Management Tools - Monitoring

CloudWatch

Content Prepared By: Chandra Lingam, Cotton Cola Designs LLC

amazon web services

CloudWatch Architecture

- CloudWatch is a metrics repository
- AWS Services and Custom applications store the metrics into the repository
- CloudWatch Console can calculate statistics based on the metrics and present graphs
- CloudWatch provides data by region – no cross region data aggregation
- Figure: CloudWatch Architecture

Amazon CloudWatch Log

- Monitor, Store, Access log files
- Look for phrases, values or patterns and publish to a custom metric
- Application, System, CloudTrail Logs

Amazon CloudWatch Events

- Delivers a near real-time stream of system events that describe changes in your infrastructure
- Route the events for your custom processing to:
  - Kinesis streams, SNS topics, Amazon Lambda functions
- Example: EC2 Instance State change events, Auto Scaling changes, S3 Object events and so forth

Retention Period

- AWS Launched extended retention in Nov, 2016. Before this all metrics were kept for 14 days.

Interval	Retention
1 minute datapoints	15 days (2 weeks)
5 minute datapoints	63 days (9 weeks)
1 hour datapoints	455 days (65 weeks)

Metrics automatically expire after 15 months (~455 days)

CloudWatch – Related Services

- Simple Notification Service (SNS) – used for sending notifications related to CloudWatch alarms
- Auto Scaling – Auto Scaling can use CloudWatch alarm for scaling EC2 instances based on demand
- CloudTrail – To audit CloudWatch API calls in your account
- Identity and Access Management (IAM) – Control who can access CloudWatch metric data

CloudWatch Concepts

- Namespaces
- Metrics
- Dimensions
- Statistics
- Percentiles
- Alarms

AWS Metrics and Dimensions Reference

CloudWatch - Namespaces

- Namespace is a container for metrics - EC2 namespace, EBS namespace, Your application namespace
- Each metric is associated with a namespace
- Metrics in different namespaces are isolated from each other
- Prevents incorrect aggregation of metrics from different applications
    - Your application is tracking error rate and RDS is tracking database error rate – these are tracking different things and cannot be rolled up in a single statistic like average error

CloudWatch - Namespaces

- AWS Services use the AWS/*service* convention for namespace – *AWS/EC2* for EC2

CloudWatch - Metrics

- Metric is the variable that we are monitoring
- Metric datapoints represent the value of the variable over a period of time
    - CPU Utilization is a metric provided by EC2
    - Time when the utilization was observed is recorded with the metric
    - Time ordered set
- Metrics can be added in any order and at any rate – but retrieved as time ordered set
- Metrics retained for 15 months

CloudWatch - Metrics

- Metrics are uniquely identified by
  - Name (example: CPU Utilization, Disk Read Latency)
  - Namespace (example: AWS/EC2, AWS/EBS)
  - Dimensions (example: instance-id, volume-id)
- Every Metric data point consists of:
  - Timestamp, Namespace, Metric Name, Metric Value, Dimensions, Optional unit of measure (example: Seconds, Megabytes/Second)

Metric - Timestamp

- Metrics must be marked with a timestamp
- Timestamp can be two weeks in the past and up to two hours in the future
- If timestamp is not provided, CloudWatch puts the current time
- Timestamp is in UTC

Metric - Dimensions

- Dimensions provide contextual information for a metric. Key-Value pairs
- CloudWatch treats each unique combination of dimensions as a separate metric; even if metric name is same
- To retrieve statistics, you have to provide dimension combination that was used to publish
- Assign up to 10 dimensions to a metric
- Dimensions are used for filtering

Metric - Dimensions

- For metrics from certain services like EC2, CloudWatch can aggregate data across dimensions
    - Average CPU utilization across all instances
- CloudWatch does not aggregate across dimensions for custom metrics
- Table: Example Dimension Dataset

Metric Statistics

- Aggregate metrics data point over a period of time
- Available Statistics

CloudWatch Alarms

- CloudWatch alarms – Notify or take automated action about trend that is emerging in infrastructure
  - Application Error rate, High CPU utilization, Request Queue length increase and so forth
- For notification to occur, alarm state must change and persists for specified time period. Alarm states are:
  - *OK* – metric is within defined threshold
  - *ALARM* – metric is outside of defined threshold
  - *INSUFFICIENT_DATA* – not enough data available to determine alarm state

CloudWatch Alarms

- Alarm action is invoked once the ALARM state is maintained for specified number of periods
- Subsequent behavior depends on type of action
  - Simple Notification Service notifications – notification is sent once
  - Auto Scaling Notification – Alarm continues to invoke action for every period that alarm remains activated
- Example of alarm state
- Stop EC2 Instance Example

Percentiles

- Relative standing of a value in a data set
- Median or 50th percentile value – indicates a value that is greater than 50% of the values
- 95th percentile – 95% of the data is below this value and 5% is above this value
- Used to isolate anomalies
- CPU utilization – average may hide anomalies; maximum a single value can skew the results; if 95th percentile is very high it can confirm unusually high load
- Can be used for defining alarms

CloudWatch Limits

Table: CloudWatch Limits

Monitoring

CloudWatch Dashboard

- Central Monitoring of resources – across different regions
- Single view for selected metrics
- Operational playbook that provides guidance for team members on how to respond to specific events
- Shared and Common view of critical resources
- Link graphs
- Change refresh interval

CloudWatch

- Gain system wide visibility into
  - Resource utilization
  - Application performance
  - Operational health
  - Automatically react to changes
- View Graphs
- Security - Integrated with Identity and Access Management
  - Limit access to CloudWatch metrics
  - Cannot limit access by resource type

Monitoring

- Maintain Reliability, Availability, Performance
  - EC2 Instances
  - Your Solution
- Collect monitoring data from all parts of your solution - Easier to debug multi-point failure
- Make monitoring a priority
- Automate monitoring as much as possible

Monitoring Plan

- Develop a monitoring plan
  - Goals for monitoring
  - Resources you will monitor
  - Source of monitoring data
  - Frequency of monitoring resources
  - Tools for monitoring
  - Who will perform monitoring tasks
  - Who should be notified when thing go wrong

Monitoring - Establish Baseline

- Establish a baseline for EC2 performance
  - Under different load conditions
  - At different times
- Useful for comparing current performance against baseline
  - Normal pattern
  - Performance anomalies
  - Device methods to address performance issues

EC2 Monitoring

- Basic monitoring – Automatically enabled
  - Seven preselected metrics at five minute frequency
  - Three status check metrics at one minute frequency
  - No additional charge
- Detailed Monitoring – User enabled
  - All metrics provided with Basic monitoring at One minute frequency
  - Additional charge applies
  - Enables aggregation by EC2 AMI ID and Instance Type
- Memory metrics are not provided – need to use a separate script

Automated System Status Check

- Monitors health of underlying AWS systems every minute
- Problem that can cause system status checks to fail:
  - Physical Host software or hardware issues
  - Loss of System Power
  - Loss of Network connectivity
- Requires AWS help to repair or resolve it yourself
  - Stop/Start, Recover to migrate instance to new host
  - Alarm actions for automated recovery or EC2 dashboard for manual observation

Automated Instance Status Checks

- Monitors health of individual instance every minute
- Typically requires your involvement to repair
- Problem that can cause instance status checks to fail:
    - Failed System Checks
    - Incorrect networking or startup configuration
    - Exhausted Memory
    - Corrupted file system
    - Kernel compatibility issues
- Alarm action for automated recovery or EC2 dashboard for manual observation

EC2 Monitoring – Custom Metrics

- Custom metrics are published by scripts or application
- Amazon has provided Perl scripts that can be scheduled to collect metrics on:
  - Memory
  - Disk Linux swap space / Windows page file
  - Disk space utilization
- CloudWatch Logs to monitor log files: Application, System, CloudTrail Logs
  - Publish custom metrics
  - Monitor, Store, Access log files

CloudWatch Alarms

- Monitor a single metric or status check
- Perform an automated action when metric crosses threshold over a time period
- Alarm invokes actions only when there is a sustained state change – ignore minor/temporary blips
- Action taken is notification using SNS or apply Auto Scaling policy or auto recovery and so forth

Status Check Alarms

- Create CloudWatch Alarms to notify you when status check fails
- Optionally, take action to correct the problem:
  - Recover Instance – To move instance to a different physical host – only supported on specific instance types
  - Stop
  - Terminate
  - Reboot
- Configure how long the condition needs to persist before taking action

EC2 Available Metrics

Table: List of Available Metrics

Statistics: Data points can be aggregated by time periods (1 minute to 30 days)

EC2 Monitoring – Recommended Metrics

Table: EC2 items to Monitor and Source

EC2 Monitoring

- Aggregate EC2 instance metrics by:
  - Auto Scaling Group
  - Elastic Load Balancing
  - Provided with both basic and detailed monitoring

Manual Monitoring Tools

- EC2 and CloudWatch Dashboards
- State of your EC2 instances
  - Scheduled Events
  - Status Checks
- Graph of metrics

Scheduled Maintenance Events

- AWS can schedule maintenance events for
  - Reboot
  - Stop/Start
  - Retirement
  - System maintenance (temporary network or power)
- Visible in EC2 Dashboard -> Events
- AWS sends email (primary account) prior to the event with details along with start and end date
- Depending on the issue, you can take action (example: recover instance or stop/start to a new host)

Monitor Other AWS Resources

- Elastic Load Balancers for request count, latency
- EBS Volumes for read, write latency
- RDS instances
- SQS for messages send , received
- SNS for messages published, delivered
- Several other AWS products
- No additional software needs to be installed for monitoring
- No additional charge

Automated Monitoring Tools

AWS Management Pack for Windows Servers

Custom Metrics

Publish Custom Metrics

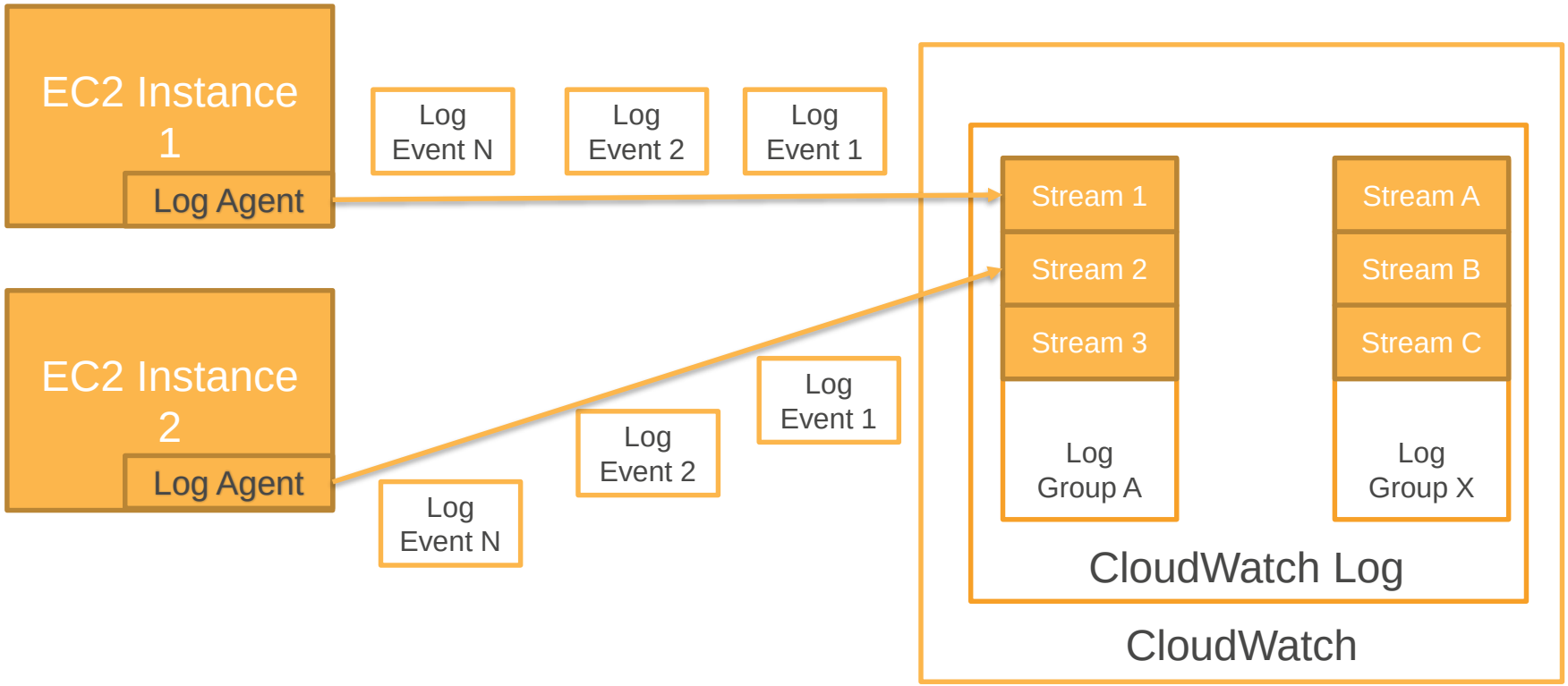
Example: Custom Metrics

CloudWatch Logs

CloudWatch Logs

- Custom application and System log file monitoring capability
  - Monitor for specific terms, count of occurrence
  - Monitor logs from EC2 instance in Real-time
  - Monitor CloudTrail logs for specific events
  - Send to CloudWatch metric when match found
- CloudWatch Log Agent collects log from host and sends to log service – supports rotated and non-rotated files
- Archive Log data to highly durable storage with log retention setting and access when you need it

CloudWatch Log Architecture



CloudWatch Terminologies

- Log Events
- Log Streams
- Log Groups
- Metric Filter
- Retention Settings

Log Events

- Activity recorded by an application or system
  - TimeStamp when event occured
  - Event Message (UTF8 encoded)
- Logical record
- Example: Web server events, CloudTrail events

Log Stream

- Log Streams – Sequence of log events from the same source (application instance, resource)
- Example: Webserver log files on a specific host

Log Groups

- Group of Log Streams
- Shares the same retention, monitoring, and access control settings
- Each log stream belongs to one Log Group
- Fleet of servers generating same type of log

Metric Filters

- Metric Filters convert log file events to CloudWatch data points
- Specify patterns to look for
- Match Terms in: Text, JSON, Space-delimited Log Events
- Assigned to a Log Group
- Log Group can contain one or more metric filters

Retention Settings

- Specify retention period for events kept in CloudWatch logs
- Expired log events are deleted automatically
- Retention is applied to a Log Group which is in-turn applied to their log streams

CloudWatch Log Limits

Table: CloudWatch Log Limits