



AWS X-Ray

- The classic way to debug in Production:
 - Test locally
 - Add log statements everywhere
 - Re-deploy in production
- Log formats differ across applications using CloudWatch and analytics is hard.
- Debugging: monolith is easy, distributed services can get really hard
- No common views of your entire architecture!
- AWS X-Ray helps solve these problems

Advantages of X-Ray

- 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

Compatibility

- AWS Lambda
- Elastic Beanstalk
 - add x-ray config file to `.ebextensions/` folder in your code
- ECS
- ELB
- API Gateway
- EC2 Instances or any application server (even on premise)

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

How to enable X-Ray

1. 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

- 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

X-Ray's internal magic

- X-Ray service collects data from all the different services
- Service map is computed from all the segments and traces
- X-Ray is graphical, so even non technical people can help troubleshoot

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 Instrumentation

- Instrumentation means measure performance of a product, diagnose errors and write trace information
- To instrument an application we can use the X-Ray SDK
- Concepts:
 - **Segments:** A segment records tracing information about a request that your application serves. At a minimum, a segment records the name, ID, start time, trace ID, and end time of the request
 - **Subsegments:** More granularity for segments. We can create subsegments to record calls to AWS services and resources that we make with the AWS SDK, calls to internal or external HTTP web APIs, or SQL database queries. We can also create subsegments to debug or annotate blocks of code in our application. Subsegments can contain other subsegments, so a custom subsegment that records metadata about an internal function call can contain other custom subsegments and subsegments for downstream calls
 - **Trace:** segments and subsegments collected together to form an end-to-end trace
 - **Sampling:** decrease the amount of requests sent to X-Ray in order to reduce costs
 - **Annotations:** key-value pairs added to index traces. This are very important if want our traces to be searchable or we want to filter them
 - **Metadata:** key-value pairs which are not indexed, can not be used for searching and filtering

X-Ray Sampling Rules

- They can be modified before changing code
- By default X-Ray SDK records the first request in each second and 5% of additional requests
- First request per second is called reservoir, ensures that at least 1 request is recorded each second if there is any request sent
- Additional 5% is called the rate which is the amount of additional requests sampled beyond the reservoir size
- Custom sampling rules:
 - We can create our own custom sampling rules with reservoir and rate

X-Ray Write APIs

- **PutTraceSegments:** upload a segment document to AWS X-Ray
- **PutTelemetryRecord:** upload telemetry record to AWS X-Ray. Telemetry examples: SegmentsReceivedCount, SegmentRejectedCount, BackendConnectionErrors.
- **GetSamplingRules:** retrieve all sampling rules (can be useful for X-Ray daemon to know when to sent segments)
- **GetSamplingTargets,** GetSamplingStatisticSummaries

X-Ray Read APIs

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

