

What is AWS Xray?

AWS Xray

What is AWS Xray?

AWS X-Ray is an application performance management service that enables a developer to analyze and debug applications in the Amazon Web Services (AWS) public cloud.

A developer can use AWS X-Ray to visualize how a [distributed application](#) is performing during development or production, and across multiple AWS regions and accounts. AWS X-Ray is compatible with [microservices](#) and serverless-based applications, which can be difficult to debug using conventional methods. AWS X-Ray supports applications that run in Amazon Elastic Compute Cloud ([EC2](#)), Amazon EC2 Container Service ([ECS](#)), AWS [Lambda](#) and AWS [Elastic Beanstalk](#).

AWS X-Ray enables a developer to create a service map that displays an application's architecture, including connections to components and a dependency tree. The service then evaluates response code for each request made to an application and highlights errors or performance issues, such as high latency.

X-Ray uses a trace, a group of segments containing all data points for a single component to track and relay requests to an application. Each segment is comprised of one or more sub-segments that contain data such as queries, time stamps and error statuses. A developer can retrieve trace data 30 seconds after it is collected, and the service stores data for 30 days.

AWS X-Ray supports [AWS Lambda](#), allowing a developer to see how long it takes to execute functions from trigger to termination. The developer can also see calls from the function to other services.

The AWS X-Ray agent enables a developer to send data to the service without using application programming interfaces ([APIs](#)). The agent aggregates information from log files and sends them to X-Ray for analysis and storage.

AWS X-Ray provides a set of query APIs that a developer can use to build custom analysis or visualization apps based on X-Ray records. AWS X-Ray also includes a [software development kit](#) that captures API metadata and includes additional drivers for MySQL and [PostgreSQL](#).

Why should we use Xray?

Currently, if we build and run distributed applications, we have to rely on a per-service or per-resource process to track requests for your application as it travels across various components that make up your application. This problem is further complicated by the varying log formats and storage mediums across frameworks, services, and resources your application runs on or uses. This makes it difficult to correlate the various pieces of data and create an end-to-end picture of a request from the time it originates at the end-user or service to when a response is returned by your application. X-Ray provides a user-centric model, instead of service-centric or resource-centric model, for collecting data related to requests made to our application. This model enables us to create a user-centric picture of requests as they travel across services and resources. By correlating and aggregating data on our behalf, X-Ray enables you to focus on improving the experience for end-users of your application.

What can I do with X-Ray?

X-Ray makes it easy for us to:

- **Create a service map** – By tracking requests made to your applications, X-Ray can create a map of services used by your application. This provides you with a view of connections among services in your application, and enables you to create a dependency tree, detect latency or errors when working across AWS Availability Zones or Regions, zero in on services not operating as expected, and so on.
- **Identify errors and bugs** – X-Ray can automatically highlight bugs or errors in your application code by analyzing the response code for each request made to your application. This enables easy debugging of application code without requiring you to reproduce the bug or error.
- **Build your own analysis and visualization apps** – X-Ray provides a set of query APIs you can use to build your own analysis and visualizations apps that use the data that X-Ray records.

Built for modern applications



Analyze and debug
issues quickly



End-to-end view of
individual services



Identify customer
impact



Cloud agnostic

Core concepts

Trace-

An X-Ray trace is a set of data points that share the same trace ID. For example, when a client makes a request to our application, it is assigned a unique trace ID. As the request makes its way through services in your application, the services relay information regarding the request back to X-Ray using this unique trace ID. The piece of information relayed by each service in our application to X-Ray is a segment, and a trace is a collection of segments.

Segment

An X-Ray segment encapsulates all the data points for a single component (for example, authorization service) of the distributed application. Segments include system-defined and user-defined data in the form of annotations and are composed of one or more sub-segments that represent remote calls made from the service. For example, when your application makes a call to a database in response to a request, it creates a segment for that request with a sub-segment representing the database call and its result. The sub-segment can contain data such as the query, table used, timestamp, and error status.

Annotation

An X-Ray annotation is system-defined or user-defined data associated with a segment. A segment can contain multiple annotations. System-defined annotations include data added to the segment by AWS services, whereas user-defined annotations are metadata added to a segment by a developer. For example, a segment created by your application can automatically be injected with region data for AWS service calls, whereas we might choose to add region data ourselves for calls made to non-AWS services.

Errors-

X-Ray errors are system annotations associated with a segment for a call that results in an error response. The error includes the error message, stack trace, and any additional information (for example, version or commit ID) to associate the error with a source file.

Sampling-

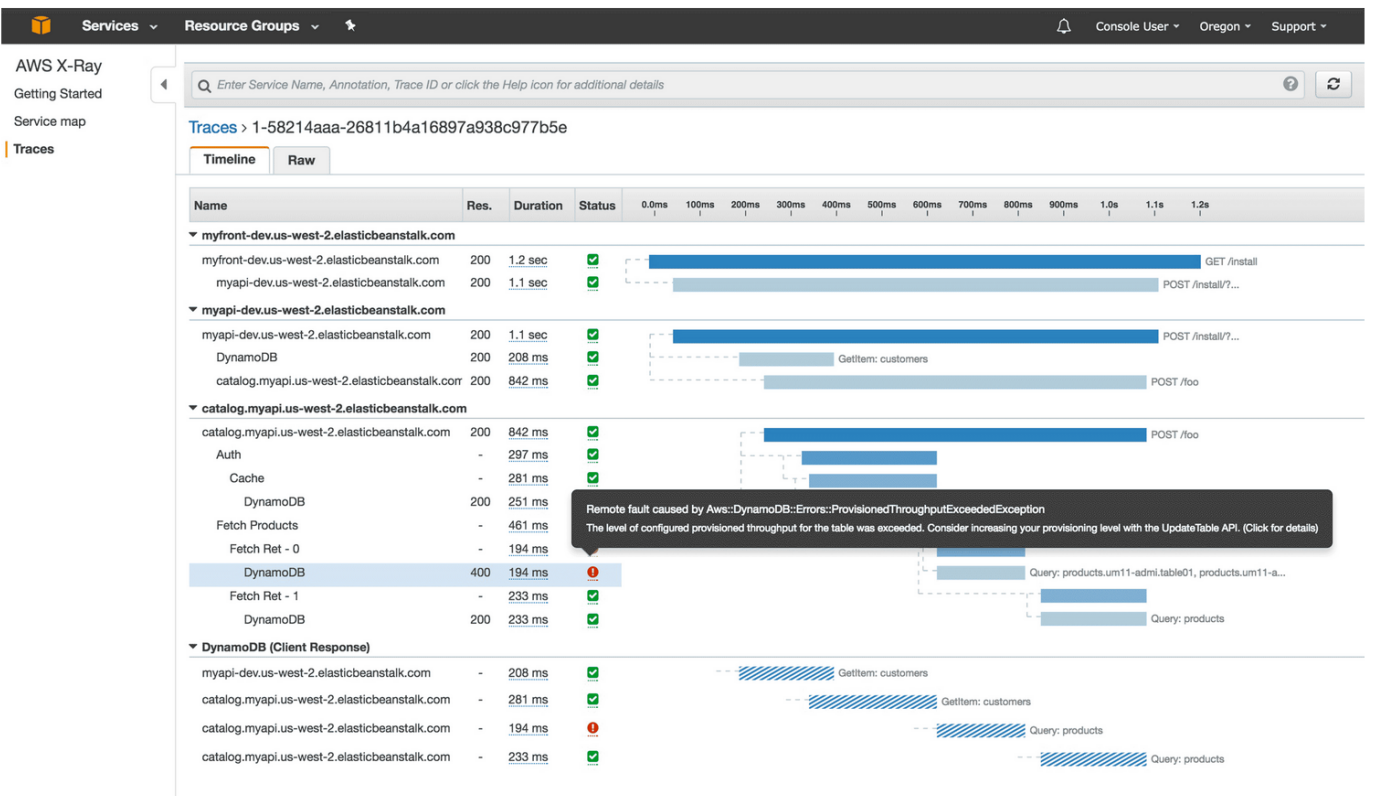
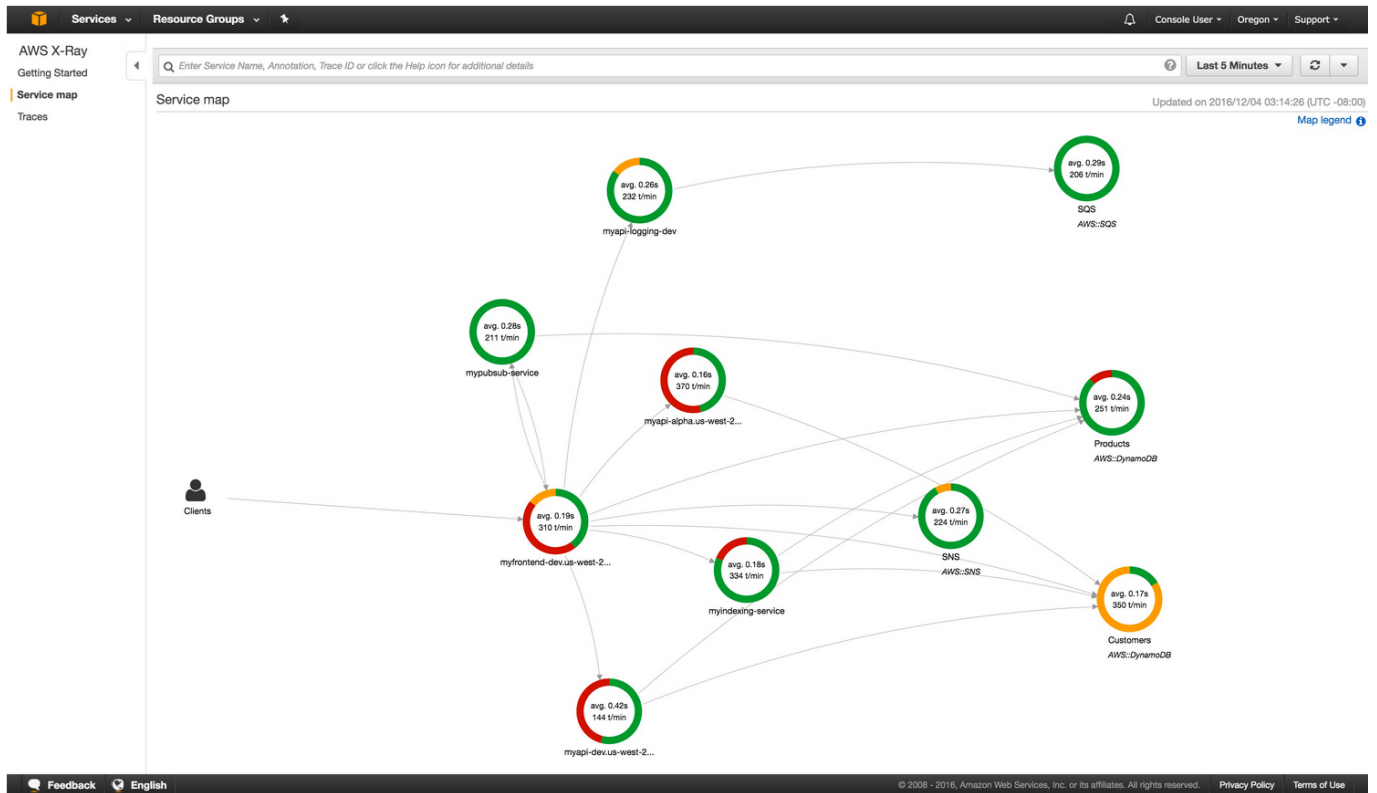
To provide a performant and cost-effective experience, X-Ray does not collect data for every request that is sent to an application. Instead, it collects data for a statistically significant number of requests. X-Ray should not be used as an audit or compliance tool because it does not guarantee data completeness.

Agent

The X-Ray agent collects data from log files and sends them to the X-Ray service for aggregation, analysis, and storage. The agent makes it easier for you to send data to the X-Ray service, instead of using the APIs directly, and is available for Amazon Linux AMI, Red Hat Enterprise Linux (RHEL), and Windows Server 2012 R2 or later operating systems.

Service Map

AWS X-Ray creates a map of services used by our application with trace data that we can use to drill into specific services or issues. This provides a view of connections between services in our application and aggregated data for each service, including average latency and failure rates. You can create dependency trees, perform cross-availability zone or region call detections, and more.



AWS services where we can use Xray-

1. EC2
2. ECS
3. Lambda
4. Elastic Beanstalk

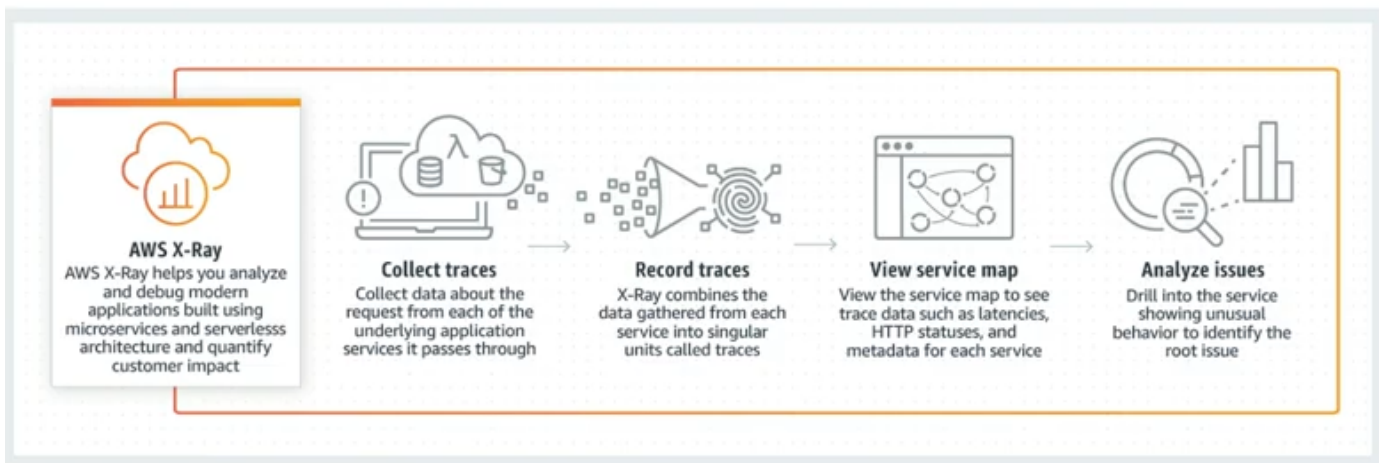
5. API Gateway

Xray supports applications running in the following languages-

1. Java
2. .Net
3. Python
4. Node.JS
5. Ruby
6. Go

How XRay service works-

How the X-Ray service works



Advantages of using AWS XRay-

X-Ray advantages

- Provides architects with a live architecture diagram
- Pinpoints the bottlenecks
- Helps optimize performance
- Helps optimize SQL queries
- Enables data-driven architecture decisions

Pricing for AWS X-Ray

Free tier:

- The first 100,000 traces recorded each month are free.
- The first 1,000,000 traces retrieved or scanned each month are free.

Additional charges:

- Beyond the free tier, traces recorded cost \$5 per 1 million traces recorded (\$0.000005 per trace).
- Beyond the free tier, traces retrieved or scanned cost \$0.50 per 1 million traces retrieved or scanned (\$0.0000005 per trace).