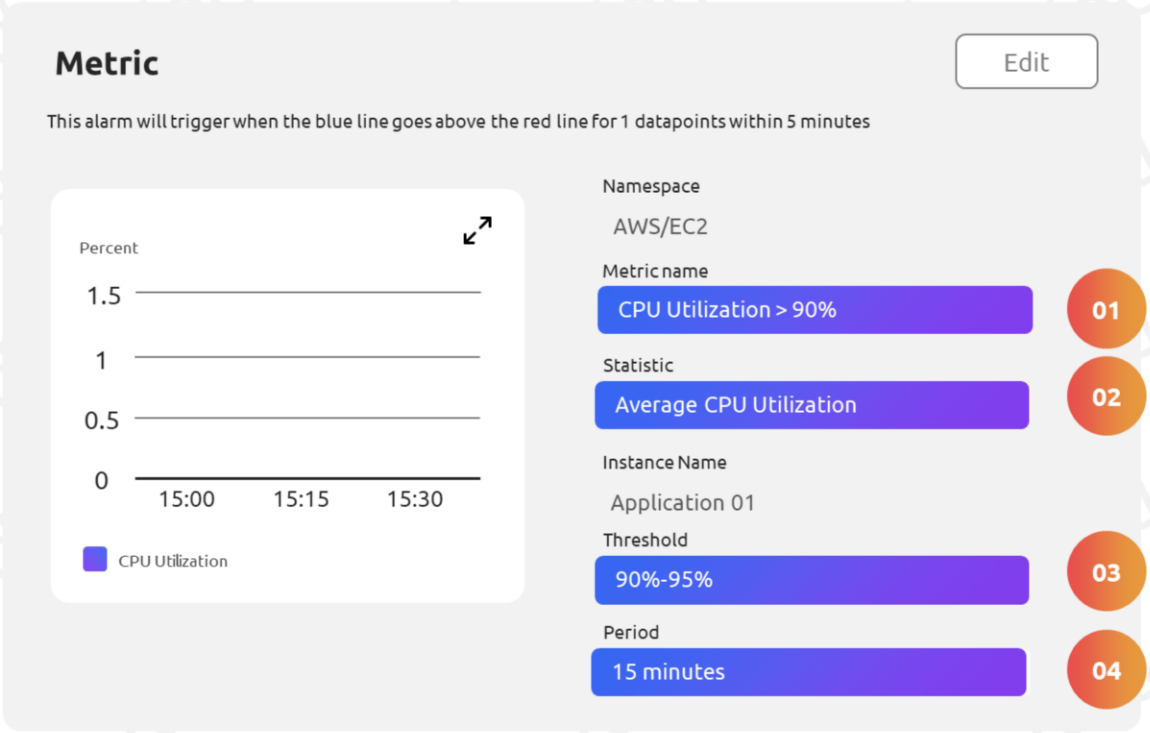
Alarm Anatomy in AWS CloudWatch



Now we have an RDS instance. What is a good alarm for this? An alarm titled CPU Utilization greater than 90% For RDS Instance Application 01. This can be a useful alarm. What is the metric? COP utilization is the metric that we want to monitor. What is the statistic? Is it the average CP utilization? Is it the sum of CP utilization? Let us go with average CP utilization. Then the threshold is a threshold, 90 percent, 95%. And then we have the period of monitoring. That is, how long do we want to monitor this? Maybe 10 minutes? 15 minutes? And what is the evaluation. When the alarm state is reached? Maybe 5 minutes. Number of data points during the evaluation maybe. If we have two consecutive data points

where we say that TCP utilization of the RDS instance is 90%, that is enough. Then the comparison operator if the CP utilization is equal to 95% is enough or should it be greater than let us say greater than or equal. And finally the RRM action that is sending out an e-mail or a call to the on compression. There you go. Now understand, based on the example that I gave you, what a good LRM should consist of. All of these terminologies is super important and has to be configured properly if you want to monitor your application effectively.

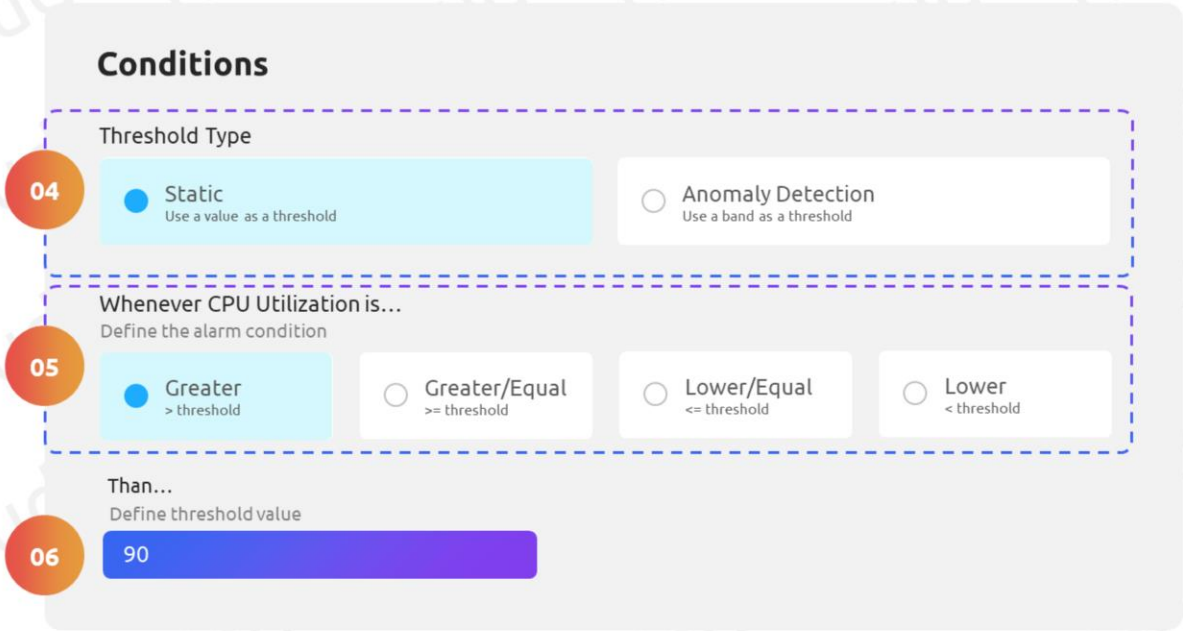
Alarm Anatomy in AWS CloudWatch



Now we have an RDS instance. What is a good alarm for this? An alarm titled CPU Utilization greater than 90% For RDS Instance Application 01. This can be a useful alarm. What is the metric? CPU utilization is the metric that we want to monitor. What is the statistic? Is it the average CP utilization? Is it the sum of CP utilization? Let us go with average CP utilization. Then the threshold is a threshold, 90 percent, 95%. And then we have the period of monitoring. That is, how long do we want to monitor this? Maybe 10 minutes? 15 minutes? And what is the evaluation. When the alarm state is reached? Maybe 5 minutes. Number of data points during the evaluation maybe. If we have two consecutive data points

where we say that TCP utilization of the RDS instance is 90%, that is enough. Then the comparison operator if the CP utilization is equal to 95% is enough or should it be greater than let us say greater than or equal. And finally the RRM action that is sending out an e-mail or a call to the on compression. There you go. Now understand, based on the example that I gave you, what a good LRM should consist of. All of these terminologies is super important and has to be configured properly if you want to monitor your application effectively.

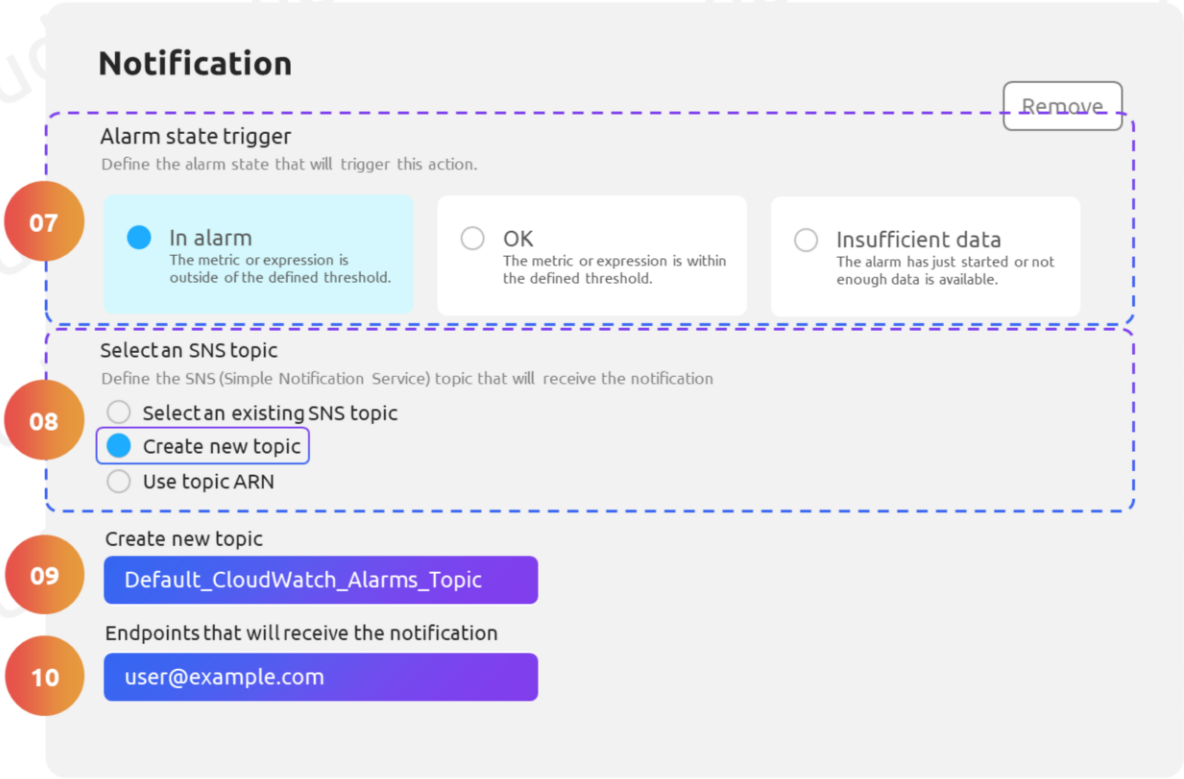
Alarm Anatomy in AWS CloudWatch



Now we have an RDS instance. What is a good alarm for this? An alarm titled CPU Utilization greater than 90% For RDS Instance Application 01. This can be a useful alarm. What is the metric? COP utilization is the metric that we want to monitor. What is the statistic? Is it the average CP utilization? Is it the sum of CP utilization? Let us go with average CP utilization. Then the threshold is a threshold, 90 percent, 95%. And then we have the period of monitoring. That is, how long do we want to monitor this? Maybe 10 minutes? 15 minutes? And what is the evaluation. When the alarm state is reached? Maybe 5 minutes. Number of data points during the evaluation maybe. If we have two consecutive data points

where we say that TCP utilization of the RDS instance is 90%, that is enough. Then the comparison operator if the CP utilization is equal to 95% is enough or should it be greater than let us say greater than or equal. And finally the RRM action that is sending out an e-mail or a call to the on compression. There you go. Now understand, based on the example that I gave you, what a good LRM should consist of. All of these terminologies is super important and has to be configured properly if you want to monitor your application effectively.

Alarm Anatomy in AWS CloudWatch

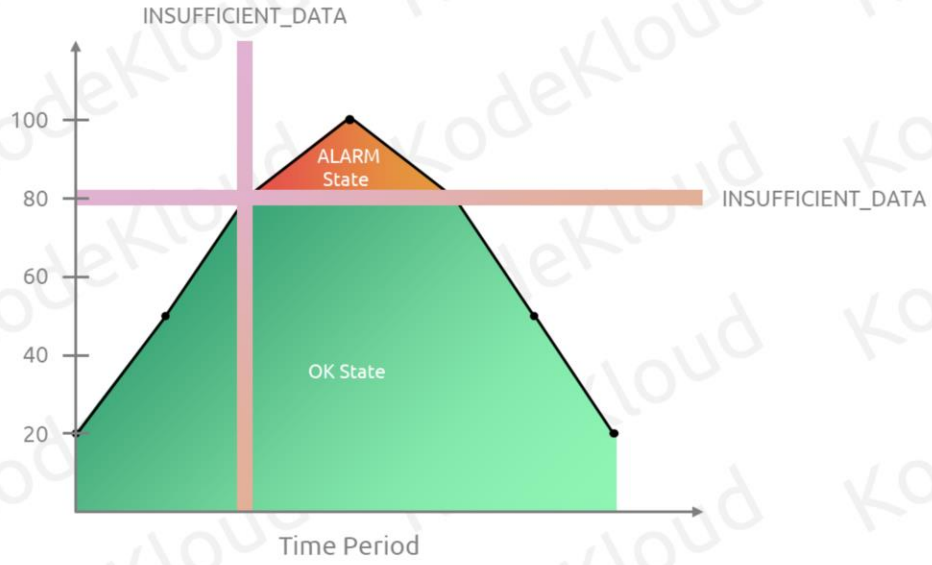


Now we have an RDS instance. What is a good alarm for this? An alarm titled CPU Utilization greater than 90% For RDS Instance Application 01. This can be a useful alarm. What is the metric? COP utilization is the metric that we want to monitor. What is the statistic? Is it the average CP utilization? Is it the sum of CP utilization? Let us go with average CP utilization. Then the threshold is a threshold, 90 percent, 95%. And then we have the period of monitoring. That is, how long do we want to monitor this? Maybe 10 minutes? 15 minutes? And what is the evaluation. When the alarm state is reached? Maybe 5 minutes. Number of data points during the evaluation maybe. If we have two consecutive data points

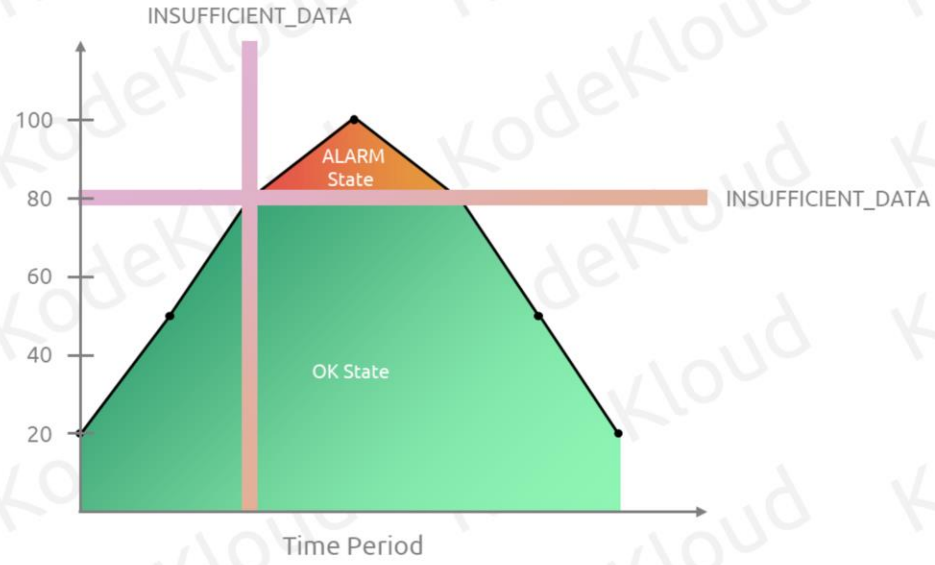
where we say that TCP utilization of the RDS instance is 90%, that is enough. Then the comparison operator if the CP utilization is equal to 95% is enough or should it be greater than let us say greater than or equal. And finally the RRM action that is sending out an e-mail or a call to the on compression. There you go. Now understand, based on the example that I gave you, what a good LRM should consist of. All of these terminologies is super important and has to be configured properly if you want to monitor your application effectively.

Alarm States

Alarm States



Alarm States



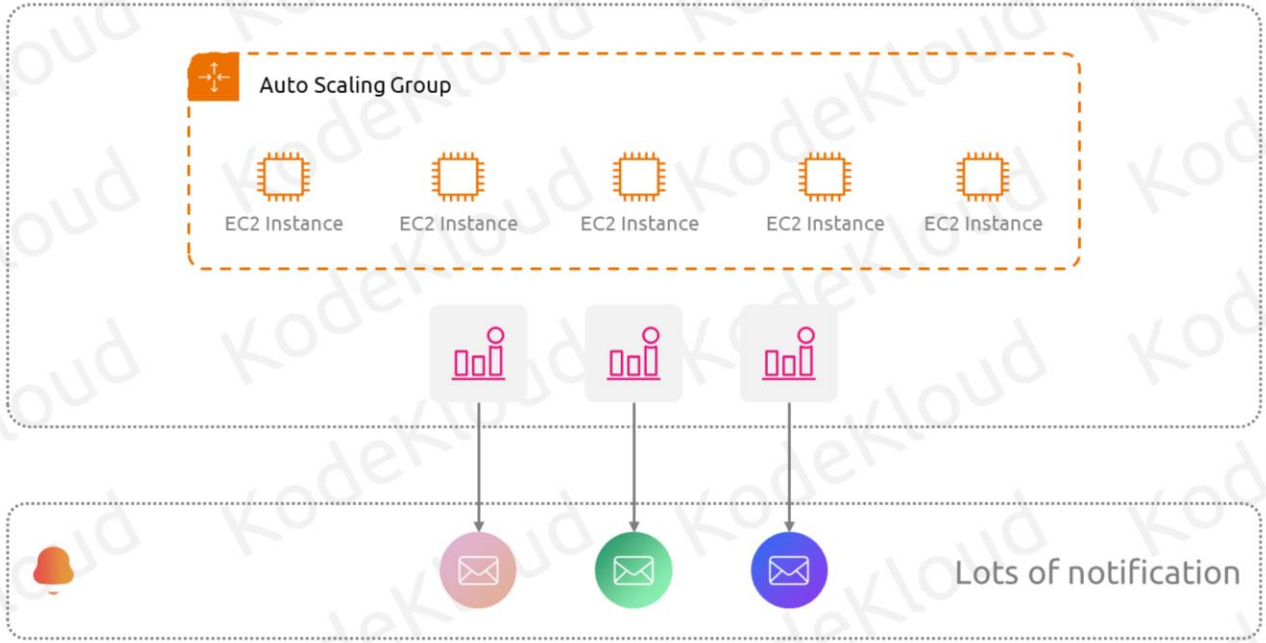
The metric is within the defined threshold.

The metric is outside of the defined threshold.

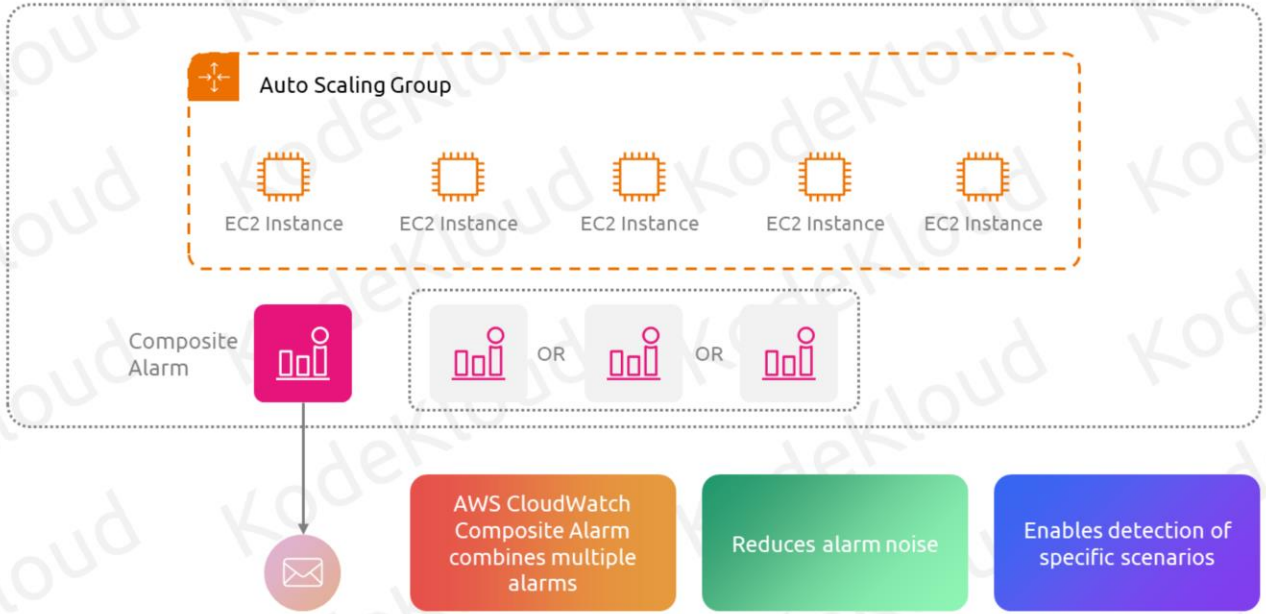
The alarm has just started, the metric is not available, or sufficient data is not available for the metric to determine the alarm state.

Composite Alarms

Composite Alarms



Composite Alarms



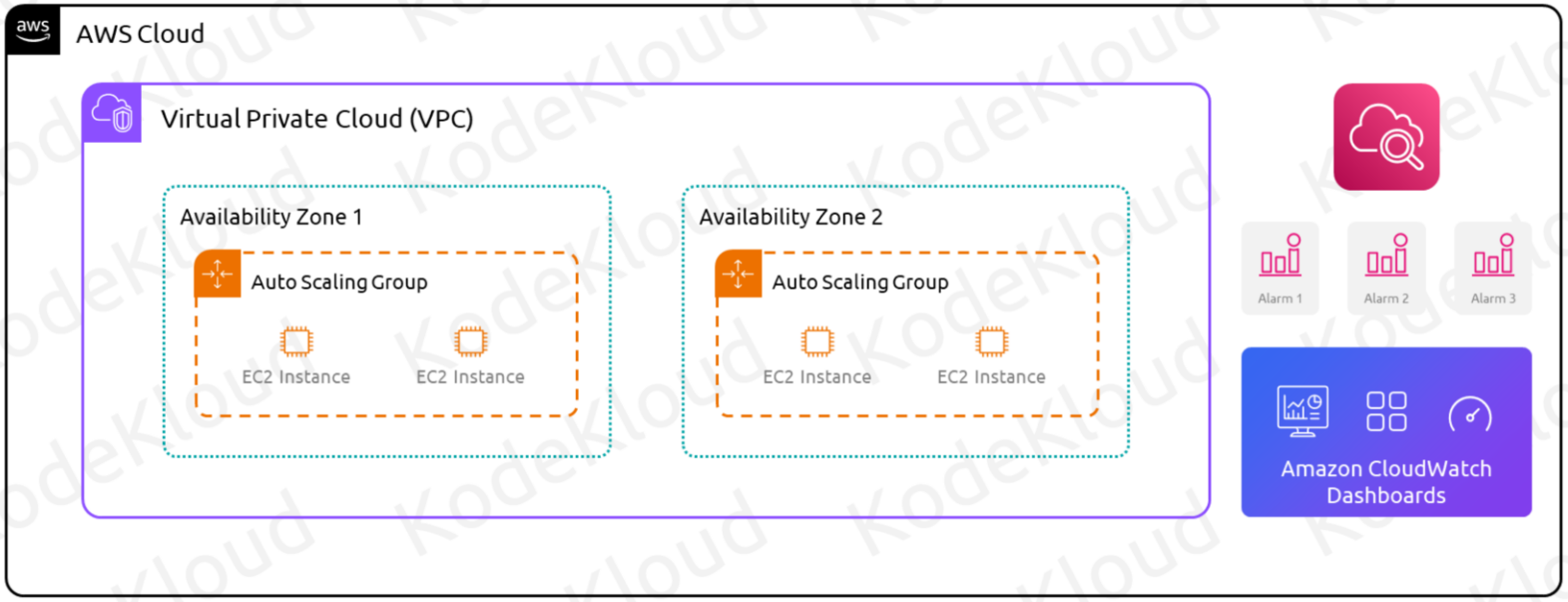
AWS CloudWatch Composite Alarm combines multiple alarms

Reduces alarm noise

Enables detection of specific scenarios

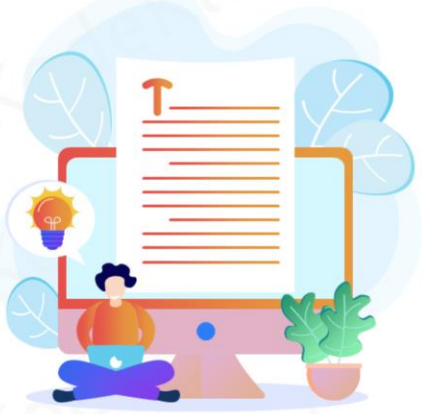
Dashboards in CloudWatch

Dashboards in CloudWatch



Dashboards in CloudWatch

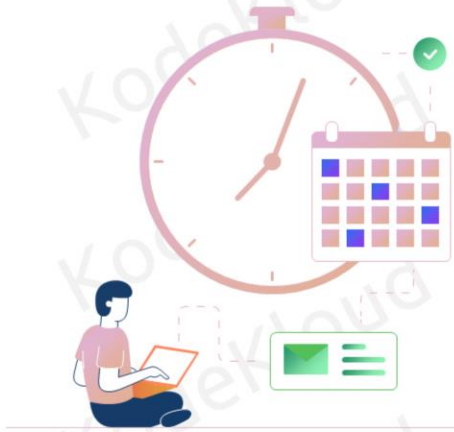
- Customizable monitoring views
- Real-time metrics
- Multi-resource dashboards
- Interactive and responsive
- Permissions and sharing
- Cost-effective monitoring



Create custom monitoring views for AWS resources

Dashboards in CloudWatch

- Customizable monitoring views
- Real-time metrics
- Multi-resource dashboards
- Interactive and responsive
- Permissions and sharing
- Cost-effective monitoring



Display real-time metrics and alarms

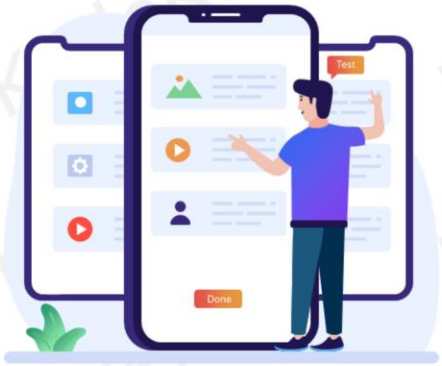
Dashboards in CloudWatch

- Customizable monitoring views
- Real-time metrics
- **Multi-resource dashboards**
- Interactive and responsive
- Permissions and sharing
- Cost-effective monitoring

Aggregate data from various sources into a single view

Dashboards in CloudWatch

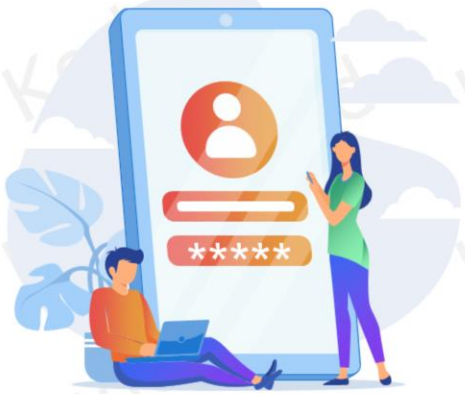
- Customizable monitoring views
- Real-time metrics
- Multi-resource dashboards
- **Interactive and responsive**
- Permissions and sharing
- Cost-effective monitoring



Easily analyze data and zoom in on specific time ranges

Dashboards in CloudWatch

- Customizable monitoring views
- Real-time metrics
- Multi-resource dashboards
- Interactive and responsive
- **Permissions and sharing**
- Cost-effective monitoring



Control access and share with team members

Dashboards in CloudWatch

- Customizable monitoring views
- Real-time metrics
- Multi-resource dashboards
- Interactive and responsive
- Permissions and sharing
- Cost-effective monitoring

Cost-effective to use, with pricing based on the number of widgets and customizations rather than the dashboard itself

Different types of Visualizations

Visualization – Types

Graphs and Charts

- Time Series
- Status History
- Bar and Pie Chart
- Heat Map

Stats and Numbers

- Stat
- Bar Gauge

Misc

- Table
- Logs

Widgets

- Text
- Alert List

CloudWatch Logs and Agents

CloudWatch Logs

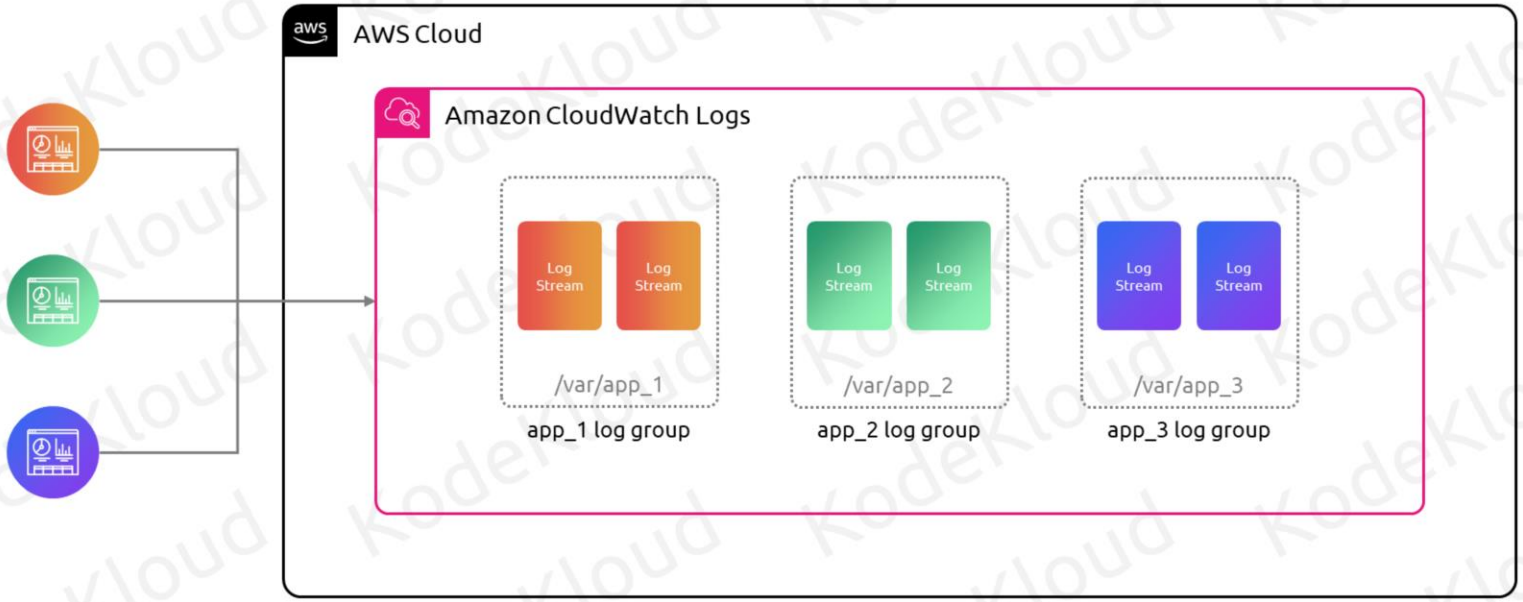
Cloudwatch Logs, a pivotal part of AWS monitoring service offers you the ability to monitor, store and access your log files from your applications not just that from other AWS services too. Whereas Cloudwatch Agents are your key tools in collecting and sending logs and metrics from your application which can be customized. Together, Cloudwatch logs and Agents form a dynamic duo, empowering you to maintain the health and efficiency of your application and have all the logs that is required for you or any kind of security or application analysis. Let us now have a closer look into Cloudwatch Logs, a core component of the AWS monitoring service.

CloudWatch Logs

Cloudwatch Logs and Agents

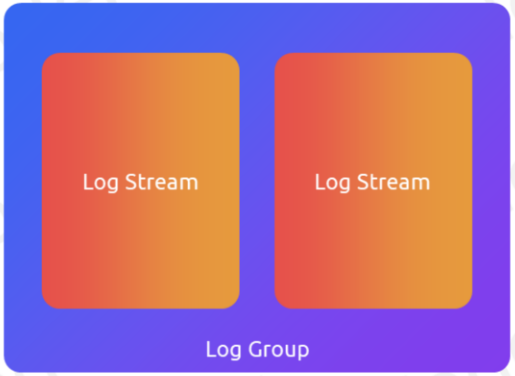
Cloudwatch Logs, a pivotal part of AWS monitoring service offers you the ability to monitor, store and access your log files from your applications not just that from other AWS services too. Whereas Cloudwatch Agents are your key tools in collecting and sending logs and metrics from your application which can be customized. Together, Cloudwatch logs and Agents form a dynamic duo, empowering you to maintain the health and efficiency of your application and have all the logs that is required for you or any kind of security or application analysis. Let us now have a closer look into Cloudwatch Logs, a core component of the AWS monitoring service.

CloudWatch Logs



Understanding Log Stream and Log Group

Understanding Log Stream and Log Group

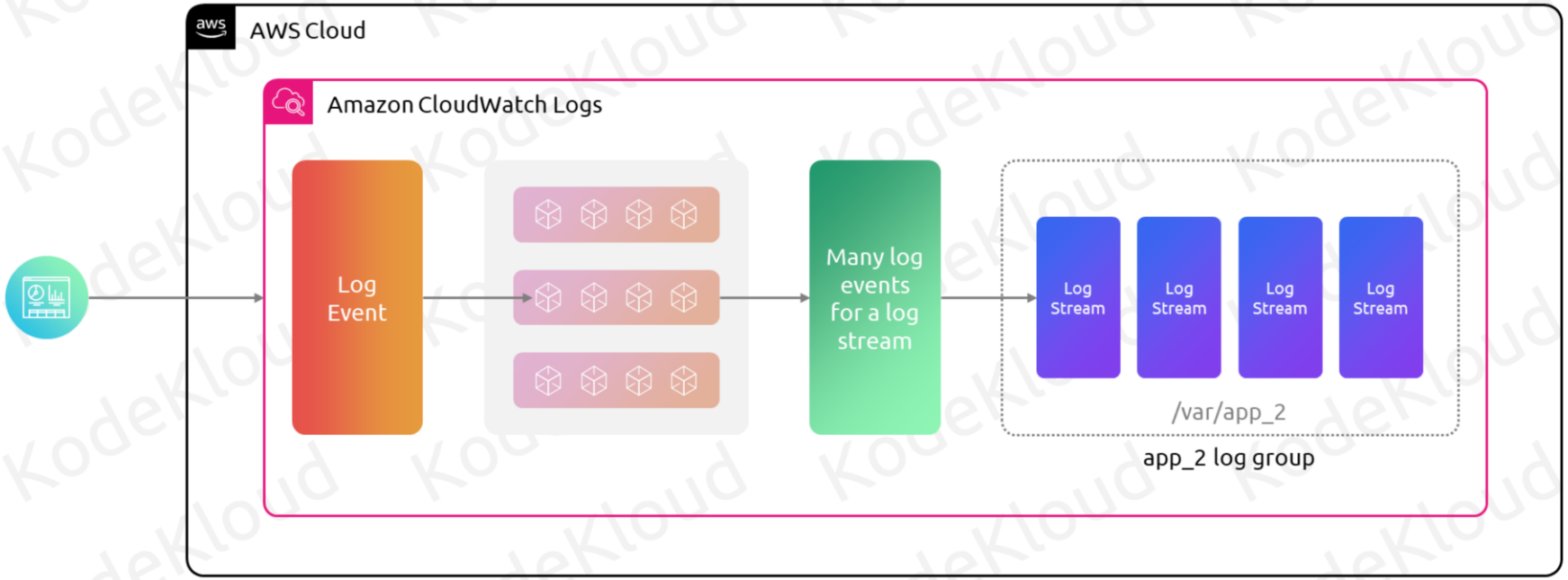


AWS CloudWatch Log Stream: Groups log events from the same source within a Log Group

AWS CloudWatch Log Group: Container for log streams with shared settings

Understanding Log Events

Understanding Log Events



Understanding Log Events

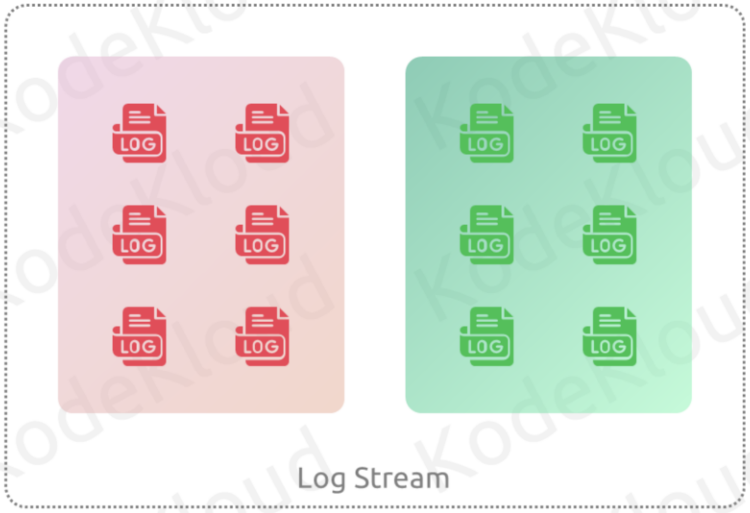
Log events


```
{
    "id": "12345678-1234-1234-1234-123456789012",
    "timestamp": 1677647621000,
    "message": "User login successful: username=johndoe",
    "logGroupName": "/aws/ec2/my-application",
    "logStreamName": "2023/10/20/instance-i-
              0abcd1234efgh5678",
    "source": "my-application",
    "instanceId": "i-0abcd1234efgh5678",
    "eventSource": "application",
    "eventType": "UserLogin",
    "applicationVersion": "1.0.0",
    "region": "us-east-1"
}
```


Each entry of Log stream is a single log event.

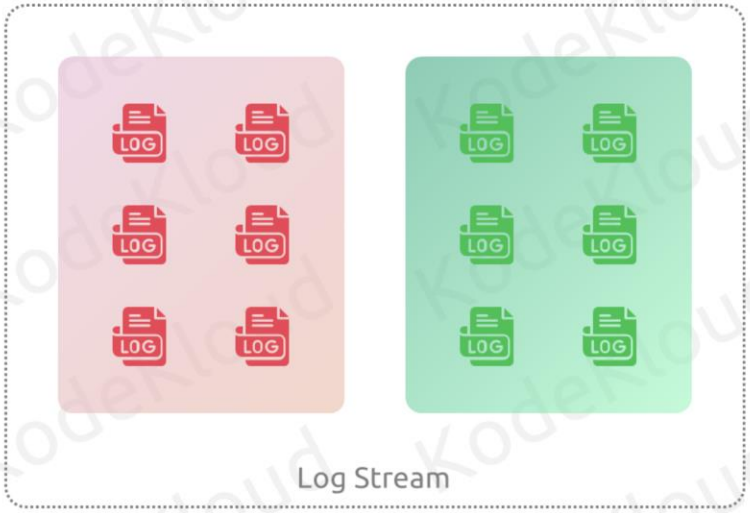
It includes error, info, or other log messages from your application or system.

Understanding Log Events



Structured and Unstructured Log events

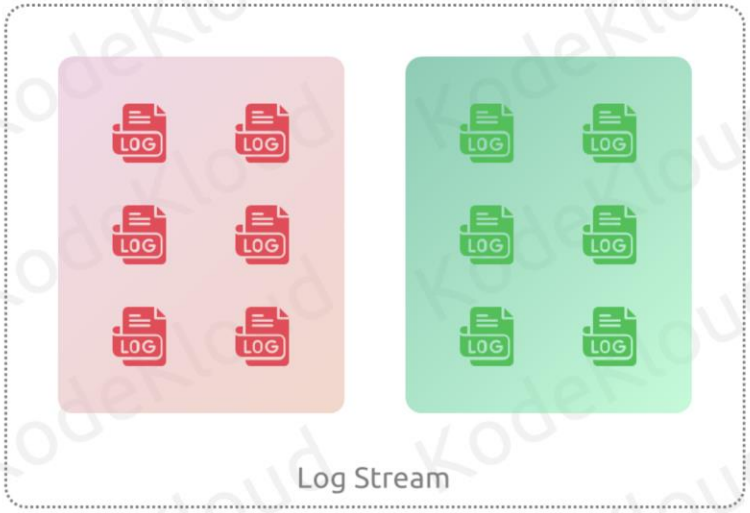
Understanding Log Events



Immutable Log Events

- Log event sent to CloudWatch Logs is immutable
- Cannot be deleted or modified
- Ensure integrity and reliability of log data

Understanding Log Events

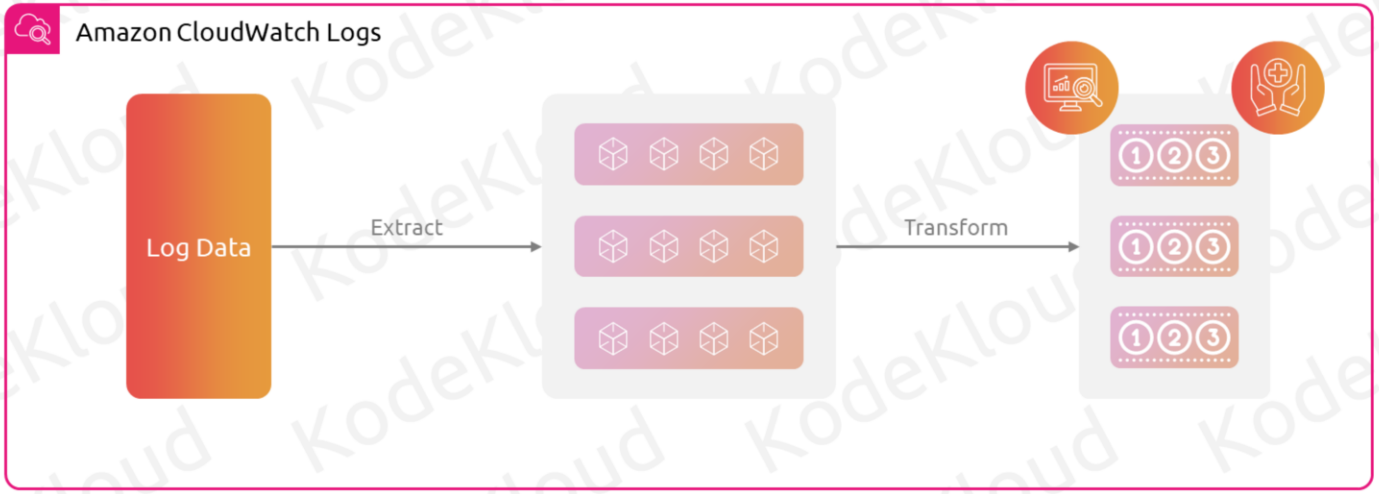


Pre-Ingestion Filtering

- Filter and process log data
- Done on client side
- Done before sending to CloudWatch Logs

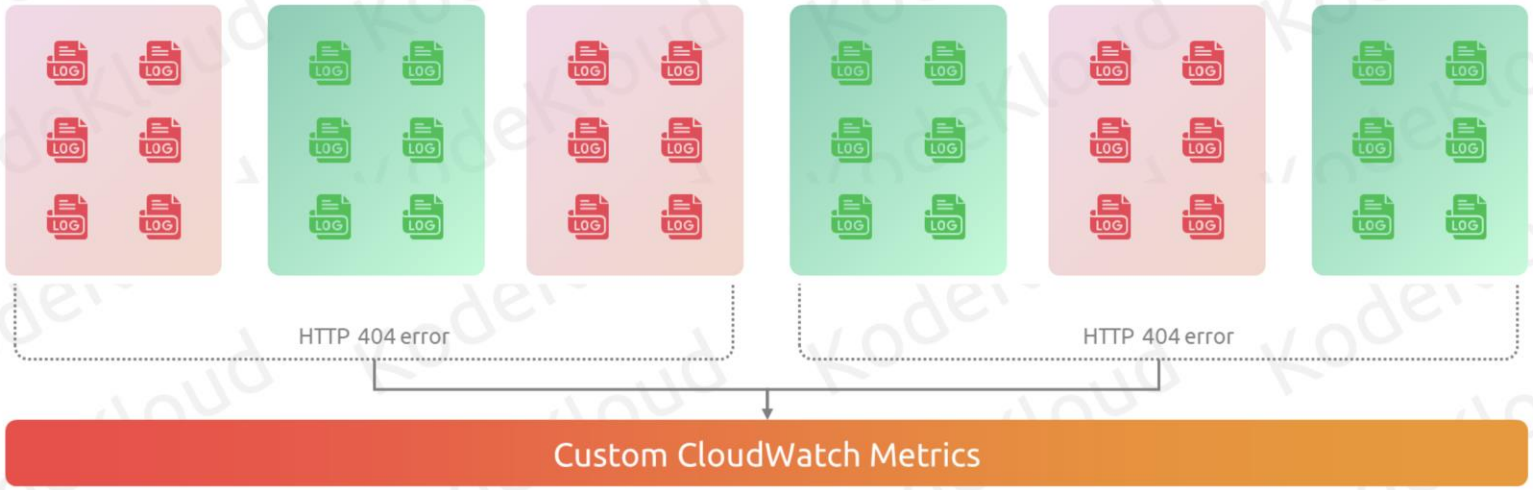
Metric Filters

Metric Filters



How do I basically create a filter on top of this and this is what is metric filter. Metric filters in AWS Cloudwatch Logs allows you to extract information from your log data and transform it into numerical metrics. These metrics can be graphed alarm on and used to monitor the performance and health of your application and system. There are basically two types of metric filter filter patterns. These filter matches log events based on their text content. For example, you create a filter pattern to match all the logs event containing the string HTTP 404. Let us start to visualize this now.

Metric Filters

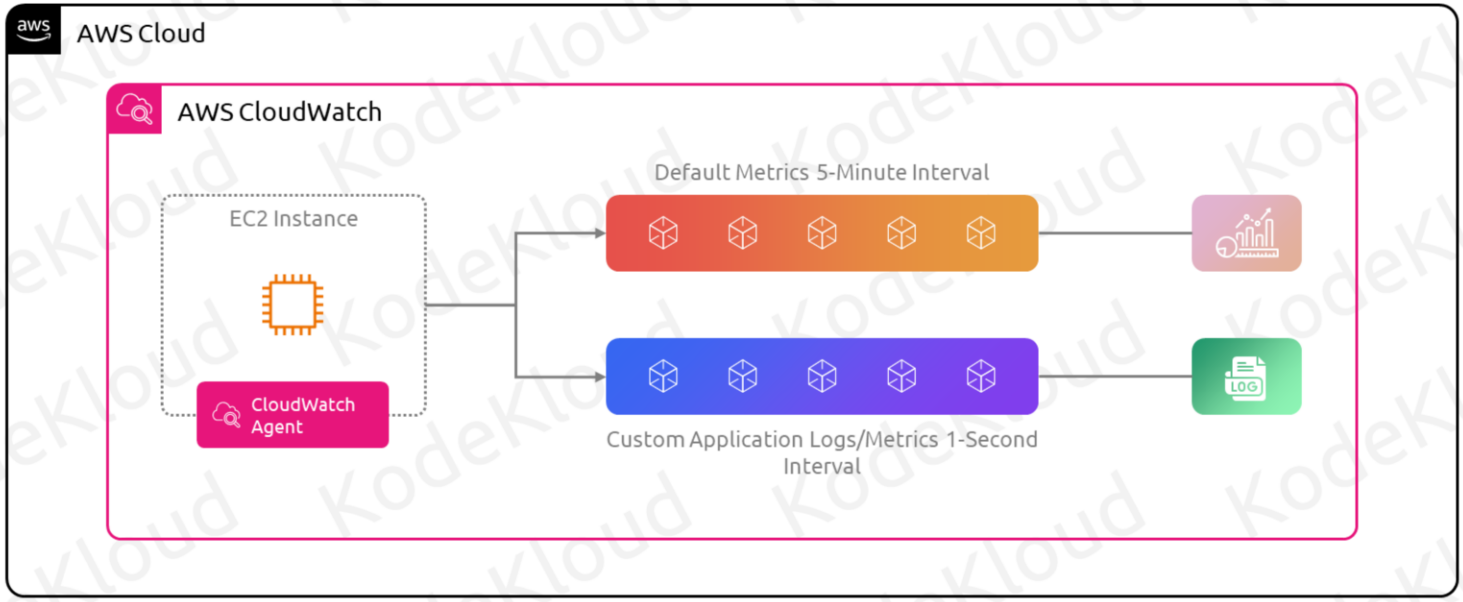


Search and Parse CloudWatch Logs data into metrics

Create Custom CloudWatch Metrics from log data for alarms and monitoring

What is Cloudwatch Agent

CloudWatch Agent



CloudWatch Agent Features

Functionality
Custom metrics
Log collection
Higher resolution data
Custom metrics pricing
Logs pricing
Restrictions and limitations

CloudWatch Agent collects system and app metrics on AWS resources

View metrics in Amazon CloudWatch

EC2 monitoring dashboards offer default metrics (CPU, disk, network)

No additional 3rd party software installation needed

CloudWatch Agent Features

- Functionality
- Custom metrics
- Log collection
- Higher resolution data
- Custom metrics pricing
- Logs pricing
- Restrictions and limitations

CloudWatch Agent collects custom metrics

Enables tailored monitoring

EC2 default monitoring offers predefined metrics

May not cover specific application needs

CloudWatch Agent Features

- Functionality
- Custom metrics
- Log collection
- Higher resolution data
- Custom metrics pricing
- Logs pricing
- Restrictions and limitations

CloudWatch Agent collects and streams logs to CloudWatch Logs

Enables centralized log management

EC2 monitoring dashboards lack built-in log collection and streaming

CloudWatch Agent Features

- Functionality
- Custom metrics
- Log collection
- **Higher resolution data**
- Custom metrics pricing
- Logs pricing
- Restrictions and limitations

CloudWatch Agent enables high-resolution data collection **(1-second intervals)**

Provides more granular insights

EC2 basic monitoring data is at **5-minute intervals**

May not suffice for real-time analysis

CloudWatch Agent Features

- Functionality
- Custom metrics
- Log collection
- Higher resolution data
- **Custom metrics pricing**
- Logs pricing
- Restrictions and limitations

First 10,000 metrics:
$0.30 each/month

Next 750,000 metrics:
$0.05 each/month

Example:
320 metrics, monthly cost is $96

CloudWatch Agent Features

- Functionality
- Custom metrics
- Log collection
- Higher resolution data
- Custom metrics pricing
- Logs pricing
- Restrictions and limitations

Ingested logs cost:
$0.50 per GB

Example:
320 metrics, average 38 KB/metric/hour

Monthly log cost:
$4.23

Total CloudWatch cost:
$100

CloudWatch Agent Features

- Functionality
- Custom metrics
- Log collection
- Higher resolution data
- Custom metrics pricing
- Logs pricing
- Restrictions and limitations

Log Size Limits

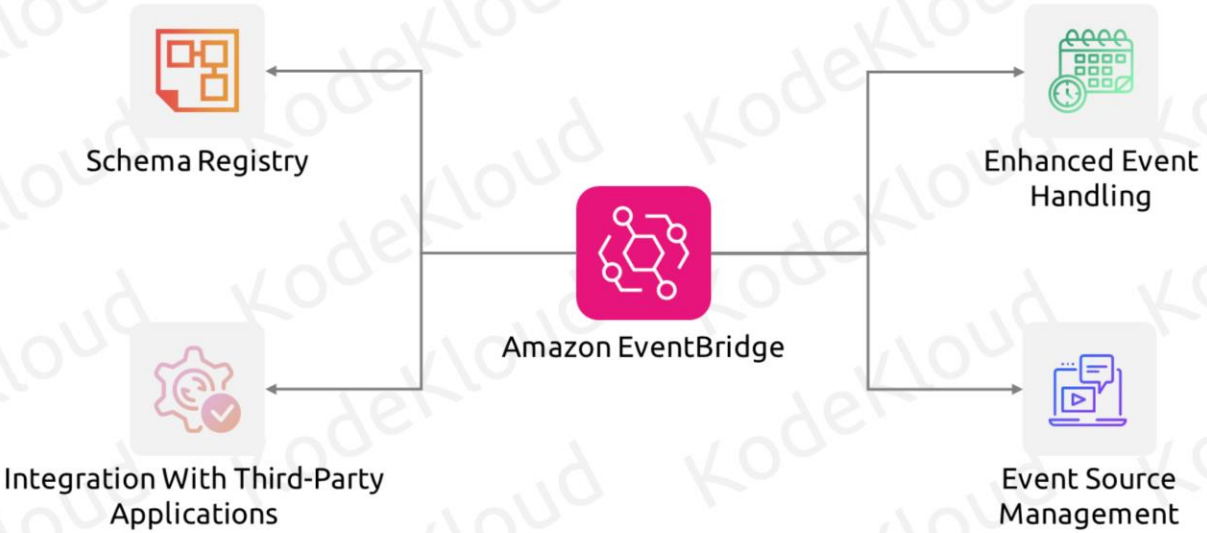
- Each log event: **Max 256 KB**
- Batch size limit: **1 MB**
- Agent skips events exceeding **256 KB**

Installation Requirements

- CloudWatch Agent via Systems Manager Run Command
- SSM Agent version: **2.2.93.0 or later**

CloudWatch Events, EventBridge, and Event Buses

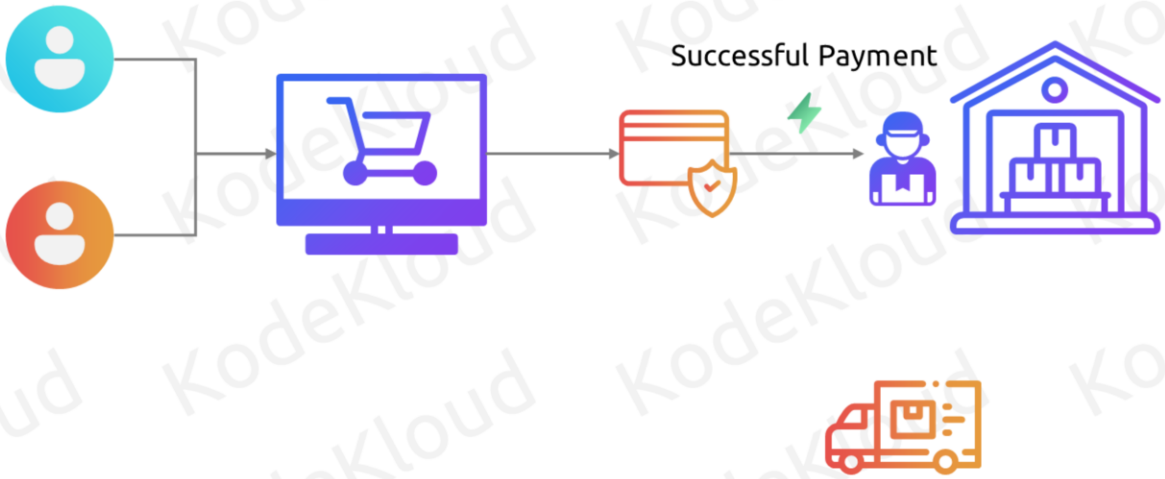
CloudWatch Events is now AWS EventBridge



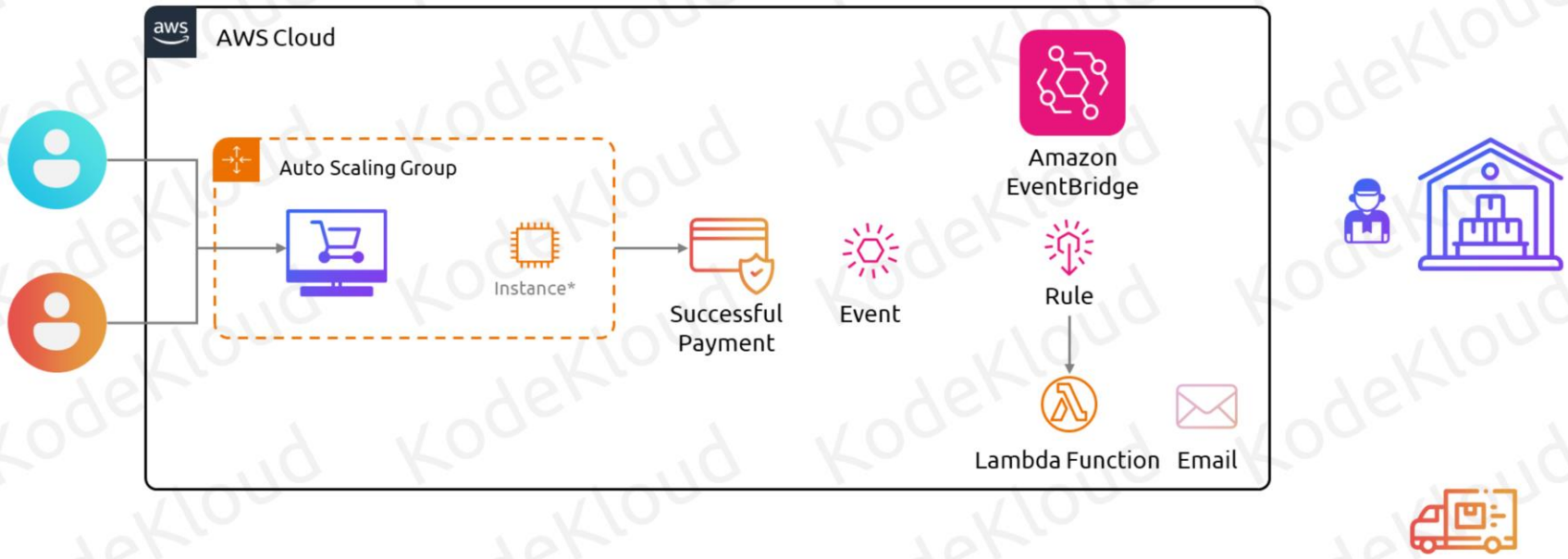
AWS EventBridge is built to handle a broader spectrum of events compared to CloudWatch Events

Enhanced Monitoring and Debugging

EventBridge – Real-World Use Case



EventBridge – Real-World Use Case



Event, Event Bus and Event Rule

Event, Event Bus, and Event Rule

Event

- JSON changes signals to event bus
- Triggers actions on matched patterns

Default Event Bus

- Routers direct events to targets
- Serverless routers route events across AWS and third-party services

Event Rule

- Routes events based on patterns or schedules
- Triggers specified actions on matching events

Event Patterns, Scheduled Events and Pipes

Event Patterns, Scheduled Events, and Pipes

Event Patterns

- Filters events by criteria
- Misconfiguration may cause issues

Scheduler

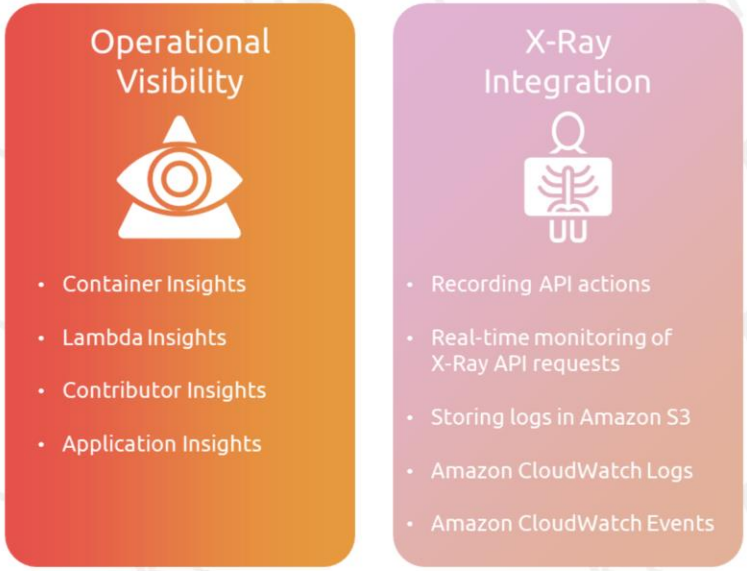
- Uses cron or rate expressions for invocations
- Two types: regular rate and specific time schedules

Pipes

- Connects sources to targets with transformations
- Reduces code, filters events, and removes PII data

CloudWatch Insights and X-Ray

CloudWatch Insights, Operational Visibility, and X-Ray

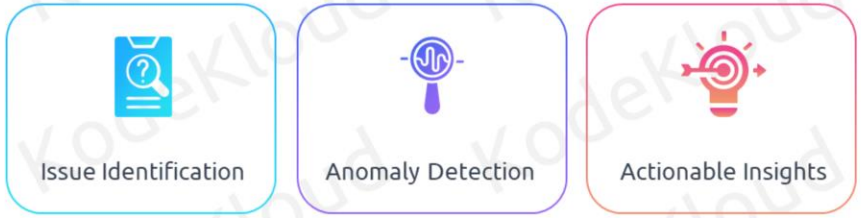


Operational Visibility: CloudWatch provides features like Container Insights, Lambda Insights, Contributor Insights, and Application Insights to gain operational visibility into AWS resources, which help in monitoring, troubleshooting and optimizing AWS resource utilizations

X-Ray Integration: AWS X-Ray integrates with CloudTrail, enabling the recording of API actions, real-time monitoring of X-Ray API requests, and storing logs in Amazon S3, Amazon CloudWatch Logs, and Amazon CloudWatch Events

This will result in Issue Identification, Anomaly Detection, Actionable Insights

CloudWatch Insights, Operational Visibility, and X-Ray



Operational Visibility: CloudWatch provides features like Container Insights, Lambda Insights, Contributor Insights, and Application Insights to gain operational visibility into AWS resources, which help in monitoring, troubleshooting and optimizing AWS resource utilizations

X-Ray Integration: AWS X-Ray integrates with CloudTrail, enabling the recording of API actions, real-time monitoring of X-Ray API requests, and storing logs in Amazon S3, Amazon CloudWatch Logs, and Amazon CloudWatch Events

This will result in Issue Identification , Anomaly Detection , Actionable Insights

Container Insights

Container Insights

Container Insights

01

Performance Monitoring

02

Log Analytics and Troubleshooting

03

Pricing

Performance Monitoring: Container Insights provides detailed monitoring of applications, allowing for real-time analysis of key metrics such as CPU, memory, disk, and network usage across containers, pods, nodes, and clusters.
Log Analytics and Troubleshooting: It facilitates deep log analytics and troubleshooting by aggregating and correlating logs across different services, helping in identifying and resolving issues quickly.
Pricing: Pricing for Container Insights is based on the amount of data ingested and retained. Users pay for the data collected, with costs varying based on the region and the chosen retention period, making it essential to manage data

volume to control costs.

Lambda Insights

Lambda Insights

01

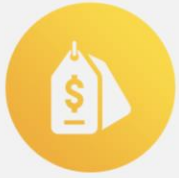
Easy Setup

02

Operational Visibility

03

Monitoring and Optimization



04

Pricing

Easy Setup: Setting up Lambda Insights is straightforward. Users can enable it on a Lambda function via a one-click toggle in the Lambda console. Alternatively, enabling can be done through the AWS CLI, AWS CloudFormation, the AWS Serverless Application Model CLI

Operational Visibility: It enhances operational visibility by automatically collating and summarizing Lambda performance metrics, errors, and logs in prebuilt dashboards, saving users from time-consuming, manual work

Monitoring and Optimization: Lambda Insights allows for the monitoring, troubleshooting, and optimization of AWS

Lambda functions.
Pricing: Pricing for Lambda Insights is based on usage. Users only pay for the metrics and logs reported by Lambda Insights for their functions.

Contributor Insights

01

High-Cardinality Data Analysis

02

Performance Impact Identification

03

Data Aggregation and Discovery

High-cardinality Data Analysis: Analyzes log data to identify top contributors impacting system performance
Performance Impact Identification: Pinpoints outliers and heavy traffic patterns to assist in issue remediation
Data Aggregation and Discovery: Unearths patterns through enhanced data aggregation in CloudWatch Logs

Application Insights

01

Web Application Monitoring

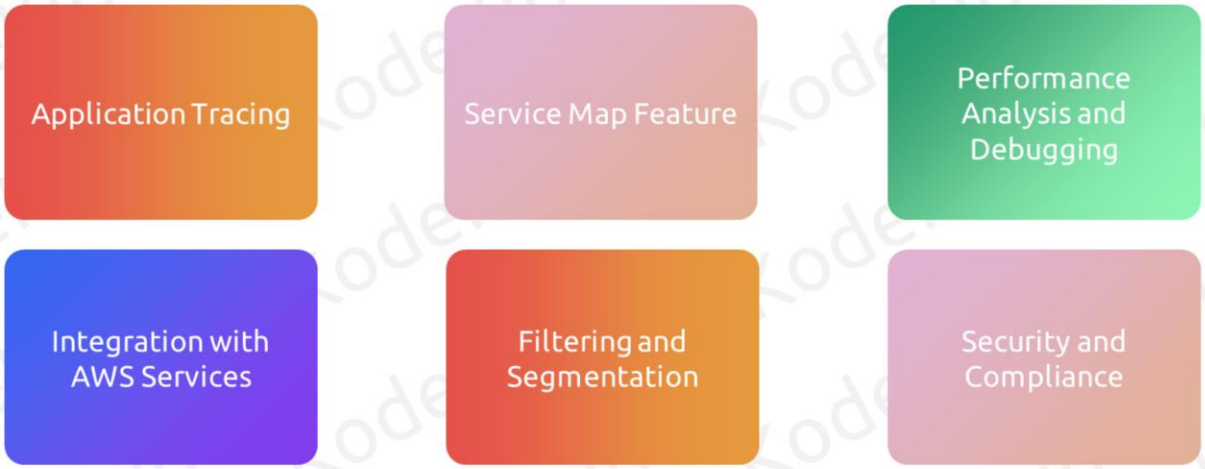
02

Telemetry Data Utilization

Web Application Monitoring: Monitors and analyzes web application performance across various platforms
Telemetry Data Utilization: Leverages telemetry data for proactive monitoring and health overview dashboards

Introduction to AWS X-Ray and Service Map

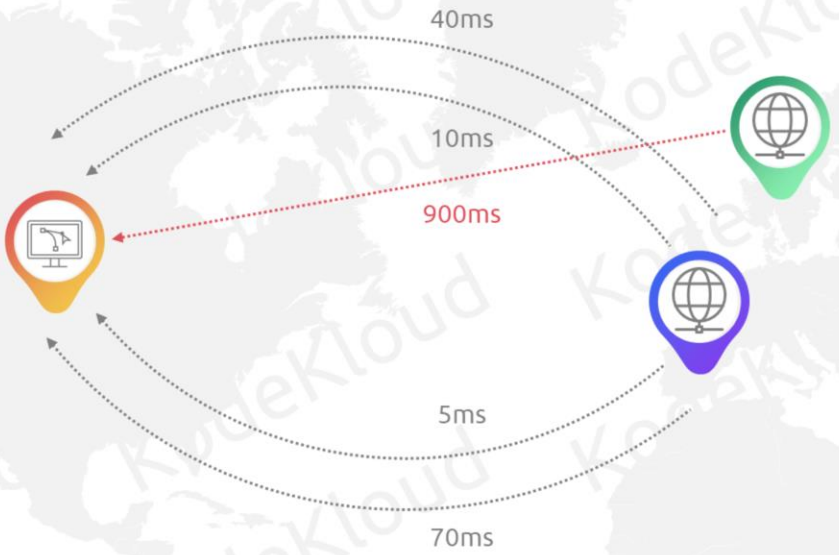
AWS X-Ray and Service Map – Introduction



Advanced Observability With CloudWatch

Internet Monitor

Internet Monitor



Internet Monitor

Monitors network flow data for **traffic patterns** and IP addresses

Enables **anomaly detection** for unexpected traffic spikes

Integrates with other AWS services for enhanced monitoring

Provides **real-time visibility** into network activity

Allows setting **alarms** based on specific network thresholds

Supports **VPC flow logs** to capture and analyze traffic data

Synthetics Canaries

Synthetics Canaries – Going Back in History (1986)

Coal Mine


© Copyright KodeKloud


CH4
CH4
CH4
CH4
CH4
CH4
CH4
CH4
© Copyright KodeKloud


CH4
CH4
CH4
CH4
CH4
CH4
CH4
CH4
© Copyright KodeKloud

Synthetics Canaries in AWS CloudWatch

01 | Canaries

- Lightweight and automated
- Configurable scripts for monitoring endpoints and APIs
- Simulate user interactions
- Test application functionality

Synthetics Canaries in AWS CloudWatch

02 | Monitoring

Continuous monitoring with CloudWatch Synthetics

Monitors applications and endpoints

Real-time performance and availability data

Synthetics Canaries in AWS CloudWatch

03 | Integration

- Seamless integration with AWS services
- Includes CloudWatch Alarms and AWS Lambda
- Works with AWS Step Functions
- Enables action triggers based on monitoring results

Synthetics Canaries in AWS CloudWatch

04 | Screenshots and HAR Files

- Canaries capture screenshots and HAR files
- Obtained during script execution
- Offer detailed insights into performance
- Useful for troubleshooting

Synthetics Canaries in AWS CloudWatch

05 | Metrics and Logs

Collect and analyze canary-generated metrics and logs

Help gain deeper insights into application behavior

Synthetics Canaries in AWS CloudWatch

06 | Canary Execution Costs

- AWS charges based on canary runs
- Payment for each run
- Include script execution time
- Data transfer costs are also considered

Synthetics Canaries in AWS CloudWatch

07 | Canary Script Costs

- AWS Lambda incurs additional costs
- Costs depend on resources used
- Duration of canary script influences expenses

Resource Health

Resource Health in AWS CloudWatch

Resource Health automatically discovers and manages the health and performance of your EC2 hosts

- Automated Discovery and Management
- Centralized View of EC2 Hosts
- Customizable Views and Filters
- Grouping and Sorting Capabilities
- Real-time Visualization and Alerts
- Prerequisites for Full Functionality
- Troubleshooting and Performance Analysis
- No Additional Charge for AWS Customers

Evidently

Evidently

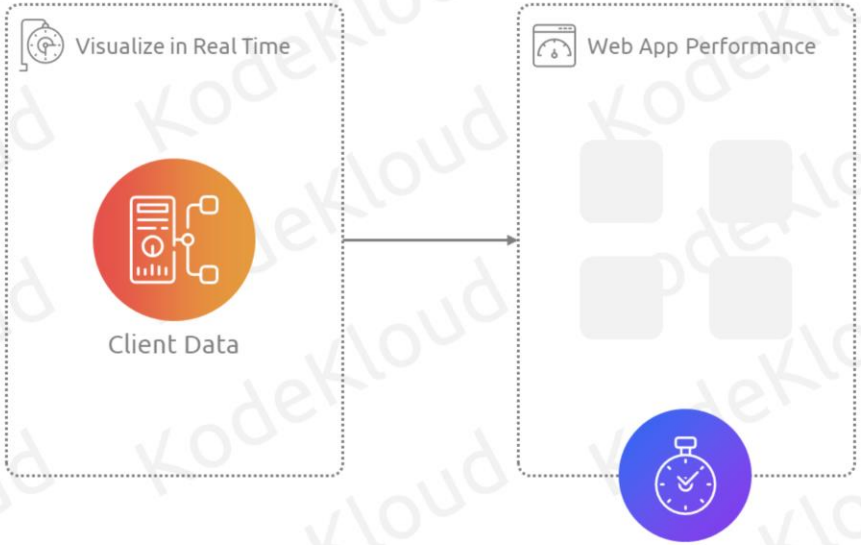
AWS CloudWatch Evidently is a distinct feature within the AWS CloudWatch suite, specifically tailored for optimizing your application's performance and user experience

- Feature Experimentation
- Feature Flags
- User Segmentation
- Real-time Analytics
- Integration and Compatibility
- Visualizations and Dashboards

Real-time User Monitoring (RUM)

Real-Time User Monitoring (RUM)

CloudWatch RUM collects and visualizes client-side data in near real time to help understand web application performance from actual user sessions.



Real-Time User Monitoring (RUM)

Performance Data Visualization

- View aggregated data
- Breakdowns by browsers and devices
- Analyze page load times
- Examine client-side errors
- Understand user behavior

Retention and Export

- Data retention: **30 days**
- Option to send copies to CloudWatch Logs
- Enables longer retention if needed

Web Application Monitoring

- Monitor client-side performance
- Real-time web app monitoring

Understanding CloudWatch Pricing

Understanding CloudWatch Pricing

Metrics Pricing	Dashboard Pricing	Alarms Pricing	Logs Pricing	Events Pricing
Custom Metrics	**Dashboards**	**Alarms**	**Logs**	**Events**
30 custom metrics, with data points reported every minute.	3 dashboards created for different teams.	20 alarms monitoring the critical metrics.	10 GB of log data ingested, with 5 GB stored for the month.	1 million custom events triggered.

A Medium-Sized E-commerce Website Monitoring

Pricing Breakdown Of Ecommerce Website

Metrics Pricing	Dashboard Pricing	Alarms Pricing	Logs Pricing	Events Pricing
Custom Metrics 30 custom metrics, with data points reported every minute.	**Dashboards** 3 dashboards created for different teams.	**Alarms** 20 alarms monitoring the critical metrics.	**Logs** 10 GB of log data ingested, with 5 GB stored for the month.	**Events** 1 million custom events triggered.
Assume **$0.30** per custom metric per month for the first **10,000 metrics**. Total Cost = 30 metrics × $0.30 = **$9.00**	Assume **$3** per dashboard per month. Total Cost = 3 dashboards × $3 = **$9.00**	Assume **$0.10** per alarm per month. Total Cost = 20 alarms × $0.10 = **$2.00**	**Ingestion**: Assume **$0.50** per GB. **Storage**: Assume **$0.03** per GB per month. Total Ingestion Cost = 10 GB × $0.50 = **$5.00** Total Storage Cost = 5 GB × $0.03 = **$0.15**	First **1 million** events per month are free. Total Cost = **$0.00**

Total Monthly Cost: $9.00 (Metrics) + $9.00 (Dashboards) + $2.00 (Alarms) + $5.00 (Log Ingestion) + $0.15 (Log Storage) = **$25.15**

A Medium-Sized E-commerce Website Monitoring

Tips for CloudWatch Cost Optimization

Tips for Cloudwatch Cost Optimization

KodeKloud

Follow us on https://kodekloud.com/ to learn more.