**AWS Elastic Beanstalk**

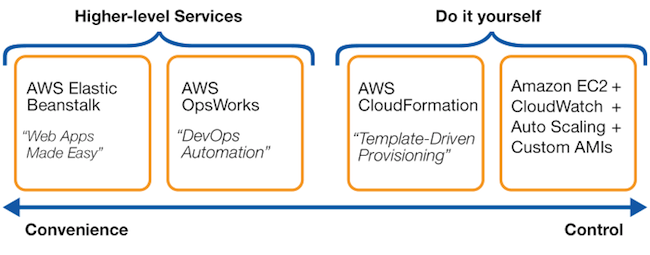
AWS Elastic Beanstalk, you provide code and configuration settings, and Elastic Beanstalk deploys the resources necessary to perform the following tasks: Adjust capacity, Load balancing, Automatic scaling, Application health monitoring

**AWS CloudFormation**

It is an infrastructure as a code (IAAC) tool. This means that you can build an environment by writing lines of code in JSON or XML files instead of using the AWS Management Console to individually provision resources.

AWS CloudFormation provisions your resources in a safe, repeatable manner, enabling you to frequently build your infrastructure and applications without having to perform manual actions. It determines the right operations to perform when managing your stack and rolls back changes automatically if it detects errors.

***Beanstalk is PaaS (platform as a service) while CloudFormation is IaaC (infrastructure as code).***



When designing applications on AWS, you can take a microservices approach with services and components that fulfill different functions.

Two services facilitate application integration:

1. **Amazon Simple Notification Service (Amazon SNS)**is a **publish/subscribe service**. Using Amazon SNS topics, a publisher publishes messages to subscribers.

In Amazon SNS, subscribers can be web servers, email addresses, AWS Lambda functions, or several other options.

1. **Amazon Simple Queue Service (Amazon SQS)** is a message **queuing service**.

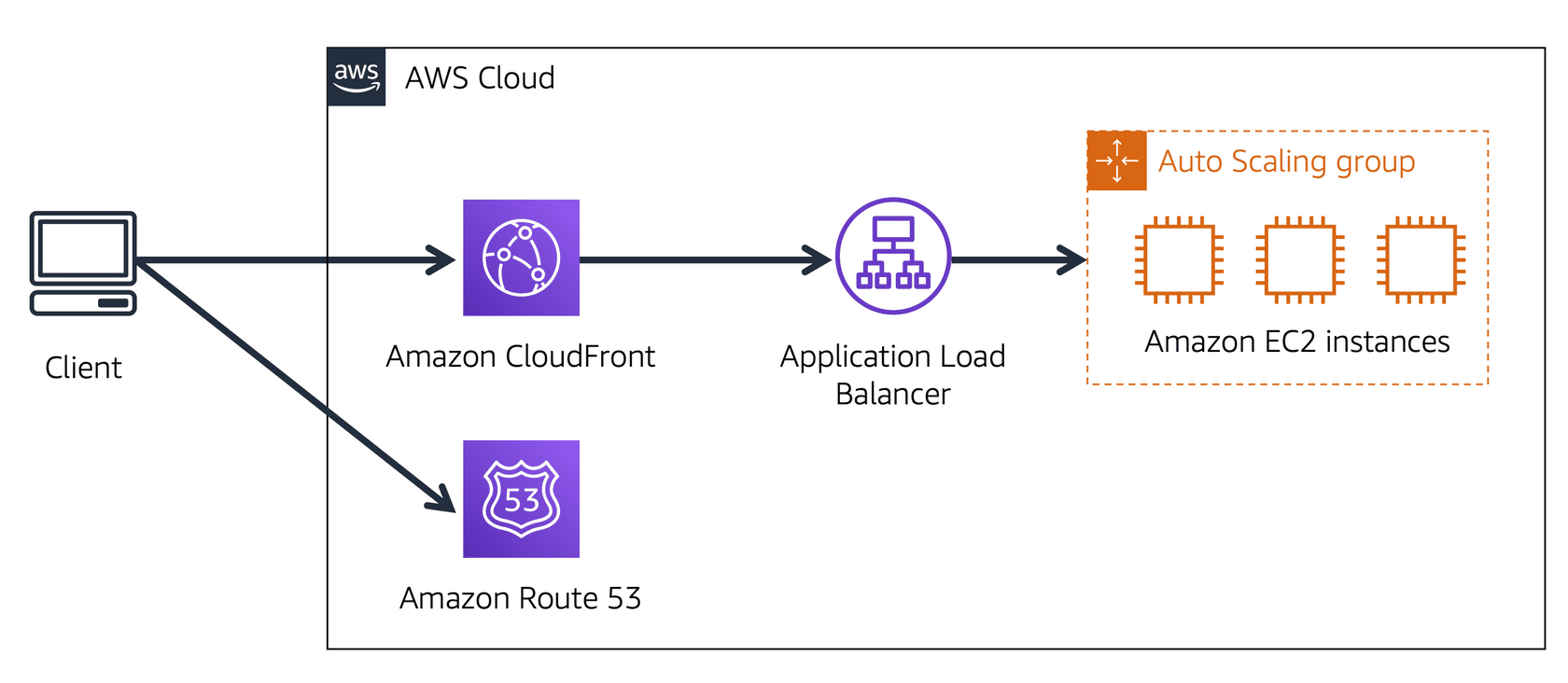
Using Amazon SQS, you can send, store, and receive messages between software components, without losing messages or requiring other services to be available. In Amazon SQS, an application sends messages into a queue. A user or service retrieves a message from the queue, processes it, and then deletes it from the queue.

[**Amazon Route 53**](https://aws.amazon.com/route53) is a DNS web service. It gives developers and businesses a reliable way to route end users to internet applications hosted in AWS.

Amazon Route 53 connects user requests to infrastructure running in AWS (such as Amazon EC2 instances and load balancers). It can route users to infrastructure outside of AWS.

Another feature of Route 53 is the ability to manage the DNS records for domain names. You can register new domain names directly in Route 53. You can also transfer DNS records for existing domain names managed by other domain registrars. This enables you to manage all of your domain names within a single location.

**Example: How Amazon Route 53 and Amazon CloudFront deliver content**



Suppose that AnyCompany’s application is running on several Amazon EC2 instances. These instances are in an Auto Scaling group that attaches to an Application Load Balancer.

1 A customer requests data from the application by going to AnyCompany’s website.

2 Amazon Route 53 uses DNS resolution to identify AnyCompany.com’s corresponding IP address, 192.0.2.0. This information is sent back to the customer.

3 The customer’s request is sent to the nearest edge location through Amazon CloudFront.

4 Amazon CloudFront connects to the Application Load Balancer, which sends the incoming packet to an Amazon EC2 instance.

[**AWS Database Migration Service (AWS DMS)**](https://aws.amazon.com/dms/) enables you to migrate relational databases, nonrelational databases, and other types of data stores.

With AWS DMS, you move data between a source database and a target database.

[The source and target databases](https://aws.amazon.com/dms/resources) can be of the same type **(homogeneous migration)** or different types **(heterogeneous migration).** During the migration, your source database remains operational, reducing downtime for any applications that rely on the database.

**Use cases for AWS DMS**  
***Development and test database migrations:*** Enabling developers to test applications against production data without affecting production users

***Database consolidation:***Combining several databases into a single database

***Continuous replication:***Sending ongoing copies of your data to other target sources instead of doing a one-time migration

**AWS Trusted Advisor** is a web service that inspects your AWS environment and provides **real-time recommendations** in accordance with AWS best practices in five categories: ***cost optimization, performance, security, fault tolerance, and service limits.***

[**Amazon CloudWatch**](https://aws.amazon.com/cloudwatch/) is a web service that enables you to monitor and manage various metrics and configure alarm actions based on data from those metrics.

With CloudWatch, you can create [**alarms**](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html) that automatically perform actions if the value of your metric has gone above or below a predefined threshold.

The CloudWatch [**dashboard**](https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/CloudWatch_Dashboards.html) feature enables you to access all the metrics for your resources from a single location.

[**AWS CloudTrail**](https://aws.amazon.com/cloudtrail/) records API calls for your account. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, and more. You can think of CloudTrail as a “trail” of breadcrumbs (or a log of actions) that someone has left behind them.

Recall that you can use API calls to provision, manage, and configure your AWS resources. With CloudTrail, you can view a complete history of user activity and API calls for your applications and resources.

Events are typically updated in CloudTrail within 15 minutes after an API call. You can filter events by specifying the time and date that an API call occurred, the user who requested the action, the type of resource that was involved in the API call, and more.

Within CloudTrail, you can also enable [**CloudTrail Insights**](https://docs.aws.amazon.com/awscloudtrail/latest/userguide/logging-insights-events-with-cloudtrail.html). This optional feature allows CloudTrail to automatically detect unusual API activities in your AWS account.

Graphical user interface, application

Description automatically generated

**AWS Key Management Service (AWS KMS)**

Applications’ data is secure while in storage **(encryption at rest)** and while it is transmitted, known as **encryption in transit**.

[**AWS Key Management Service (AWS KMS)**](https://aws.amazon.com/kms) enables you to perform encryption operations through the use of **cryptographic keys**. A cryptographic key is a random string of digits used for locking (encrypting) and unlocking (decrypting) data.

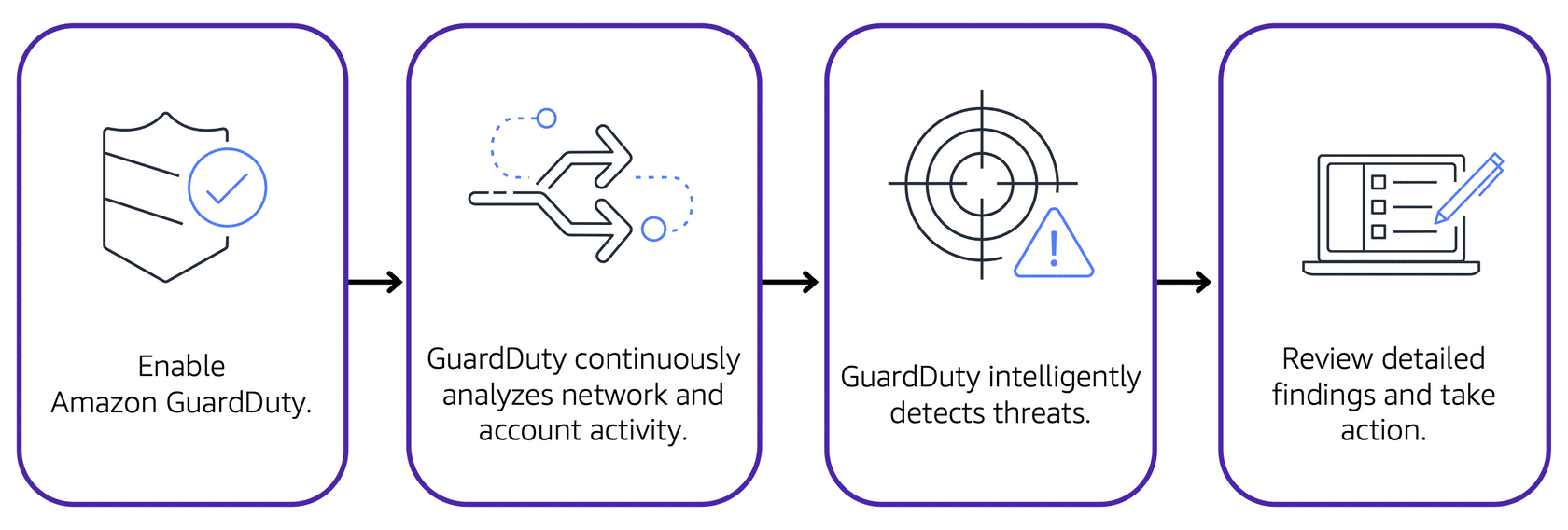
[**AWS WAF**](https://aws.amazon.com/waf) is a web application firewall that lets you monitor network requests that come into your web applications.

AWS WAF works together with Amazon CloudFront and an Application Load Balancer. Recall the network access control lists that you learned about in an earlier module. AWS WAF works in a similar way to block or allow traffic. However, it does this by using a [**web access control list (ACL)**](https://docs.aws.amazon.com/waf/latest/developerguide/web-acl.html) to protect your AWS resources.

**Amazon Inspector**

Amazon Inspector helps to improve the security and compliance of applications by running **automated security assessments**. It checks applications for security vulnerabilities and deviations from security best practices, such as open access to Amazon EC2 instances and installations of vulnerable software versions.

[**Amazon GuardDuty**](https://aws.amazon.com/guardduty) is a service that provides intelligent threat detection for your AWS infrastructure and resources. It identifies threats by continuously monitoring the network activity and account behavior within your AWS environment.



After you have enabled GuardDuty for your AWS account, GuardDuty begins monitoring your network and account activity. You do not have to deploy or manage any additional security software. GuardDuty then continuously analyzes data from multiple AWS sources, including VPC Flow Logs and DNS logs.

[**AWS Cost Explorer**](https://aws.amazon.com/aws-cost-management/aws-cost-explorer/) is a tool that enables you to visualize, understand, and manage your AWS costs and usage over time.

AWS Cost Explorer includes a default report of the costs and usage for your top five cost-accruing AWS services. You can apply custom filters and groups to analyze your data.

Create an estimate for the cost of your use cases on AWS - You can perform this action in AWS Pricing Calculator.

In [**AWS Budgets**](https://aws.amazon.com/aws-cost-management/aws-budgets), you can create budgets to plan your service usage, service costs, and instance reservations.

The information in AWS Budgets updates three times a day. This helps you to accurately determine how close your usage is to your budgeted amounts or to the AWS Free Tier limits.

In AWS Budgets, you can also set custom alerts when your usage exceeds (or is forecasted to exceed) the budgeted amount.