**Lecture Notes for AWS Developer**

**Module-5**

**AWS X-Ray**

AWS X-Ray helps developers analyze and debug production, distributed applications, such as those built using a microservices architecture. With X-Ray, you can understand how your application and its underlying services are performing to identify and troubleshoot the root cause of performance issues and errors. X-Ray provides an end-to-end view of requests as they travel through your application, and shows a map of your application’s underlying components. You can use X-Ray to analyze both applications in development and in production, from simple three-tier applications to complex microservices applications consisting of thousands of services.

**Benefits**

**Review request behavior**

AWS X-Ray traces user requests as they travel through your entire application. It aggregates the data generated by the individual services and resources that make up your application, providing you an end-to-end view of how your application is performing.

**Discover application issues**

With AWS X-Ray, you can glean insights into how your application is performing and discover root causes. With X-Ray’s tracing features, you can follow request paths to pinpoint where in your application and what is causing performance issues. X-Ray provides annotations so you can append metadata to traces, making it possible to tag and filter trace data so you can discover patterns and diagnose issues.

**Improve application performance**

AWS X-Ray helps you identify performance bottlenecks. X-Ray’s service maps let you see relationships between services and resources in your application in real time. You can easily detect where high latencies are occurring, visualize node and edge latency distribution for services, and then drill down into the specific services and paths impacting application performance.

**Ready to use with AWS**

AWS X-Ray works with Amazon EC2, Amazon EC2 Container Service (Amazon ECS)\, AWS Lambda, and AWS Elastic Beanstalk. You can use X-Ray with applications written in Java, Node.js, and .NET that are deployed on these services.

**Designed for a variety of applications**

AWS X-Ray works for both simple and complex applications, either in development or in production. You can analyze simple asynchronous event calls, three-tier web applications, or complex microservices applications consisting of thousands of services. With X-Ray, you can trace requests made to applications that span multiple AWS accounts, AWS Regions, and Availability Zones.

**How it works**



**Amazon CloudWatch**

Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers. You can use CloudWatch to detect anomalous behavior in your environments, set alarms, visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications
running smoothly.

**AWS CloudTrail**

AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account. With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure. CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services. This event history simplifies security analysis, resource change tracking, and troubleshooting. In addition, you can use CloudTrail to detect unusual activity in your AWS accounts. These capabilities help simplify operational analysis and troubleshooting.

**Difference between CloudWatch and CloudTrail**

1. CloudWatch is a monitoring service for AWS resources and applications. CloudTrail is a web service that records API activity in your AWS account. They are both useful monitoring tools in AWS.
2. CloudWatch offers free basic monitoring for your resources by default, such as EC2 instances, EBS volumes, and RDS DB instances. CloudTrail is also enabled by default when you create your AWS account.
3. With CloudWatch, you can collect and track metrics, collect and monitor log files, and set alarms. On the other hand, CloudTrail logs information on who made a request, the services used, the actions performed, parameters for the actions, and the response elements returned by the AWS service. CloudTrail Logs are then stored in an S3 bucket or a CloudWatch Logs log group that you specify.
4. Typically, CloudTrail delivers an event within 15 minutes of the API call. CloudWatch delivers metric data in 5 minutes periods for basic monitoring and 1 minute periods for detailed monitoring. The CloudWatch Logs Agent will send log data every five seconds by default.
5. You can enable detailed monitoring from your AWS resources to send metric data to CloudWatch more frequently, with an additional cost. CloudTrail helps you ensure compliance and regulatory standards.
6. CloudWatch Logs reports on application logs, while CloudTrail Logs provide you specific information on what occurred in your AWS account.

**Amazon Simple Notification Service (Amazon SNS)**

Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both application-to-application (A2A) and application-to-person (A2P) communication.

The A2A pub/sub functionality provides topics for high-throughput, push-based, many-to-many messaging between distributed systems, microservices, and event-driven serverless applications. Using Amazon SNS topics, your publisher systems can fanout messages to a large number of subscriber systems including Amazon SQS queues, AWS Lambda functions and HTTPS endpoints, for parallel processing, and Amazon Kinesis Data Firehose. The A2P functionality enables you to send messages to users at scale via SMS, mobile push, and email.

**Benefits**

**Modernize and decouple your applications**

Amazon SNS enables you to modernize your applications and decouple them into smaller, independent components that are easier to develop, deploy and maintain. Leveraging a pub/sub event-driven architecture for your application improves performance, reliability, and allows each component to scale independently.

**Send messages directly to millions of users**

Amazon SNS enables you to send messages or notifications directly to users with SMS text messages to over 200 countries, mobile push on Apple, Android, and other platforms or email (SMTP). Amazon SNS provides redundancy across multiple SMS providers and allows you to push mobile notifications using a single endpoint for all platforms.

**Reliably deliver messages**

Amazon SNS uses a number of strategies that work together to provide message durability. To start, published messages are stored across multiple, geographically-separated servers and data centers. If a subscribed endpoint isn't available, Amazon SNS executes a message delivery retry policy.

**Automatically scale your workload**

Amazon SNS leverages the proven AWS cloud to dynamically scale with your application. Amazon SNS is a fully managed service, taking care of the heavy lifting related to capacity planning, provisioning, monitoring, and patching. The service is designed to handle high-throughput, bursty traffic patterns and enables you to send millions of messages per second.

**Ensure accuracy with message ordering and deduplication**

Amazon SNS FIFO topics work with Amazon SQS FIFO queues to ensure messages are delivered in a strictly ordered manner and are only processed once (deduplicated). This enables you to maintain consistency when processing transactions across a single or multiple independent services where it’s critical that messages are in the correct order. It also allows you to offload the effort of writing custom code for ordering and message deduplication.

**Simplify your architecture with Message Filtering**

Amazon SNS helps you simplify your pub/sub messaging architecture by offloading the message filtering logic from your subscriber systems, and message routing logic from your publisher systems. With Amazon SNS message filtering, subscribing endpoints receive only the messages of interest, instead of all messages published to the topic. Amazon CloudWatch gives visibility into your filtering activity, and AWS CloudFormation allows you to deploy subscription filter policies in an automated and secure manner.

**What is Pub/Sub Messaging?**

Publish/subscribe messaging, or pub/sub messaging, is a form of asynchronous service-to-service communication used in serverless and microservices architectures. In a pub/sub model, any message published to a topic is immediately received by all of the subscribers to the topic. Pub/sub messaging can be used to enable event-driven architectures, or to decouple applications in order to increase performance, reliability and scalability.

**How it works**



**Amazon Simple Queue Service (SQS)**

Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. SQS eliminates the complexity and overhead associated with managing and operating message oriented middleware, and empowers developers to focus on differentiating work. Using SQS, you can send, store, and receive messages between software components at any volume, without losing messages or requiring other services to be available. Get started with SQS in minutes using the AWS console, Command Line Interface or SDK of your choice, and three simple commands.

SQS offers two types of message queues. Standard queues offer maximum throughput, best-effort ordering, and at-least-once delivery. SQS FIFO queues are designed to guarantee that messages are processed exactly once, in the exact order that they are sent.

**Benefits**

**Eliminate administrative overhead**

AWS manages all ongoing operations and underlying infrastructure needed to provide a highly available and scalable message queuing service. With SQS, there is no upfront cost, no need to acquire, install, and configure messaging software, and no time-consuming build-out and maintenance of supporting infrastructure. SQS queues are dynamically created and scale automatically so you can build and grow applications quickly and efficiently.

**Reliably deliver messages**

Use Amazon SQS to transmit any volume of data, at any level of throughput, without losing messages or requiring other services to be available. SQS lets you decouple application components so that they run and fail independently, increasing the overall fault tolerance of the system. Multiple copies of every message are stored redundantly across multiple availability zones so that they are available whenever needed.

**Keep sensitive data secure**

You can use Amazon SQS to exchange sensitive data between applications using server-side encryption (SSE) to encrypt each message body. Amazon SQS SSE integration with AWS Key Management Service (KMS) allows you to centrally manage the keys that protect SQS messages along with keys that protect your other AWS resources. AWS KMS logs every use of your encryption keys to AWS CloudTrail to help meet your regulatory and compliance needs.

**Scale elastically and cost-effectively**

Amazon SQS leverages the AWS cloud to dynamically scale based on demand. SQS scales elastically with your application so you don’t have to worry about capacity planning and pre-provisioning. There is no limit to the number of messages per queue, and standard queues provide nearly unlimited throughput. Costs are based on usage which provides significant cost saving versus the “always-on” model of self-managed messaging middleware.

**Difference between AWS SNS and SQS**

1. Entity Type
SQS : Queue (Similar to JMS)
SNS : Topic (Pub/Sub system)
2. Message consumption
SQS : Pull Mechanism — Consumers poll and pull messages from SQS
SNS : Push Mechanism — SNS Pushes messages to consumers
3. Use Case
SQS : Decoupling two applications and allowing parallel asynchronous processing
SNS : Fanout — Meaning allowing same message to be processed in multiple ways
4. Persistence
SQS : Messages are persisted for some (configurable) duration is no consumer available
SNS : No persistence. Whichever consumer is present at the time of message arrival, get the message and the message is deleted. If no consumers available then the message is lost.
5. In SQS the message delivery is guaranteed but in SNS it is not.
6. Consumer Type
SQS : All the consumers are supposed to be identical and hence process the messages in exact same way
SNS : All the consumers are (supposed to be) processing the messages in different ways
7. Sample applications
SQS : Jobs framework. Where the Jobs are submitted to SQS and the consumers at the other end can process the jobs asynchronously. And if the job frequency increases then the number of consumers can be increased for parallel processing
SNS : Image processing. If someone uploads an image to S3 then watermark that image, create a thumbnail and also send a ThankYou email. In that case S3 can send notification to SNS Topic and 3 consumers can be attached to SNS topic.

**Amazon Kinesis**

Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information. Amazon Kinesis offers key capabilities to cost-effectively process streaming data at any scale, along with the flexibility to choose the tools that best suit the requirements of your application. With Amazon Kinesis, you can ingest real-time data such as video, audio, application logs, website clickstreams, and IoT telemetry data for machine learning, analytics, and other applications. Amazon Kinesis enables you to process and analyze data as it arrives and respond instantly instead of having to wait until all your data is collected before the processing can begin.

**Benefits**

**Real-time**

Amazon Kinesis enables you to ingest, buffer, and process streaming data in real-time, so you can derive insights in seconds or minutes instead of hours or days.

**Fully managed**

Amazon Kinesis is fully managed and runs your streaming applications without requiring you to manage any infrastructure.

**Scalable**

Amazon Kinesis can handle any amount of streaming data and process data from hundreds of thousands of sources with very low latencies.