You Need to Know

Everything About AWS CloudWatch

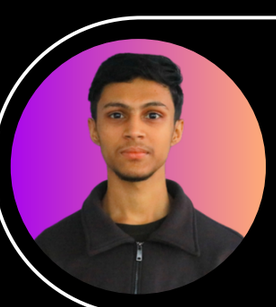
Why Monitoring is important

- **Proactive Problem Detection:** Monitoring allows you to identify potential issues with the application's performance (latency) or stability (outages) before they become severe enough for users to notice.

- **Reduced User Complaints:** By catching problems early, you can prevent them from causing outages or slowdowns that would lead to user frustration and complaints.

- **Continuous Learning:** Monitoring data provides valuable insights into application behavior. By analyzing trends and performance metrics, you can identify areas for improvement and make data-driven decisions about future deployments and optimizations.

In essence, monitoring empowers you to move beyond simply reacting to user complaints and towards a preventative approach that ensures a smooth and efficient user experience.



Rajan Kafle

REPOST

AWS CloudWatch

Amazon CloudWatch is a service specifically designed to address the **monitoring needs** you outlined. Here's a breakdown of how CloudWatch fits in:

- **Unified Monitoring:** CloudWatch collects metrics, logs, and events from all your AWS resources in one place.
- **Alarms and Optimization:** Set alarms to catch issues and optimize resource usage based on monitoring data.
- **Troubleshooting:** Logs and metrics help you troubleshoot and identify root causes of problems.
- **Custom Dashboards:** Visualize key metrics and logs for quick issue identification.

In short, CloudWatch provides a comprehensive monitoring solution that empowers you to proactively manage the health, performance, and cost-efficiency of your AWS applications and resources.

Rajan Kafle

REPOST

CloudWatch Metrics

- **Monitor Everything:** CloudWatch tracks a wide range of metrics for all your AWS services.
- **Track What Matters:** Metrics like CPU utilization and NetworkIn provide insight into resource performance.
- **Organized by Service:** Metrics are grouped by namespaces (e.g., EC2, S3) for easier navigation.
- **Add Context:** Dimensions (like instance ID or environment) provide additional details for specific metrics.
- **Up to 30 Details:** You can add up to 30 dimensions to each metric for a granular view.
- **Track Over Time:** Timestamps allow you to see changes in metrics and identify trends.
- **Visualize Your Data:** Create CloudWatch dashboards to see key metrics at a glance.

Rajan Kafle

REPOST

CloudWatch Custom Metrics

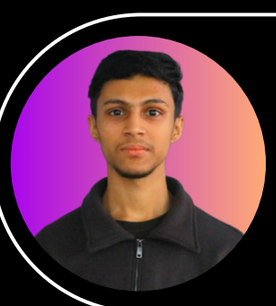
- **Track More Than AWS Metrics:** Monitor things beyond what AWS services provide (memory usage, disk space, logged in users).

- **Send Your Own Data:** Use the PutMetricData API call to send custom data to CloudWatch.

- **Add Labels for Clarity:** Organize your custom metrics with labels (dimensions) like instance ID or environment.

- **Choose How Often to Store Data:** Select standard (1 minute) or high resolution (more frequent, incurs cost) for storing your data.

- **Data Freshness Matters:** CloudWatch accepts data points from the past two weeks and up to two hours in the future. Ensure your EC2 instance time is accurate for proper data collection.



Rajan Kafle

REPOST

CloudWatch Logs

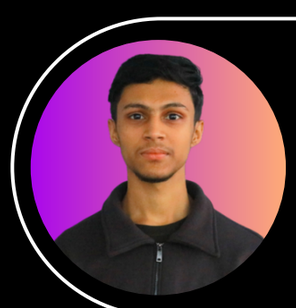
- **Log groups:** Arbitrary name, usually representing an application

- **Log stream:** Instances within application / log files / containers

- Can define log expiration policies (never expire, 1 day to 10 years)

- CloudWatch Logs can send logs to:

- Logs are encrypted by default
- Can setup KMS-based encryption with your own keys



Rajan Kafle

REPOST

CloudWatch Logs Sources

- **SDK, CloudWatch Logs Agent, CloudWatch Unified Agent**
- **Elastic Beanstalk:** Collection of logs from application
- **ECS:** Collection from containers
- **AWS Lambda:** Collection from function logs
- **VPC Flow Logs:** VPC specific logs
- **API Gateway**
- **CloudTrail** based on filter
- **Route53:** Log DNS queries

Rajan Kafle

REPOST

CloudWatch Logs Insights

- **Unlock Your Logs:** Analyze log data stored in CloudWatch Logs to find specific information or troubleshoot issues.

- **Search with Ease:** Use a built-in query language to pinpoint data (e.g., find a specific IP address).

- **Powerful Analysis:**
  - Count errors, filter events, calculate statistics, and sort results.
  - Discover fields automatically for AWS services and JSON logs.

- **Save and Share:** Save frequently used queries and add them to dashboards for easy monitoring.

- **Broad Reach:** Query logs across multiple log groups, even in different accounts.

- **Not Real-Time:** Designed for in-depth analysis, not for live monitoring.

Rajan Kafle

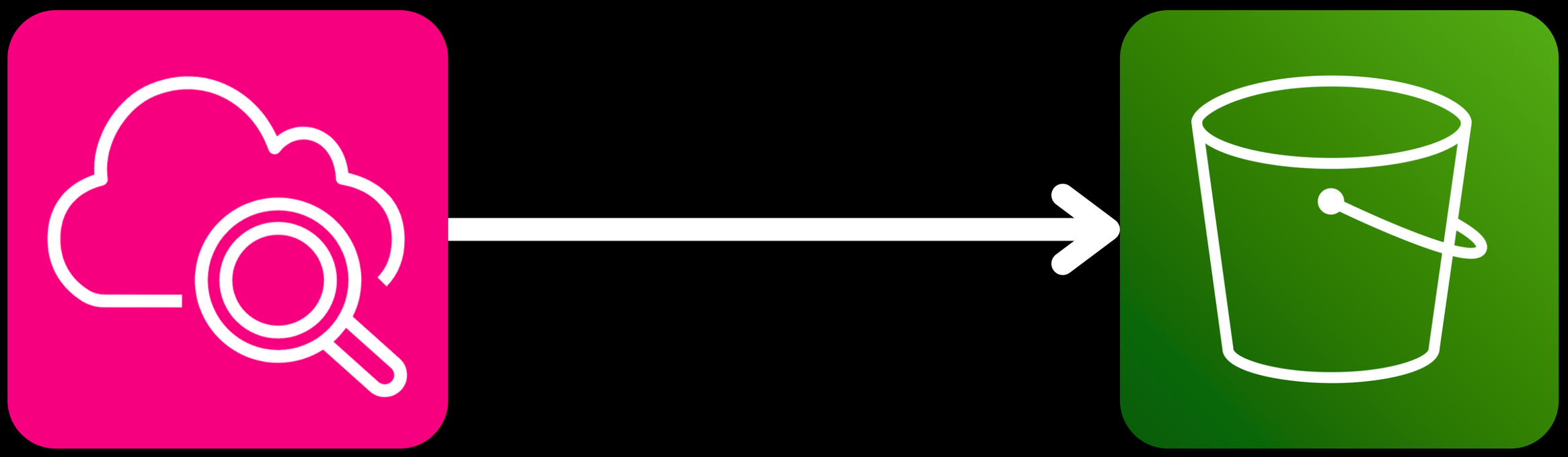
REPOST

CloudWatch Logs S3 Export

- **Export Your Logs:** Move your CloudWatch logs to an S3 bucket for long-term storage and further analysis.

- **Not Instantaneous:** Exported logs may take up to 12 hours to appear in S3.

- **For Archiving, Not Monitoring:** Use CloudWatch Logs Subscriptions for near real-time log delivery if you need to monitor logs actively.



CloudWatch

AWS S3

Rajan Kafle

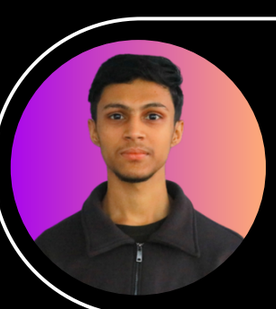
REPOST

CloudWatch Logs Subscriptions

- **Real-Time Logs, Delivered:** Get instant access to your CloudWatch logs for further processing and analysis.

- **Pick Your Destination:** Send your logs to Kinesis Data Streams, Firehose, or Lambda functions for real-time processing.

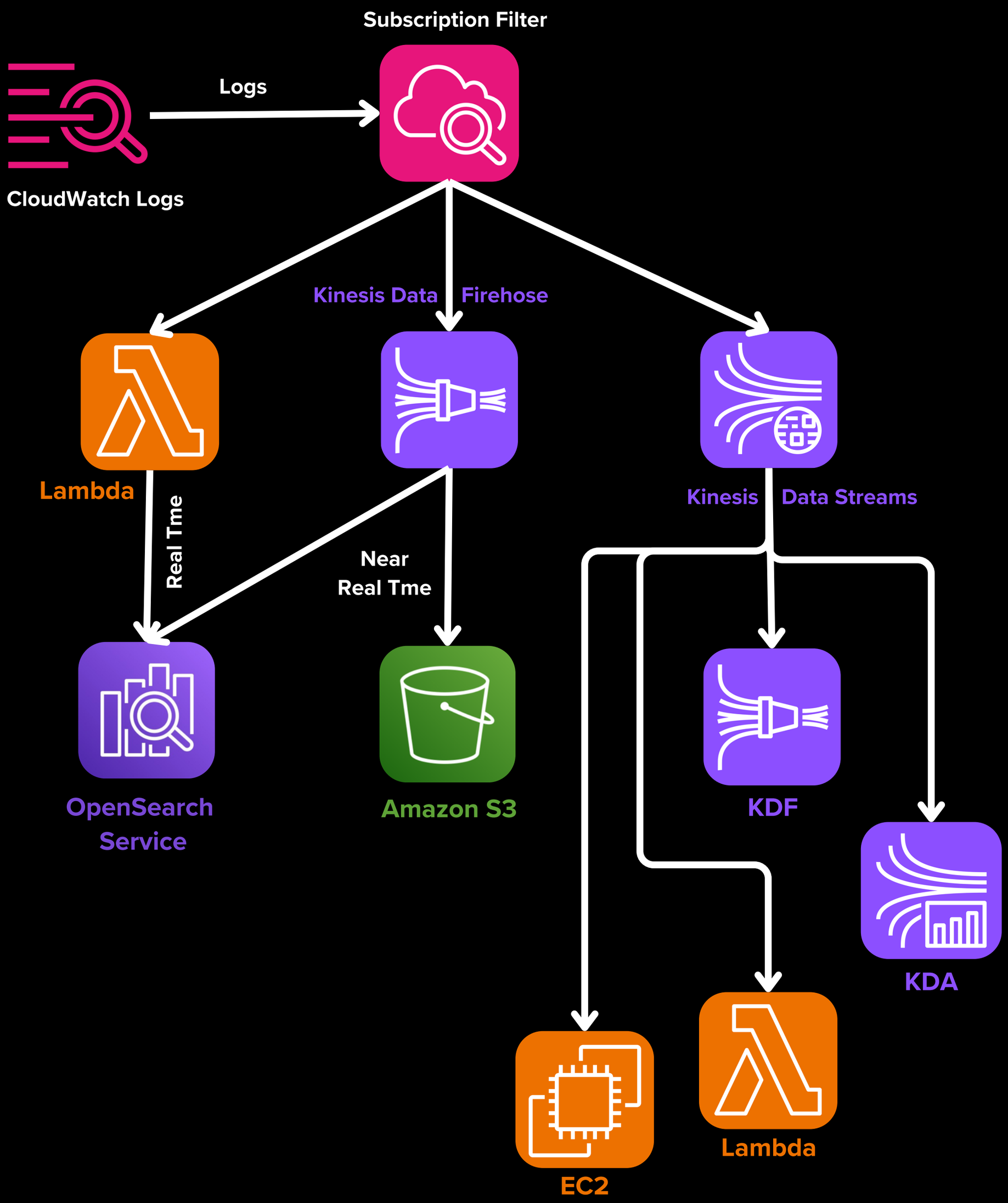
- **Filter What You Send:** Use subscription filters to define which log events get delivered, focusing only on the data you need.

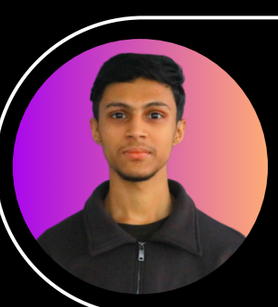


Rajan Kafle

REPOST

CloudWatch Logs Subscriptions





Rajan Kafle

REPOST

CloudWatch Logs Aggregation

Account A
Region 1

CloudWatch Logs

Subscription Filter

Account B
Region 2

CloudWatch Logs

Subscription Filter

Account B
Region 3

CloudWatch Logs

Subscription Filter

Kinesis Data Streams

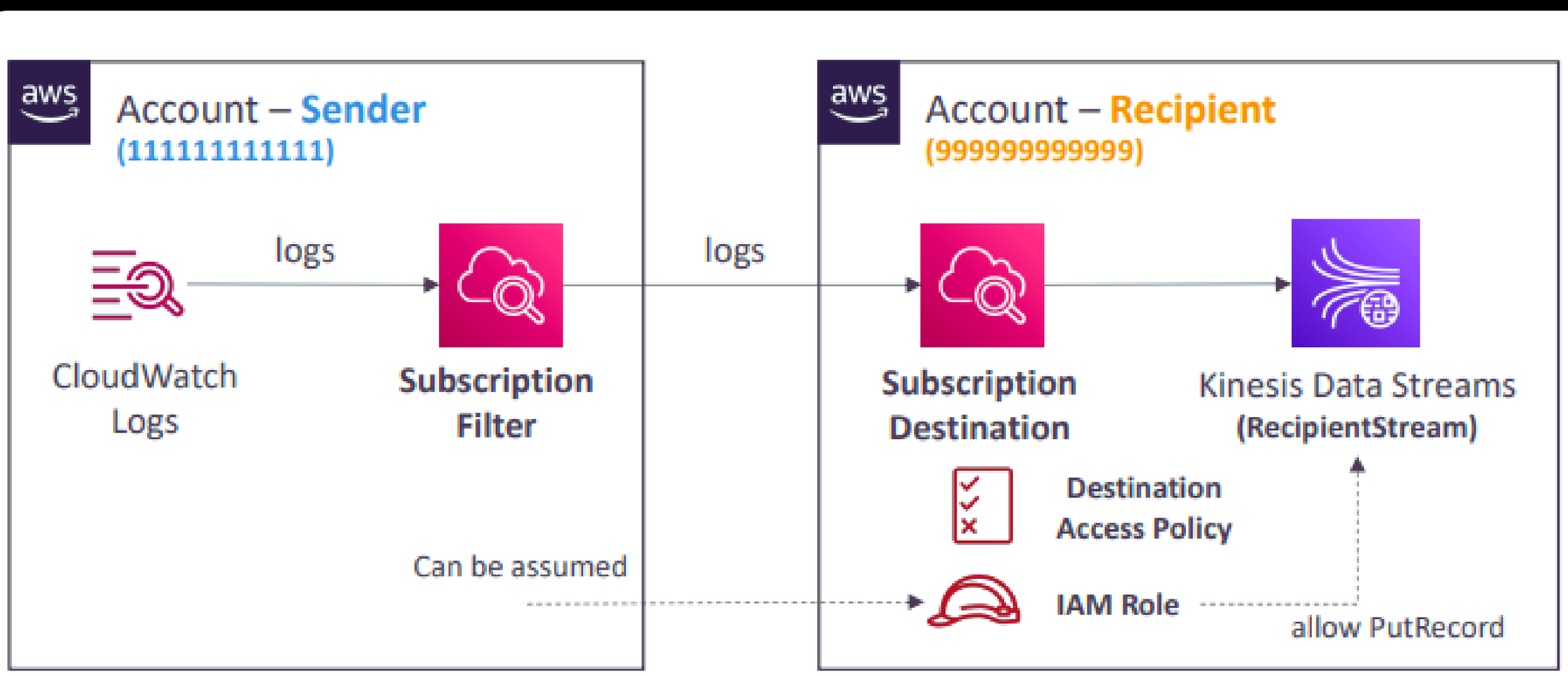
Kinesis Data Firehose

Near Real Time

Amazon S3

CloudWatch Logs Subscription

Cross-Account Subscription: Send log events to resources in a different AWS account (KDS, KDF)



IAM Role (Cross Account)

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "kinesis:PutRecord",
      "Resource": "arn:aws:kinesis:us-east-1:
999999999999:stream/RecipientStream"
    }
  ]
}
```

Destination Access Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "AWS": "111111111111"
      },
      "Action": "logs:PutSubscriptionFilter",
      "Resource": "arn:aws:logs:us-east-1:999999999999:
destination:testDestination"
    }
  ]
}
```

Rajan Kafle

REPOST

CloudWatch Logs & Unified Agent

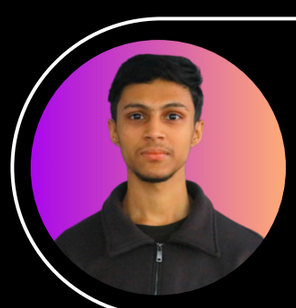
Do you need to send logs from virtual servers (EC2 or on-premise)? Here's a quick guide to CloudWatch agents:

CloudWatch Logs Agent (Legacy):

- **Simpler option:** Designed specifically for sending logs to CloudWatch Logs.
- **Limited functionality:** Focuses solely on log collection.

CloudWatch Unified Agent (Recommended):

- **Do more:** Collects logs and additional system-level metrics (RAM usage, processes).
- **Centralized configuration:** Manage configurations easily using AWS Systems Manager Parameter Store.



Rajan Kafle

REPOST

CloudWatch Unified Agent Metrics

Do you need to send logs from virtual servers (EC2 or on-premise)? Here's a quick guide to CloudWatch agents:

- Collected directly on your Linux server / EC2 instance
- **CPU** (active, guest, idle, system, user, steal)
- **Disk metrics** (free, used, total), Disk IO (writes, reads, bytes, iops)
- **RAM** (free, inactive, used, total, cached)
- **Netstat** (number of TCP and UDP connections, net packets, bytes)
- **Processes** (total, dead, bloqued, idle, running, sleep)
- **Swap Space** (free, used, used %)
- **Reminder:** Out-of-the-box metrics for EC2
  - Disk
  - CPU
  - Network (High Level)

Rajan Kafle

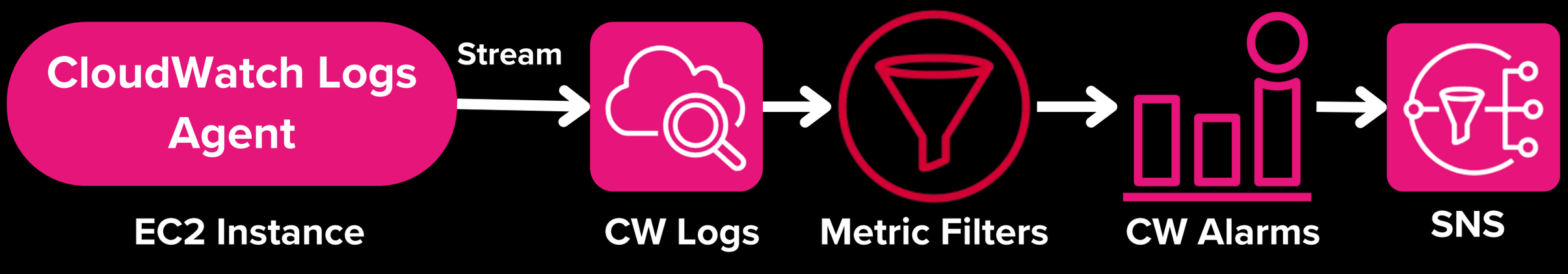
REPOST

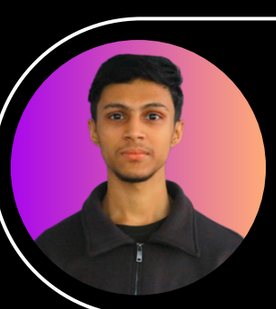
CloudWatch Logs Metric Filter

- CloudWatch Logs can **use filter expressions**
  - For example, find a **specific IP** inside of a log
  - Or count **occurrences of “ERROR”** in your logs
  - Metric filters can be used to **trigger alarms**

- **Filters do not retroactively filter data**. Filters only publish the metric data points for events that happen after the filter was created.

- Ability to specify up to **3 Dimensions** for the Metric Filter (optional)





Rajan Kafle

REPOST

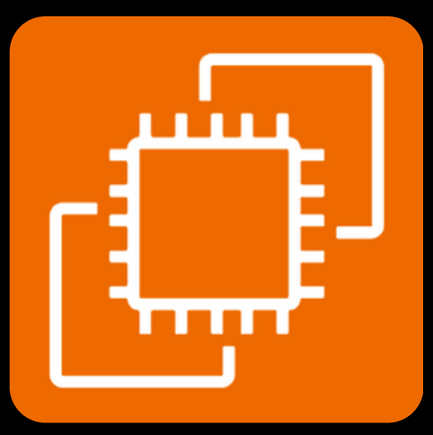
CloudWatch Alarms

- Alarms are used to trigger notifications for any metric
- Various options (sampling, %, max, min, etc...)

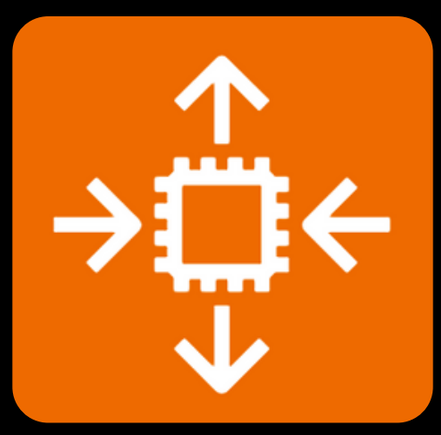
- **Alarm States:**
  - OK
  - INSUFFICIENT_DATA
  - ALARM

- **Period:**
  - Length of time in seconds to evaluate the metric
  - High resolution custom metrics: 10 sec, 30 sec or multiples of 60 sec

CloudWatch Alarm Targates:

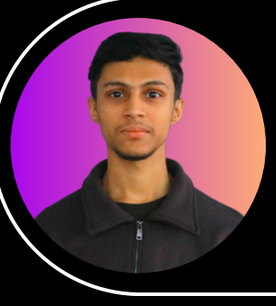


Amazon EC2



EC2 Auto-Scaling

Amazon SNS



Rajan Kafle

REPOST

CloudWatch Composite Alarms

Composite CloudWatch Alarms:

- The powerhouses of alarm monitoring!

- Monitor the health of your system by combining the states (OK, ALARM, INSUFFICIENT_DATA) of multiple regular CloudWatch alarms.

- Leverage logical operators (AND, OR) to define complex alerting conditions.

Benefits of Composite Alarms:

- **Reduce Alarm Noise:** By grouping related alarms into a composite, you get notified only when a critical condition arises, preventing alert fatigue.

- **Create Multi-Factor Monitoring:** Monitor the health of your system from different angles by combining metrics from various resources.

- **Improve Troubleshooting Efficiency:** Identify the root cause of issues faster by analyzing the state of multiple alarms within a composite.

Rajan Kafle

REPOST

REPOST

FOLLOW FOR MORE GUIDES!