

CS2032 - Cloud Computing (Ciclo 2024-2)

Otros tópicos

Semana 14 - Clase 1: AWS Well-Architected

ELABORADO POR: GERALDO COLCHADO

Contenido

AWS Well-Architected

1. Objetivo de la sesión
2. AWS Well-Architected
3. Pilares
4. Enfoques
5. Enfoque: Aplicaciones Serverless
6. Cierre

Objetivo de la sesión

AWS Well-Architected

- Comprender qué es el AWS Well-Architected
- Identificar los 6 pilares del AWS Well-Architected
- Conocer algunos escenarios de Enfoque de Aplicaciones sin servidor

Contenido

AWS Well-Architected

1. Objetivo de la sesión
2. **AWS Well-Architected**
3. Pilares
4. Enfoques
5. Enfoque: Aplicaciones Serverless
6. Cierre

AWS Well-Architected Alcance

AWS Well-Architected

Aprenda, calcule y cree con prácticas recomendadas para arquitecturas

AWS Well-Architected **ayuda** a los arquitectos de la nube a **crear una infraestructura segura**, de **alto rendimiento**, **resistente** y **eficiente** para una **variedad de aplicaciones y cargas de trabajo**. Este marco, **creado en torno a seis pilares** (excelencia operativa, seguridad, fiabilidad, eficiencia de rendimiento, optimización de costos y sostenibilidad), ofrece un enfoque coherente para que los clientes y los socios evalúen las arquitecturas e implementen **diseños escalables**.

AWS Well-Architected Framework **incluye enfoques de dominios específicos**, laboratorios prácticos y AWS Well-Architected Tool. **AWS Well-Architected Tool**, disponible sin costo alguno en la **consola de administración de AWS**, proporciona un mecanismo para evaluar regularmente las cargas de trabajo, identificar los problemas de alto riesgo y registrar las mejoras.

Contenido

AWS Well-Architected

1. Objetivo de la sesión
2. AWS Well-Architected
3. **Pilares**
4. Enfoques
5. Enfoque: Aplicaciones Serverless
6. Cierre

AWS Well-Architected Pilares



AWS Well-Architected Pilares

Pilar de excelencia operativa

El pilar de la excelencia operativa se concentra en **ejecutar y monitorear los sistemas** y en **mejorar constantemente los procesos** y los procedimientos. Entre los temas clave se incluyen la **automatización de cambios**, **la respuesta a eventos** y la **definición de estándares para administrar las operaciones diarias**.

Pilar de seguridad

El pilar de la seguridad se concentra en **proteger la información y los sistemas**. Entre los temas clave se incluyen la **confidencialidad** y la **integridad de los datos**, la administración de los **permisos de usuarios** y el establecimiento de **controles para detectar eventos de seguridad**.

AWS Well-Architected Pilares

Pilar de eficacia del rendimiento

El pilar de eficacia del rendimiento se centra en la asignación estructurada y simplificada de TI y en los recursos informáticos. Entre los temas clave se incluyen la **selección de los tipos y tamaños de recursos optimizados para los requisitos de la carga de trabajo**, la **supervisión del rendimiento** y el **mantenimiento de la eficacia a medida que evolucionan las necesidades de la empresa**.

Pilar de optimización de costos

El pilar de optimización de costos se centra en **evitar gastos innecesarios**. Entre los temas clave se incluyen la comprensión del tiempo dedicado y el control de la asignación de fondos, la **selección de recursos para el tipo y la cantidad adecuados** y el **escalado** para cumplir con las necesidades de la empresa **sin gastos excesivos**.

AWS Well-Architected Pilares

Pilar de fiabilidad

El pilar de fiabilidad se centra en las cargas de trabajo que realizan las funciones previstas y en **cómo recuperarse rápidamente de los errores para cumplir con las demandas**. Entre los temas clave se incluyen el **diseño de sistemas distribuidos**, la **planificación de la recuperación** y cómo adaptarse a los requisitos cambiantes.

Pilar de sostenibilidad

El pilar de sostenibilidad se centra en **minimizar los impactos ambientales de ejecutar cargas de trabajo en la nube**. Entre los temas clave se incluyen un modelo de responsabilidad compartida para la sostenibilidad, la comprensión del impacto y la maximización del uso para minimizar los recursos necesarios y reducir los impactos posteriores.

Contenido

AWS Well-Architected

1. Objetivo de la sesión
2. AWS Well-Architected
3. Pilares
4. **Enfoques**
5. Enfoque: Aplicaciones Serverless
6. Cierre

AWS Well-Architected

Enfoques

Enfoques de AWS Well-Architected

Los enfoques de AWS Well-Architected **amplían** la orientación que brinda AWS Well-Architected para **abordar dominios tecnológicos e industriales específicos**, como **machine learning (ML)**, **análisis de datos**, **sin servidor**, computación de alto rendimiento (HPC), **IoT**, **SAP**, **streaming de contenido multimedia**, el sector de los **videojuegos**, redes híbridas y servicios de finanzas. **Para evaluar por completo sus cargas de trabajo, utilice enfoques correspondientes junto con AWS Well-Architected Framework y los seis pilares.**

Contenido

AWS Well-Architected

1. Objetivo de la sesión
2. AWS Well-Architected
3. Pilares
4. Enfoques
5. **Enfoque: Aplicaciones Serverless**
6. Cierre

AWS Well-Architected

Enfoque: Aplicaciones Serverless

Serverless Applications Lens - AWS Well-Architected Framework

[PDF](#) | [RSS](#)

Publication date: **July 14, 2022** ([Document revisions](#))

This document describes the Serverless Applications Lens for the [AWS Well-Architected Framework](#) [🔗](#). The document covers common serverless applications scenarios and identifies key elements to ensure that your workloads are architected according to best practices.

AWS Well-Architected

Enfoque: Aplicaciones Serverless - Escenarios

Reference architecture



RESTful microservices

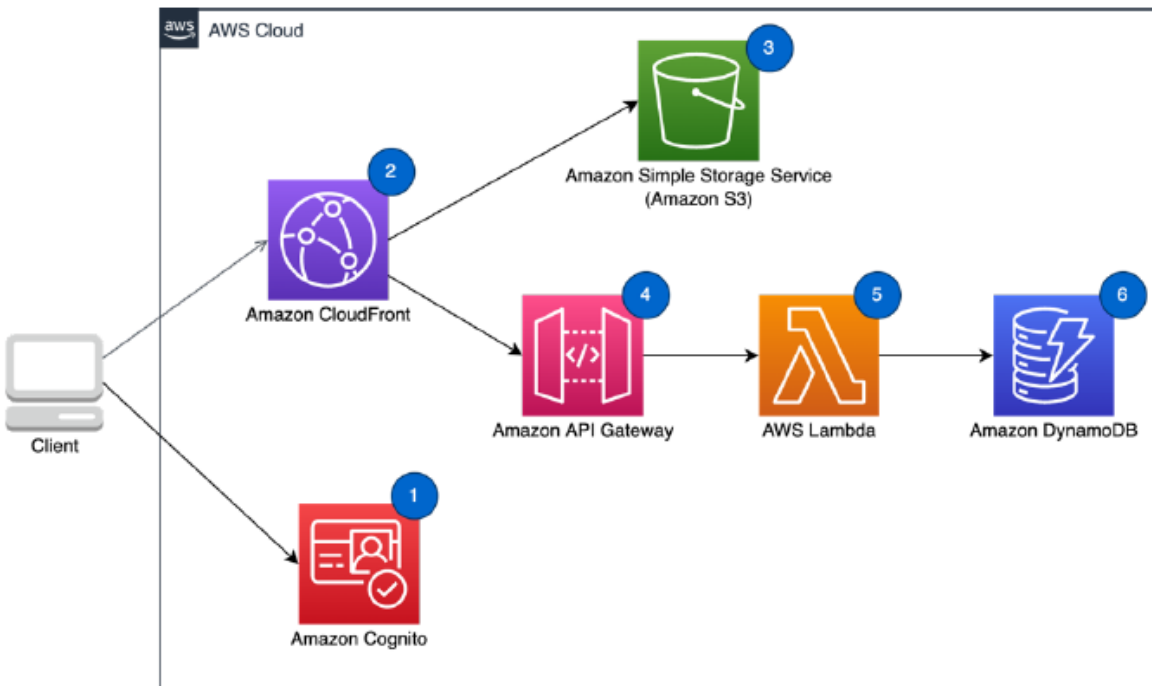
1. **Customers** leverage your microservices by making HTTP API calls. Ideally, your consumers should have a tightly bound service contract to your API to achieve consistent expectations of service levels and change control.
2. **Amazon API Gateway** hosts RESTful HTTP requests and responses to customers. In this scenario, API Gateway provides **built-in authorization**, throttling, security, fault tolerance, request and response mapping, and performance optimizations.
3. **AWS Lambda** contains the **business logic** to process incoming API calls and use DynamoDB as a persistent storage.
4. **Amazon DynamoDB** **persistently stores microservices data and scales based on demand**. Since microservices are often designed to do one thing well, a schemaless NoSQL data store is regularly incorporated.

AWS Well-Architected

Enfoque: Aplicaciones Serverless - Escenarios

Web application

Reference architecture



1. **Amazon Cognito user pools** provides user management and identity provider features for your web application. Tokens issued by Amazon Cognito are used to authenticate users when making request to Amazon API Gateway.
2. **Amazon CloudFront** provides a better user experience by accelerating content delivery of static assets and calls to your backend compute layer. CloudFront brings content closer to clients using AWS's global Points of Presence (PoPs). CloudFront can also cache API calls to reduce calls to compute backends while also providing optimal network routing for non-cacheable API calls.
3. **Amazon S3** hosts web application static assets such as HTML, CSS, JavaScript and images. Content is securely served through CloudFront.
4. **Amazon API Gateway** serves as the secure HTTPS endpoint. Web applications make REST API calls to a public HTTPS endpoint using either a custom domain name or a unique API Gateway-provided domain.
5. An **AWS Lambda** function provides create, read, update, and delete (CRUD) operations on top of DynamoDB for your web application.
6. **Amazon DynamoDB** can provide a NoSQL data store which elastically scales with your web application.

AWS Well-Architected

Enfoque: Aplicaciones Serverless - Escenarios

Event-driven architectures

Event-driven architectures are becoming a popular and preferable way of building large distributed microservice-based applications. This approach helps you build scalable, resilient, agile and cost-effective solutions.

Using AWS serverless services to implement event-driven approach will allow you to build scalable, fault tolerant applications. You can use messaging services like Amazon SQS for reliable and durable communication between microservices. For fan out of the events you can use Amazon SNS topics. If you need event filtering and routing you can utilize Amazon EventBridge.

Typical use cases for event-driven architectures:

- Communication between microservices
- Integration with third-party SaaS applications
- Cross-account and cross region data replication
- Parallel event processing, fanout

Contenido

AWS Well-Architected

1. Objetivo de la sesión
2. AWS Well-Architected
3. Pilares
4. Enfoques
5. Enfoque: Aplicaciones Serverless
6. **Cierre**

Cierre

Explique con sus propias palabras

- Qué es el AWS Well-Architected?
- Uno de los 6 pilares del AWS Well-Architected
- Un escenario de Enfoque de Aplicaciones sin servidor

Gracias

Docente: Geraldo Colchado