AWS Monitoring, Troubleshooting & Audit

CloudWatch, X-Ray and CloudTrail

Monitoring in AWS

- AWS CloudWatch:
    - Metrics: Collect and track key metrics
    - Logs: Collect, monitor, analyze and store log files
    - Events: Send notifications when certain events happen in your AWS
- Alarms: React in real-time to metrics / events
- AWS X-Ray:
    - Troubleshooting application performance and errors
    - Distributed tracing of microservices
- AWS CloudTrail:
    - Internal monitoring of API calls being made
    - Audit changes to AWS Resources by your users

AWS CloudWatch EC2 Detailed monitoring

- EC2 instance metrics have metrics "every 5 minutes"
- With detailed monitoring (for a cost), you get data "every 1 minute"
- Use detailed monitoring if you want to more prompt scale your ASG!
- The AWS Free Tier allows us to have 10 detailed monitoring metrics
- Note: EC2 Memory usage is by default not pushed (must be pushed from inside the instance as a custom metric)

AWS CloudWatch Custom Metrics

- Possibility to define and send your own custom metrics to CloudWatch
- Ability to use dimensions (attributes) to segment metrics
  - Instance.id
  - Environment.name
- Metric resolution (StorageResolution API parameter – two possible value):
  - Standard: 1 minute (60 seconds)
  - High Resolution: 1 second – Higher cost
- Use API call PutMetricData
- Use exponential back off in case of throttle errors

AWS CloudWatch Alarms

- Alarms are used to trigger notifications for any metric
- Alarms can go to Auto Scaling, EC2 Actions, SNS notifications
- 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: can only choose 10 sec or 30 sec

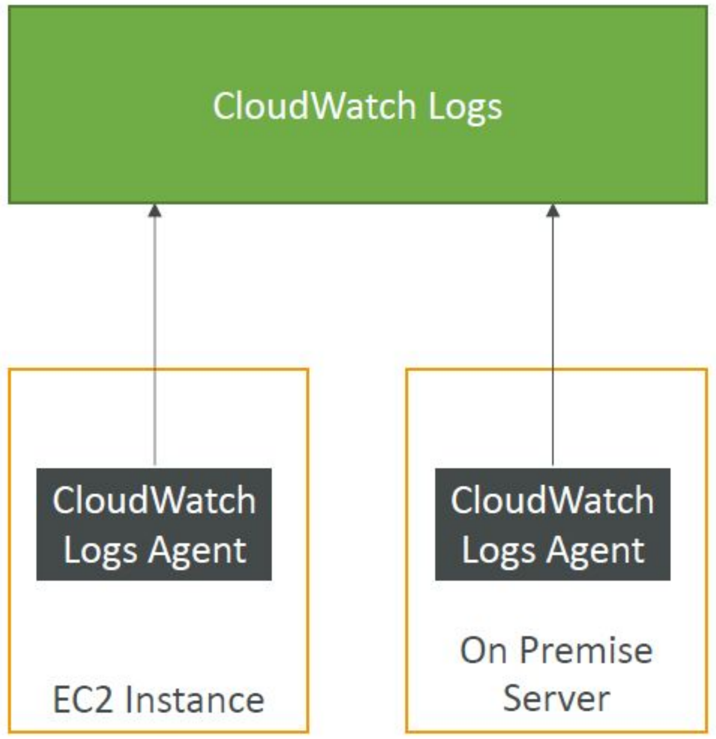
AWS CloudWatch Logs

- Applications can send logs to CloudWatch using the SDK
- CloudWatch can collect log from:
  - 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
  - CloudWatch log agents: for example on EC2 machines
  - Route53: Log DNS queries
- CloudWatch Logs can go to:
  - Batch exporter to S3 for archival
  - Stream to ElasticSearch cluster for further analytics

- CloudWatch logs can use filter expressions
- Can define log expiration policies (never expire, 30 days, etc..)
- Using the AWS CLI we can tail CloudWatch logs
- To send logs to CloudWatch, make sure IAM permissions are correct!
- Security: encryption of logs using KMS at the Group Level

CloudWatch Logs for EC2

- By default, no logs from your EC2 machine will go to CloudWatch
- You need to run a CloudWatch agent on EC2 to push the log files you want
- Make sure IAM permissions are correct
- The CloudWatch log agent can be setup on-premises too



CloudWatch Logs Agent & Unified Agent

- For virtual servers (EC2 instances, on-premise servers…)
- CloudWatch Logs Agent
  - Old version of the agent
  - Can only send to CloudWatch Logs
- CloudWatch Unified Agent
  - Collect additional system-level metrics such as RAM, processes, etc…
  - Collect logs to send to CloudWatch Logs
  - Centralized configuration using SSM- Systems ManagerParameter Store

Metrics

- 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 %)

AWS CloudWatch Events

- Schedule: Cron jobs
- Event Pattern: Event rules to react to a service doing something
  - Ex: CodePipeline state changes!
- Triggers to Lambda functions, SQS/SNS/Kinesis Messages
- CloudWatch Event creates a small JSON document to give information about the change

Amazon EventBridge

- EventBridge is the next evolution of CloudWatch Events

Amazon EventBridge vs CloudWatch Events

- Amazon EventBridge builds upon and extends CloudWatch Events.
- It uses the same service API and endpoint, and the same underlying service
infrastructure.
- EventBridge allows extension to add event buses for your custom applications and your third-party SaaS apps.
- Event Bridge has the Schema Registry capability

AWS X-Ray

- Debugging in Production, the good old way:
    - Test locally
    - Add log statements everywhere
    - Re-deploy in production
- Log formats differ across applications using CloudWatch and analytics is hard.
- Debugging: monolith “easy”, distributed services “hard”
- No common views of your entire architecture!

AWS X-Ray advantages

- Troubleshooting performance (bottlenecks)
- Understand dependencies in a microservice architecture
- Pinpoint service issues
- Review request behavior
- Find errors and exceptions
- Are we meeting time SLA?
- Where I am throttled?
- Identify users that are impacted

X-Ray compatibility

- AWS Lambda
- Elastic Beanstalk
- ECS
- ELB
- API Gateway
- EC2 Instances or any application server (even on premise)

AWS X-Ray Leverages Tracing

- Tracing is an end to end way to following a “request”
- Each component dealing with the request adds its own “trace”
- Tracing is made of segments (+ sub segments)
- Annotations can be added to traces to provide extra-information
- Ability to trace:
    - Every request
    - Sample request (as a % for example or a rate per minute)
- X-Ray Security:
    - IAM for authorization
    - KMS for encryption at rest

AWS X-Ray

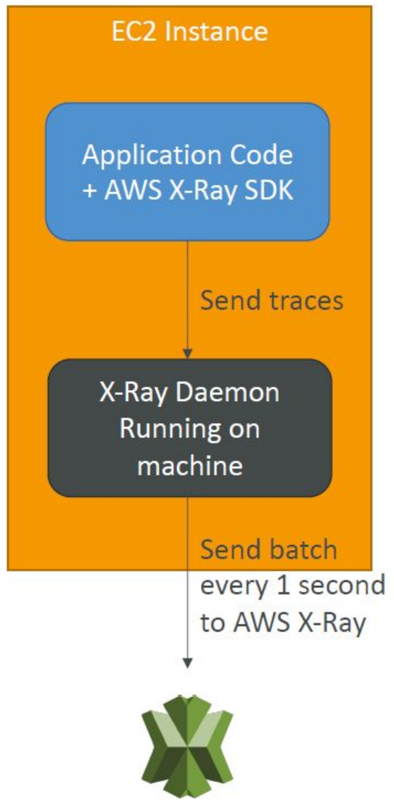
How to enable it?

I) Your code (Java, Python, Go, Node.js, .NET) must import the AWS X-Ray SDK

- Very little code modification needed
- The application SDK will then capture:
- Calls to AWS services
- HTTP / HTTPS requests
- Database Calls (MySQL, PostgreSQL, DynamoDB)
- Queue calls (SQS)

2) Install the X-Ray daemon or enable X-Ray AWS Integration

- X-Ray daemon works as a low level UDP packet interceptor(Linux / Windows / Mac…)
- AWS Lambda / other AWS services already run the X-Ray daemon for you
- Each application must have the IAM rights to write data to X-Ray


EC2 Instance
Application Code + AWS X-Ray SDK
Send traces
X-Ray Daemon Running on machine
Send batch every 1 second to AWS X-Ray

AWS X-Ray Troubleshooting

- If X-Ray is not working on EC2
    - Ensure the EC2 IAM Role has the proper permissions
    - Ensure the EC2 instance is running the X-Ray Daemon
- To enable on AWS Lambda:
    - Ensure it has an IAM execution role with proper policy(AWSX-RayWriteOnlyAccess)
    - Ensure that X-Ray is imported in the code

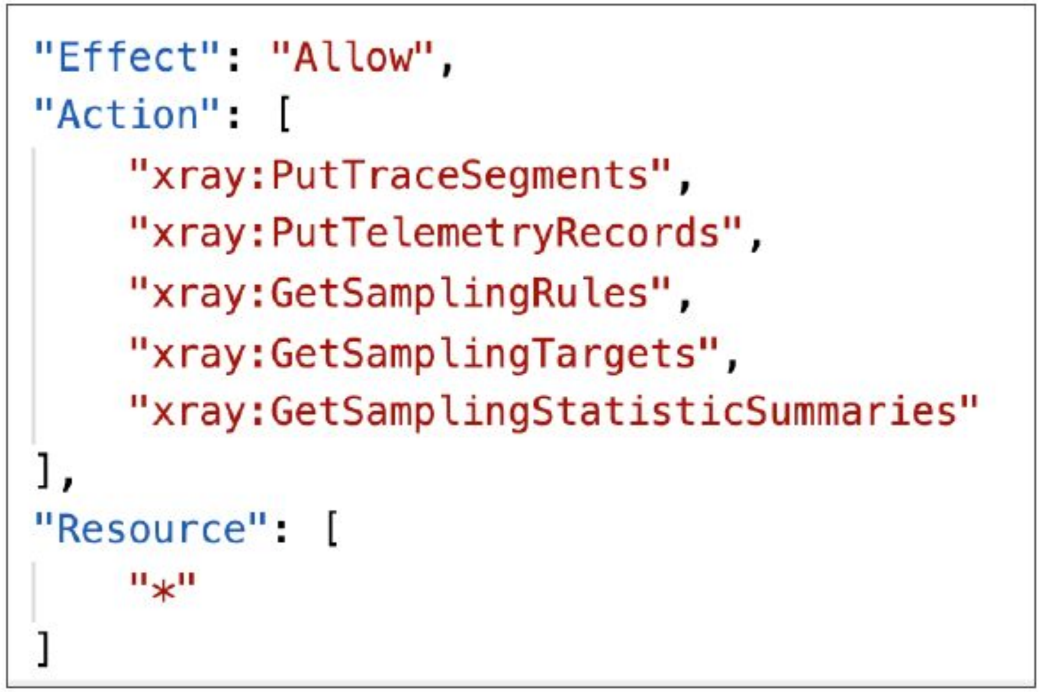
X-Ray Concepts

- Segments: each application / service will send them
- Subsegments: if you need more details in your segment
- Trace: segments collected together to form an end-to-end trace
- Sampling: decrease the amount of requests sent to X-Ray, reduce cost
- Annotations: Key Value pairs used to index traces and use with filters
- Metadata: Key Value pairs, not indexed, not used for searching
- The X-Ray daemon / agent has a config to send traces cross account:
- make sure the IAM permissions are correct – the agent will assume the role
- This allows to have a central account for all your application tracing

X-Ray Sampling Rules

- With sampling rules, you control the amount of data that you record
- You can modify sampling rules without changing your code
- By default, the X-Ray SDK records the first request each second, and five percent of any additional requests.
- One request per second is the *reservoir*, which ensures that at least one trace is recorded each second as long the service is serving requests.
- Five percent is the *rate* at which additional requests beyond the reservoir size are sampled.

X-Ray Write APIs (used by the X-Ray daemon)

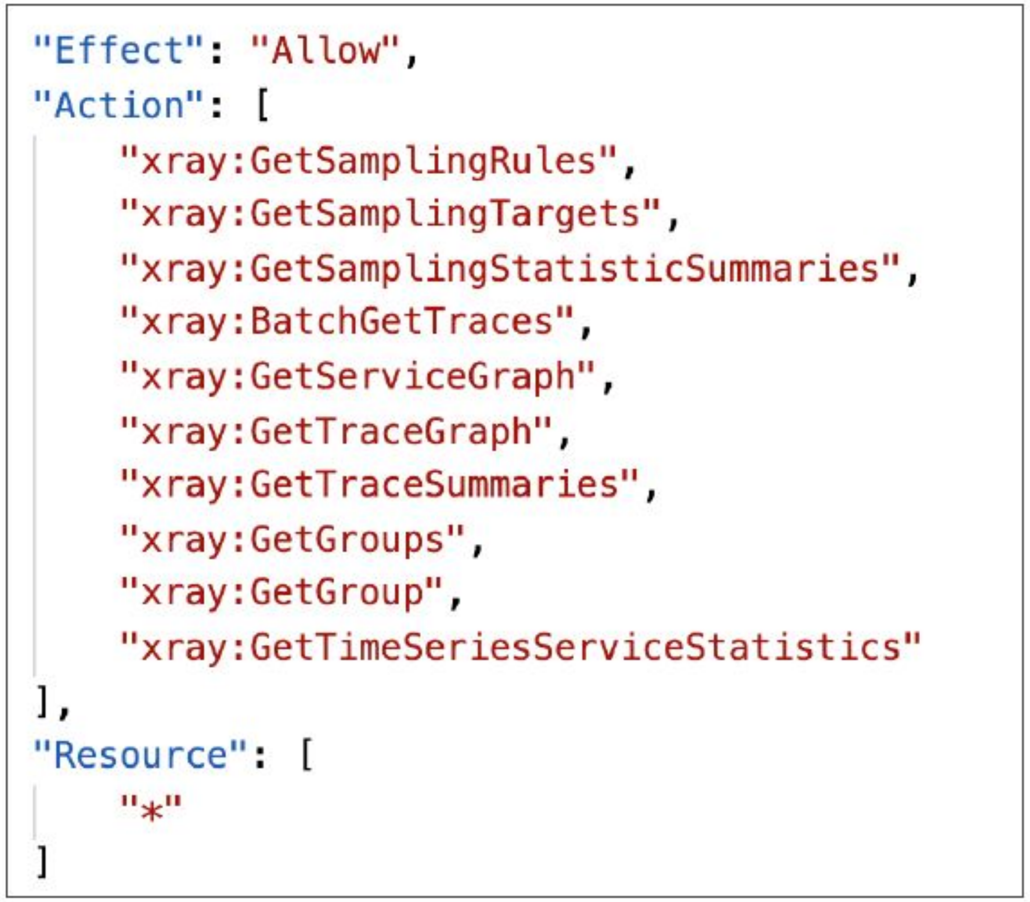


```
"Effect": "Allow",
"Action": [
    "xray:PutTraceSegments",
    "xray:PutTelemetryRecords",
    "xray:GetSamplingRules",
    "xray:GetSamplingTargets",
    "xray:GetSamplingStatisticSummaries"
],
"Resource": [
    "*"
]
```

arn:aws:iam::aws:policy/AWSXrayWriteOnlyAccess

- **PutTraceSegments:** Uploads segment documents to AWS X-Ray
- **PutTelemetryRecords:** Used by the AWS X-Ray daemon to upload telemetry.
  - SegmentsReceivedCount, SegmentsRejectedCounts, BackendConnectionErrors…
- **GetSamplingRules:** Retrieve all sampling rules (to know what/when to send)
- GetSamplingTargets & GetSamplingStatisticSummaries: advanced
- The X-Ray daemon needs to have an IAM policy authorizing the correct API calls to function correctly

X-Ray Read APIs – continued



```
"Effect": "Allow",
"Action": [
    "xray:GetSamplingRules",
    "xray:GetSamplingTargets",
    "xray:GetSamplingStatisticSummaries",
    "xray:BatchGetTraces",
    "xray:GetServiceGraph",
    "xray:GetTraceGraph",
    "xray:GetTraceSummaries",
    "xray:GetGroups",
    "xray:GetGroup",
    "xray:GetTimeSeriesServiceStatistics"
],
"Resource": [
    "*"
]
```

- **GetServiceGraph:** main graph
- **BatchGetTraces:** Retrieves a list of traces specified by ID. Each trace is a collection of segment documents that originates from a single request.
- **GetTraceSummaries:** Retrieves IDs and annotations for traces available for a specified time frame using an optional filter. To get the full traces, pass the trace IDs to BatchGetTraces.
- **GetTraceGraph:** Retrieves a service graph for one or more specific trace IDs.

X-Ray with Elastic Beanstalk

- AWS Elastic Beanstalk platforms include the X-Ray daemon
- You can run the daemon by setting an option in the Elastic Beanstalk console or with a configuration file (in .ebextensions/xray-daemon.config)
- Make sure to give your instance profile the correct IAM permissions so that the X-Ray daemon can function correctly
- Then make sure your application code is instrumented with the X-Ray SDK
- Note: The X-Ray daemon is not provided for Multicontainer Docker

AWS CloudTrail

- Provides governance, compliance and audit for your AWS Account
- CloudTrail is enabled by default!
- Get an history of events / API calls made within your AWS Account by:
    - Console
    - SDK
    - CLI
    - AWS Services
- Can put logs from CloudTrail into CloudWatch Logs
- If a resource is deleted in AWS, look into CloudTrail first!

CloudTrail vs CloudWatch vs X-Ray

- CloudTrail:
  - Audit API calls made by users / services / AWS console
  - Useful to detect unauthorized calls or root cause of changes
- CloudWatch:
  - CloudWatch Metrics over time for monitoring
  - CloudWatch Logs for storing application log
  - CloudWatch Alarms to send notifications in case of unexpected metrics
- X-Ray:
  - Automated Trace Analysis & Central Service Map Visualization
  - Latency, Errors and Fault analysis
  - Request tracking across distributed systems