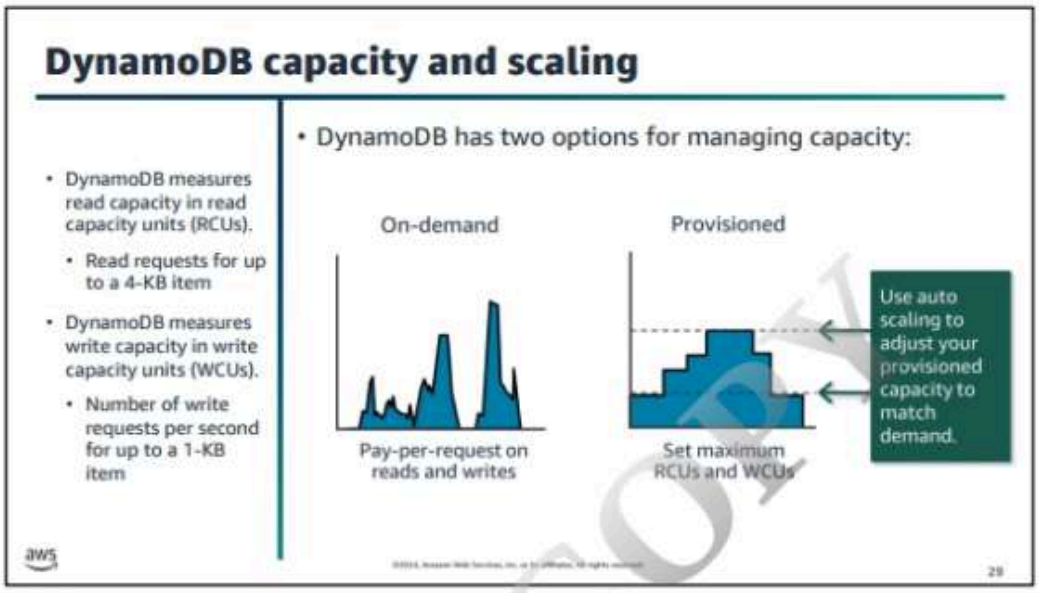
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



When planning for capacity in DynamoDB, you must consider the expected number of read/write requests per second and the size of those requests. This information helps you choose the capacity mode for your table. It also helps you design your table to manage message size.

On-demand capacity mode is a pay-per-request model. On-demand capacity mode is best in the following scenarios:

- You have unknown workloads.
- You have unpredictable traffic.
- You prefer to pay for only what you use.

With provisioned capacity mode, you set a maximum number of RCUs and WCUs. When traffic exceeds those limits, DynamoDB throttles those requests to control your costs. You can adjust your provisioned capacity by using auto scaling. Provisioned capacity mode is best in the following scenarios:

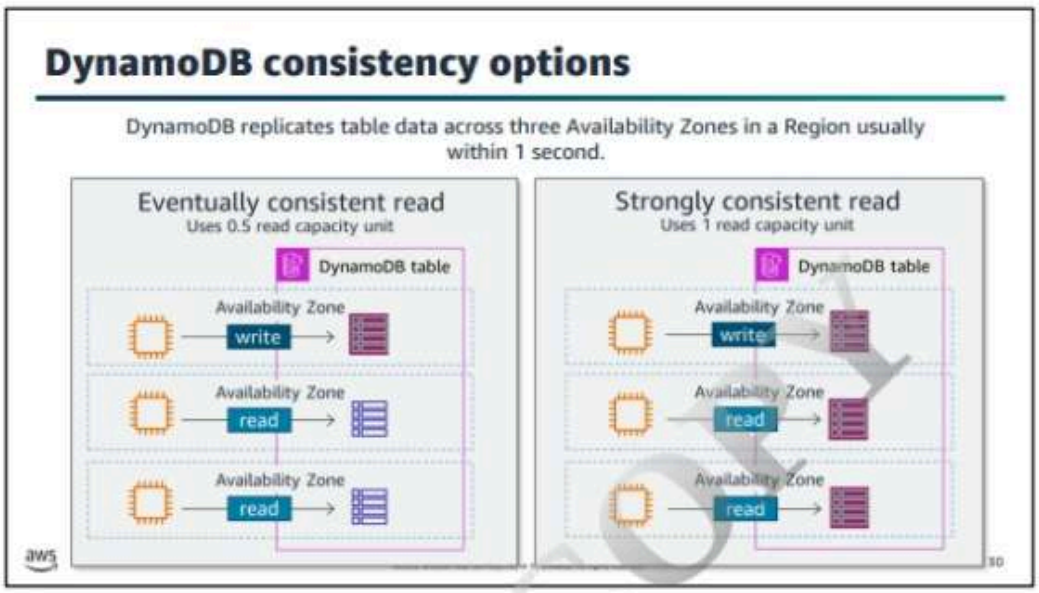
- You have predictable application traffic.
- You have traffic that is consistent or changes gradually.
- You can forecast capacity requirements to control costs.

For more information about DynamoDB auto scaling, see "Amazon DynamoDB Auto Scaling: Performance and Cost Optimization at Any Scale" in the *AWS Database Blog* at https://aws.amazon.com/blogs/database/amazon-dynamodb-auto-scaling-performance-and-cost-optimization-at-any-scale/.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



When your application writes data to a DynamoDB table and receives an HTTP 200 response (OK), the write has occurred and is durable. The data is eventually consistent across all storage locations, usually within 1 second or less. DynamoDB supports eventually consistent and strongly consistent reads:

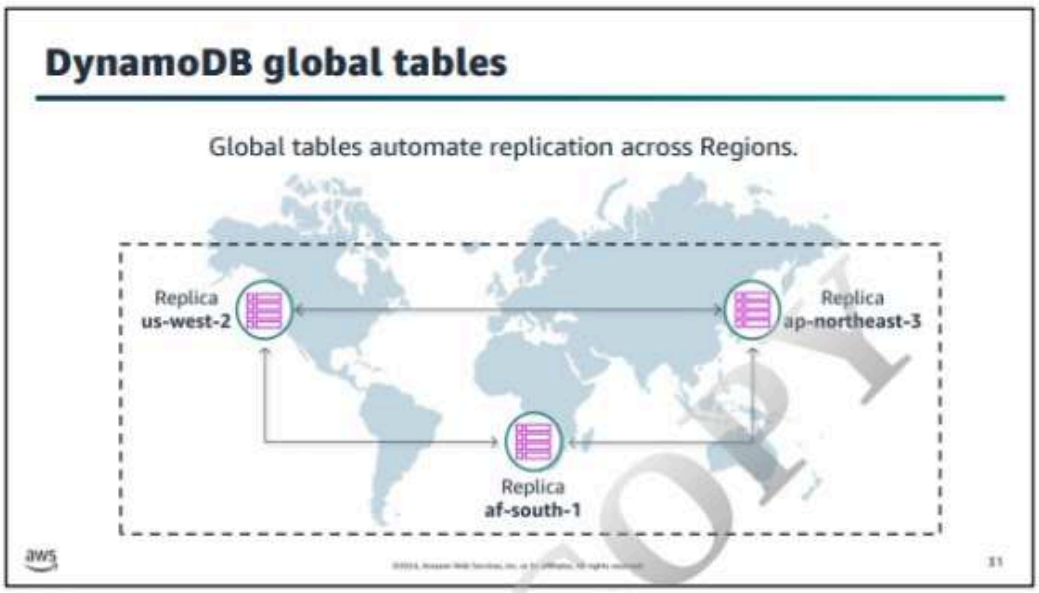
- **Eventually consistent reads:** When you read data from a DynamoDB table, the response might not reflect the results of a recently completed write operation. The response might include some stale data. If you repeat your read request after a short time, the response should return the latest data.
- **Strongly consistent reads:** When you request a strongly consistent read, DynamoDB returns a response with the most up-to-date data, which reflects updates from all prior successful write operations. A strongly consistent read might not be available if a network delay or outage occurs.

DynamoDB uses eventually consistent reads unless you specify otherwise. Read operations (such as GetItem, Query, and Scan) provide a ConsistentRead parameter. If you set this parameter to true, DynamoDB uses strongly consistent reads during the operation.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



A *global table* is a collection of one or more DynamoDB tables, which are all owned by a single AWS account, identified as replica tables. A *replica table* (or *replica* for short) is a single DynamoDB table that functions as part of a global table. Each replica stores the same set of data items. Any given global table can have only one replica table per Region, and every replica has the same table name and the same primary key schema.

DynamoDB global tables provide a fully managed solution for deploying a multi-Region, multi-active database without having to build and maintain your own replication solution. When you create a global table, you specify the Regions where you want the table to be available. DynamoDB performs all the necessary tasks to create identical tables in these Regions and propagate ongoing data changes to all of them. DynamoDB communicates these changes over the AWS network backbone.

For more information about Amazon DynamoDB global tables, see "Amazon DynamoDB Global Tables" at https://aws.amazon.com/dynamodb/global-tables/.

DO NOT COPY
javierartiga.espublico@gmail.com

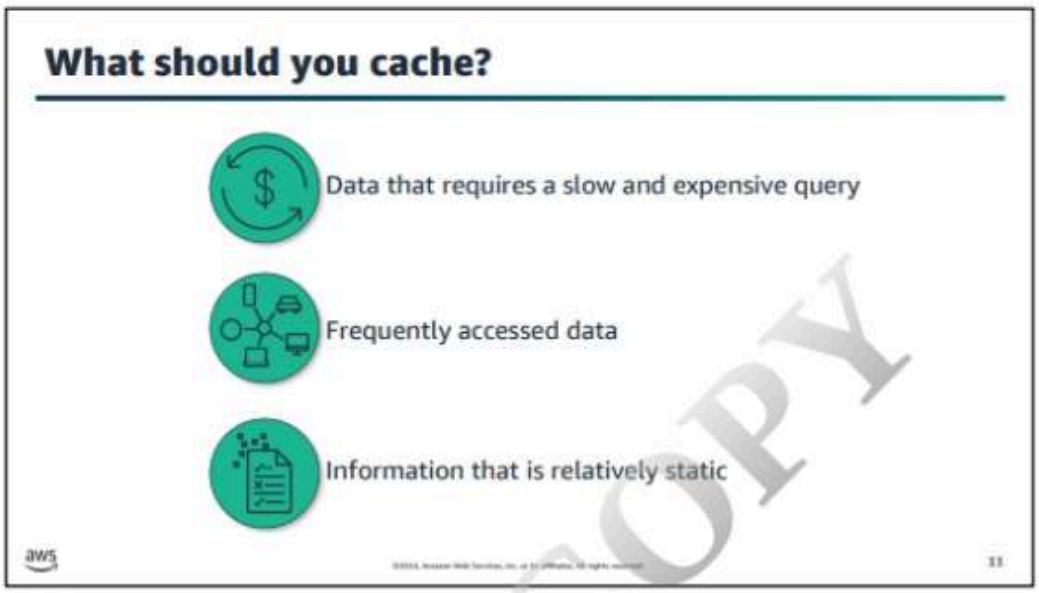
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

The database services manager asks, "How can we cache databases in the cloud to maximize performance?" The database team has identified some common queries that are causing a lot of read traffic. The company is looking for your advice on how to cache this commonly accessed data to improve performance and decrease the load on the databases.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



To determine whether your application should use database caching, consider the following points:

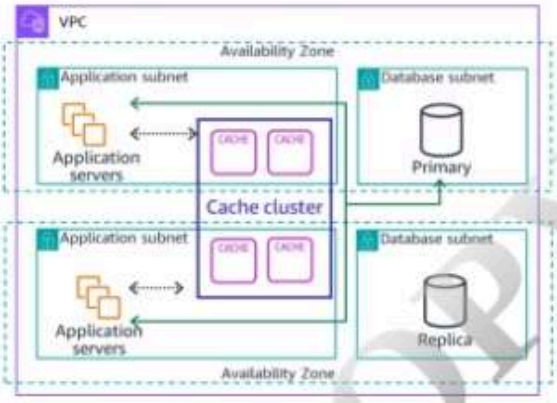
- **Speed and expense:** Some database queries are inherently slower and more expensive than others. For example, queries that perform joins on multiple tables are significantly slower and more expensive than single-table queries. If requesting data requires a slow and expensive query, it's a candidate for caching.
- **Data and access pattern:** Determining what to cache also involves understanding the data itself and its access patterns. For example, it doesn't make sense to cache data that is rapidly changing or is seldom accessed. For caching to provide meaningful benefit, the data should be relatively static and frequently accessed, such as a personal profile on a social media site.
- **Cache validity:** Data can become out-of-date while it is stored in cache. Writes that occur on that data in the database might not be reflected in that cached data. To determine whether your data is a candidate for caching, you must determine your application's tolerance for occasionally inaccurate cached data.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Caching architecture



Without caching, EC2 instances read and write directly to the database. With caching, an instance first attempts to read from a cache, which uses high performance memory. For example, database caches on AWS such as Amazon ElastiCache and Amazon DynamoDB Accelerator (DAX) are in-memory databases. They use a cache cluster that contains a set of cache nodes distributed between subnets. Resources within those subnets have high-speed access to those nodes.

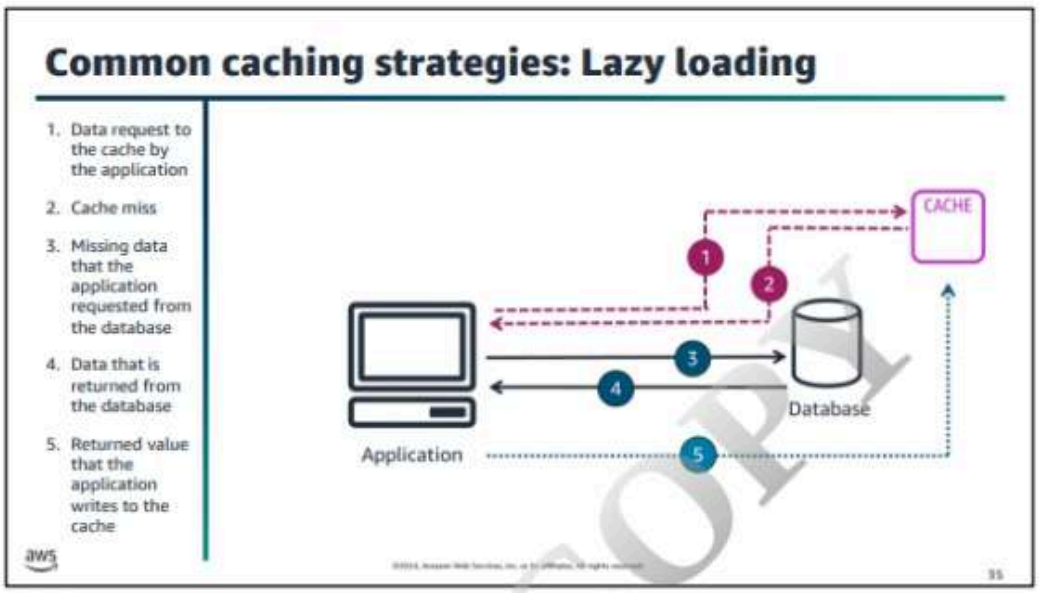
In this example, a VPC has an application subnet and a data subnet in each of two Availability Zones. The application servers in both application subnets connect to a primary database in one data subnet. The other data subnet holds a replica database. A cache cluster spans both application subnets with cache nodes in each.

When you're using a cache for a backend data store, a side-cache is a common approach. Canonical examples include both Redis and Memcached. These caches are general purpose and are decoupled from the underlying data store. They can help with both read and write throughput, depending on the workload and durability requirements.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



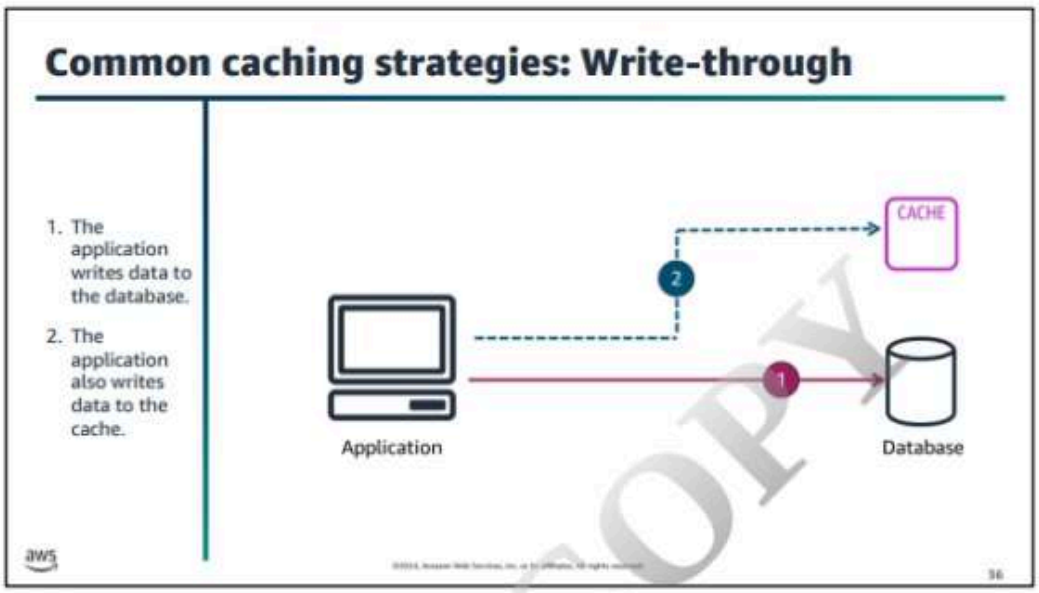
Multiple strategies are available for keeping information in the cache in sync with the database. Two common caching strategies include *lazy loading* and *write-through*.

In lazy loading, updates are made to the database without updating the cache. In the case of a cache miss, the information that is retrieved from the database can be subsequently written to the cache. Lazy loading loads data that the application needs into the cache, but it can result in high cache-miss-to-cache-hit ratios in some use cases.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



An alternative strategy is to write through to the cache every time the database is accessed. This approach results in fewer cache misses. The result is improved performance, but it requires additional storage for data that the applications might not need.

The best strategy depends on your use case. It is critical to understand the impact of stale data on your use case. If the impact is high, then consider maintaining freshness with write-throughs. If the impact is low, then lazy loading might be sufficient. It also helps to understand the frequency of change of the underlying data because it affects the performance and cost trade-offs of the caching strategies.

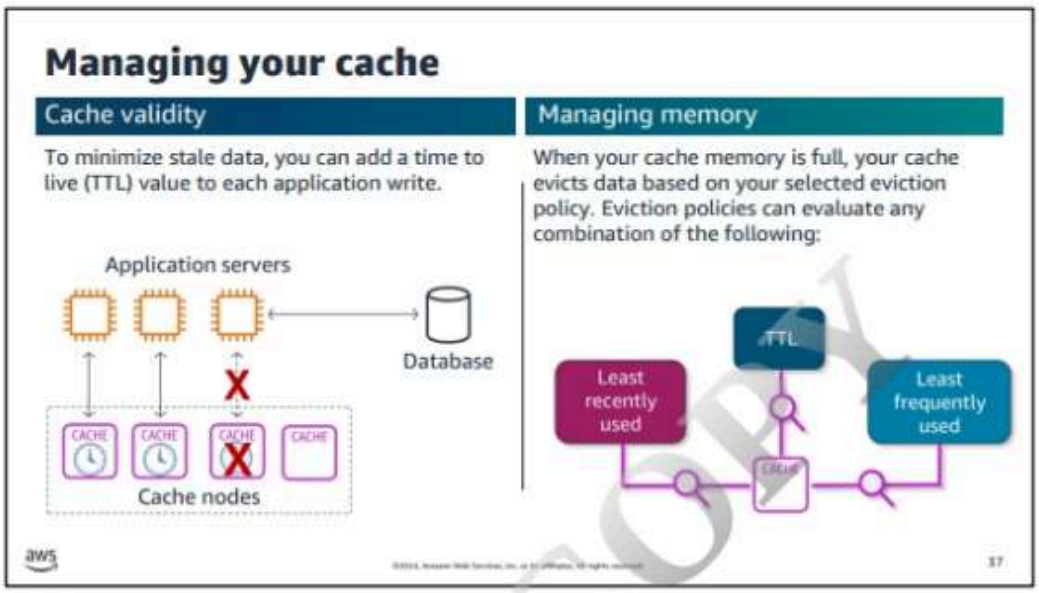
After you decide on a strategy for maintaining your cache, you will want to implement this approach within your application.

For more information about ElastiCache, see "Caching Strategies for Memcached" in the *Amazon ElastiCache User Guide* at https://docs.aws.amazon.com/AmazonElastiCache/latest/dg/Strategies.html.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Lazy loading can have stale data but doesn't fail with empty nodes. Write-through maintains fresh data but can fail with empty nodes and can populate the cache with superfluous data. By adding a time to live (TTL) value to each write to the cache, you can maintain fresh data without cluttering the cache with extra data.

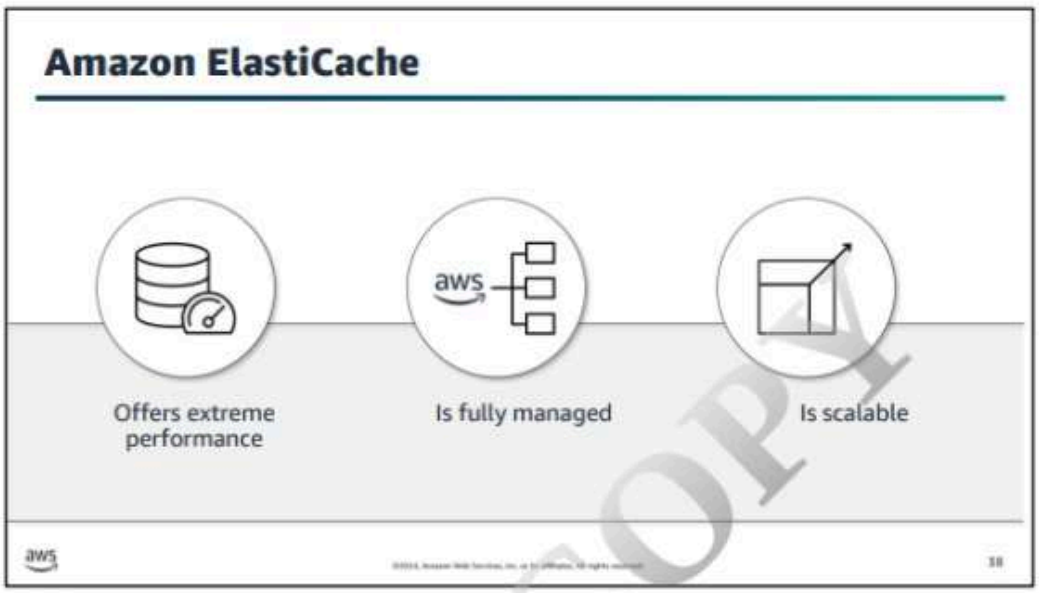
TTL is an integer value that specifies the number of seconds or milliseconds until the key expires. When an application attempts to read an expired key, it is treated as though the data is not found in cache. Therefore, the database is queried and the cache is updated. As a result, it keeps data from getting too stale and requires that values in the cache are occasionally refreshed from the database.

When the cache memory is full, the cache engine removes data from memory to make space for new data. It chooses this data based on the eviction policy that you set. An eviction policy evaluates the following characteristics of your data:

- Which data was accessed least recently?
- Which data has been accessed least frequently?
- Which data has a TTL set and the TTL value?

DO NOT COPY javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Amazon ElastiCache is a web service that facilitates setting up, managing, and scaling a distributed in-memory data store or cache environment in the cloud. It provides a high-performance, scalable, and cost-effective caching solution. At the same time, it helps remove the complexity that is associated with deploying and managing a distributed cache environment. There is also Amazon ElastiCache Serverless, a serverless option for Amazon ElastiCache that simplifies cache management and instantly scales to support the most demanding applications.

ElastiCache supports three open source in-memory engines (in-memory as compared to disk):

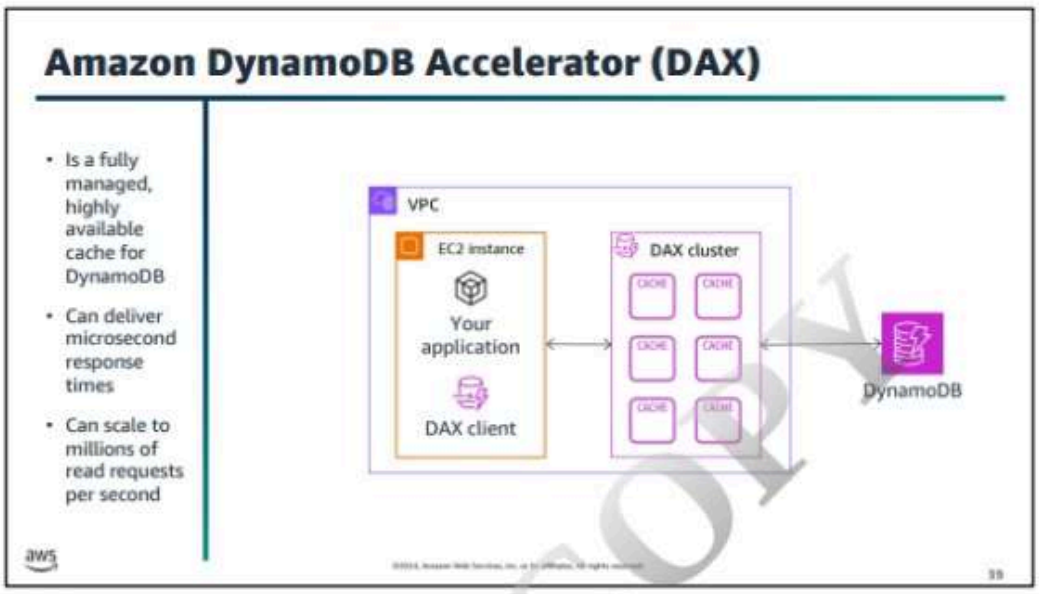
- Redis
- Memcached
- Valkey

For more information about ElastiCache, see "Amazon ElastiCache" at https://aws.amazon.com/elasticache/.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



DynamoDB is designed for scale and performance. In most cases, the DynamoDB response times can be measured in single-digit milliseconds. However, certain use cases require response times in microseconds. For those use cases, DynamoDB Accelerator (DAX) delivers fast response times for accessing eventually consistent data.

DAX is a caching service compatible with DynamoDB that provides fast in-memory performance for demanding applications.

You create a DAX cluster in a VPC in your AWS account to store cached data closer to your application. You install a DAX client on the EC2 instance that is running your application in that VPC. At runtime, the DAX client directs all of your application's DynamoDB requests to the DAX cluster. If DAX can process a request directly, it does so. Otherwise, it passes the request through to DynamoDB.

For more information about memory acceleration, see "In-Memory Acceleration with DynamoDB Accelerator (DAX)" in the *Amazon DynamoDB Developer Guide* at https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DAX.html.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

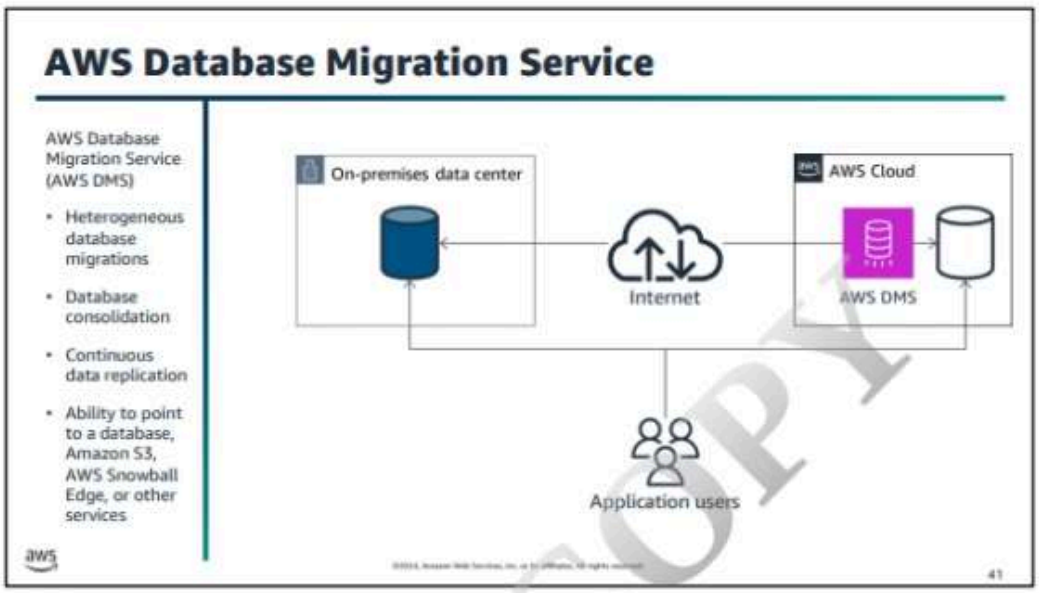
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

The database services manager asks, "What tools are available for migrating an existing database to the AWS Cloud?" The database team is planning to move some of their on-premises databases to the cloud. The company wants you to identify tools that can help them with this process and minimize downtime.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



AWS Database Migration Service (AWS DMS) replicates data from a source to a target database in the AWS Cloud. You create a source and a target connection to tell AWS DMS where to extract from and load to. Then you schedule a task that runs on this server to move your data. AWS DMS creates the tables and associated primary keys if they don't exist on the target.

AWS DMS supports migration between the most widely used databases, which include Oracle, PostgreSQL, SQL Server, Amazon Redshift, Aurora, MariaDB, and MySQL. It also supports homogenous (same engine) and heterogeneous (different engines) migrations. You can use the service to migrate between on-premises databases, Amazon EC2 databases, and Amazon RDS databases. However, you cannot migrate between two on-premises databases. Either the source or the target database (or both) must reside in Amazon RDS or on Amazon EC2.

With AWS DMS, you can also use an AWS Snowball Edge device as a migration target. You would use this method if your environment has poor internet connectivity or if the source database is too large to move over the internet. You would also use it if your organization has privacy or security requirements.

AWS DMS automatically handles formatting of the source data for consumption by the target database. It does not perform schema or code conversion.

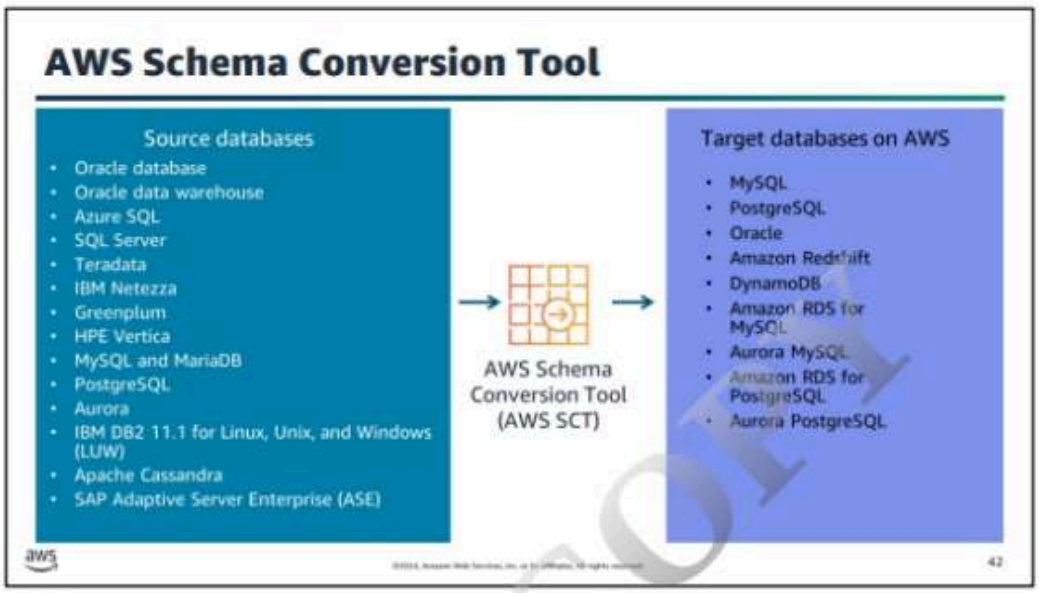
For homogenous migrations, you can use native tools to perform these conversions. For heterogeneous migrations, you can use the AWS Schema Conversion Tool (AWS SCT).

For more information about AWS DMS, see "What Is AWS Database Migration Service?" in the *AWS Database Migration Service User Guide* at https://docs.aws.amazon.com/dms/latest/userguide/Welcome.html.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



The AWS SCT makes heterogeneous database migrations predictable. It automatically converts the source database schema and a majority of the database code objects. The conversion includes views, stored procedures, and functions. They are converted to a format that is compatible with the target database. Any objects that cannot be automatically converted are marked so that they can be manually converted to complete the migration.

The AWS SCT can also scan your application source code for embedded SQL statements and convert them as part of a database schema conversion project. During this process, the AWS SCT optimizes code that is built for the cloud. It converts legacy Oracle and SQL Server functions to their equivalent AWS service, which modernizes the applications at the same time of database migration.

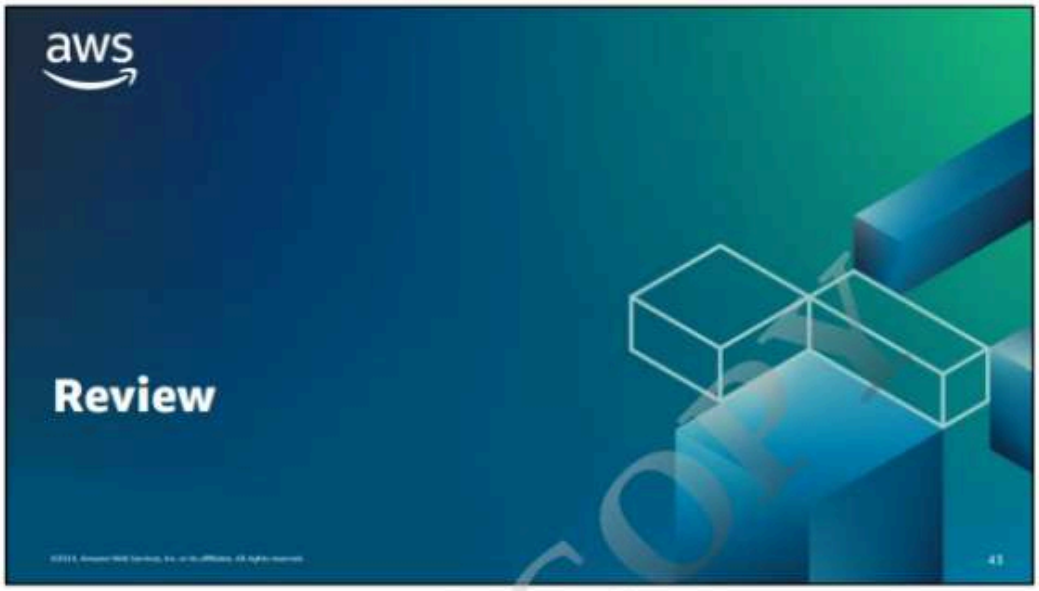
After the schema conversion is complete, the AWS SCT can help migrate data from various data warehouses to Amazon Redshift by using built-in data migration agents.

For more information about the AWS SCT, see "Schema Conversion" at https://aws.amazon.com/dms/schema-conversion-tool/.

DO NOT COPY javierartiga.espublico@gmail.com

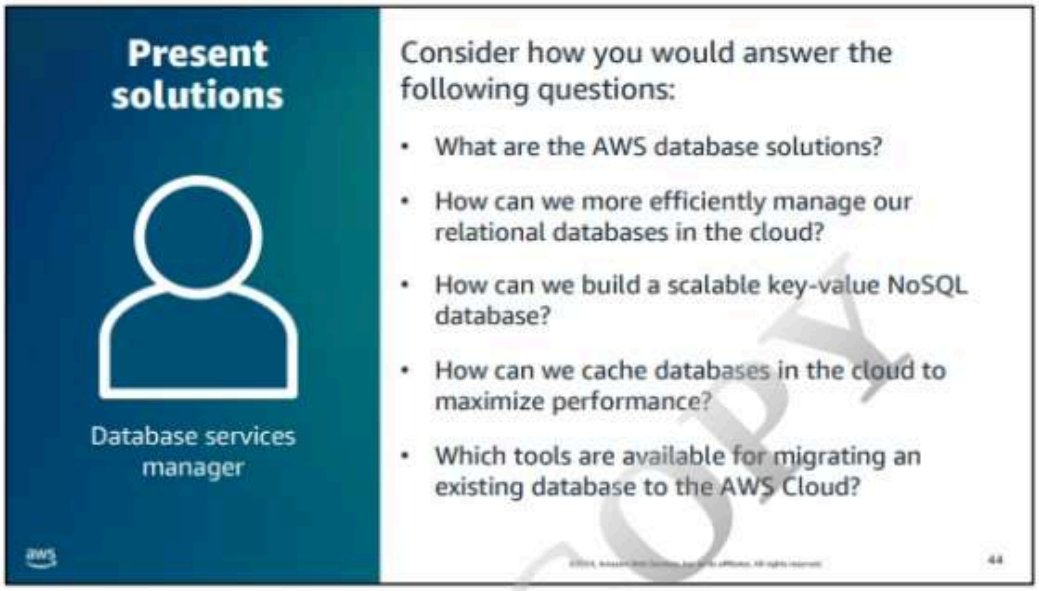
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Imagine that you are now ready to talk to the database services manager and present solutions that meet their architectural needs.

Think about how you would answer the questions from the beginning of the lesson.

Your answers should include the following solutions:
- AWS offers services that support relational and nonrelational databases.
- You can manage your relational databases efficiently with Amazon RDS and Aurora.
- You can manage your nonrelational key-value databases with DynamoDB.
- You can use ElastiCache and DAX for database caching.
- You can use AWS DMS to migrate your databases to the cloud.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Module review

In this module, you learned about the following:

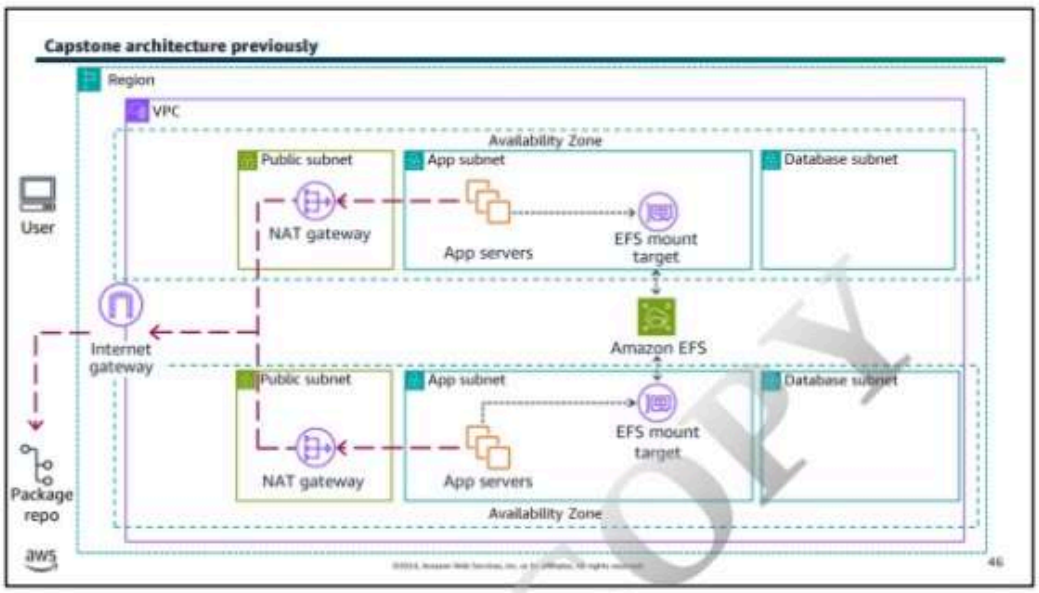
- ✓ Database services
- ✓ Amazon RDS
- ✓ DynamoDB
- ✓ Database caching
- ✓ Database migration tools

Next, you review the following:

- Capstone check-in
- Lab introduction
- Knowledge check

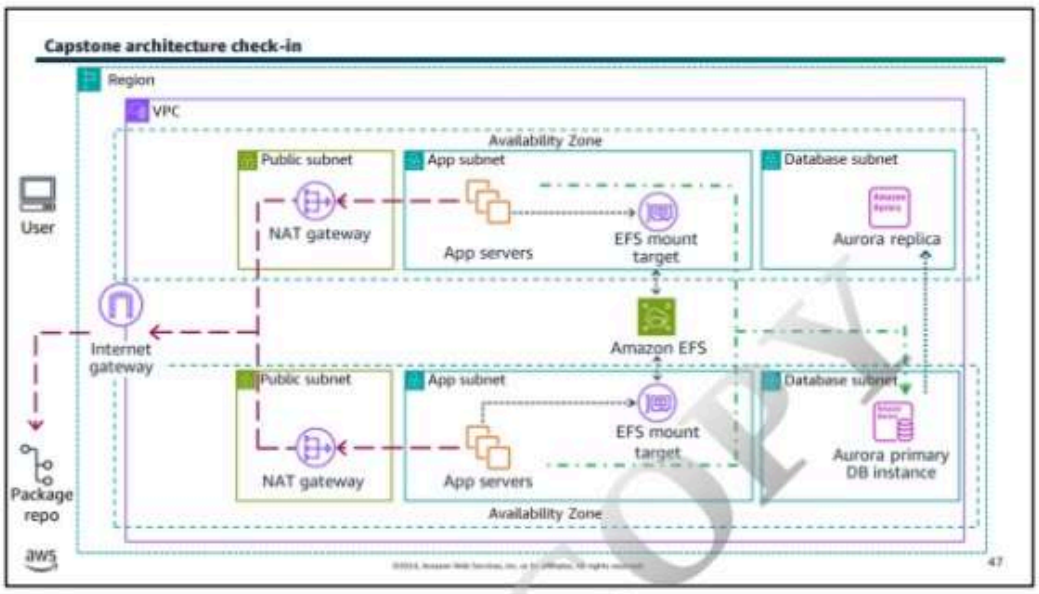
DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



In this module, you explored AWS database services and resources.

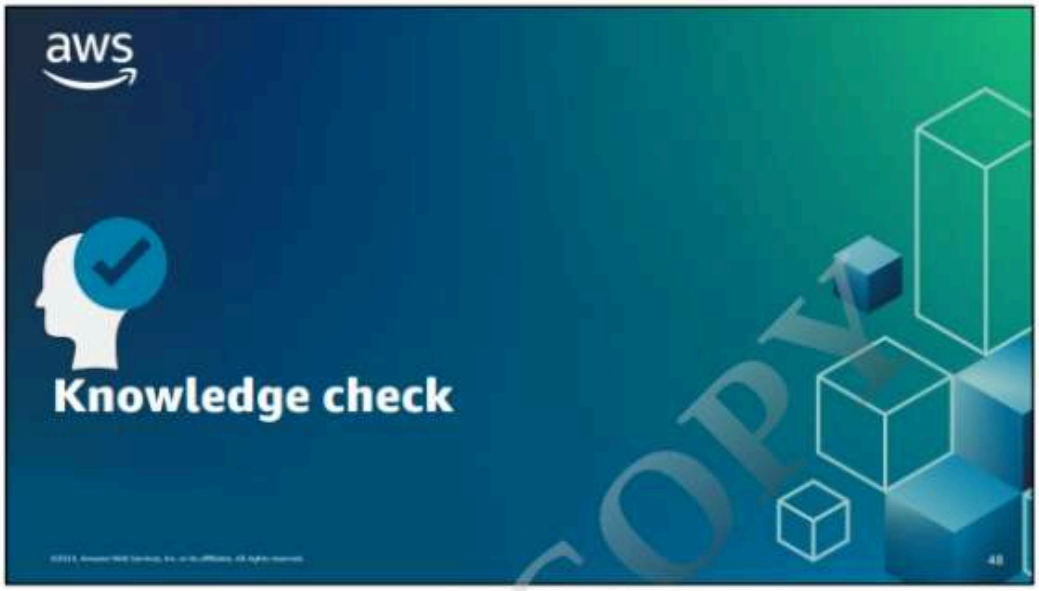
Review the capstone architecture to explore some of the design decisions. This architecture helps you provide the following benefits:

- In the capstone, you set up an application that requires a MySQL database. Aurora supports this engine. Using a managed service for hosting the database also provides the following benefits:
  - You do not have to manage server OS patches or database software updates.
  - Aurora helps you configure scaling, backups, and high availability.
- The capstone architecture achieves resiliency by using an Aurora replica. If the primary instance fails, the replica can be promoted to primary.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 1

What is a benefit of using Amazon RDS in a Multi-AZ configuration?

Choice	Response
A	It delivers two live copies of the database running concurrently.
B	It provides automatic failover across Availability Zones.
C	It provides automatic cross-Region replication.
D	It eliminates the need for read replicas.

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 1 and answer

What is a benefit of using Amazon RDS in a Multi-AZ configuration?

Choice	Response
A	It delivers two live copies of the database running concurrently.
B correct	It provides automatic failover across Availability Zones.
C	It provides automatic cross-Region replication.
D	It eliminates the need for read replicas.

The correct answer is B: It provides automatic failover across Availability Zones.

When you provision a Multi-AZ DB instance, Amazon RDS synchronously replicates the data to a standby instance in a different Availability Zone. In case of the primary instance failure, Amazon RDS performs an automatic failover to the standby instance. Database operations can be resumed as soon as the failover is complete. Because the endpoint for your DB instance remains the same after a failover, your application can resume database operation. You don't need to manually intervene in the administration.

DO NOT COPY javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 2

Which statement is true regarding Amazon DynamoDB global tables?

Choice	Response
A	Tables are updated manually or through automation tools.
B	Only two tables are active at one time.
C	Select different instance sizes to adjust performance.
D	Tables can be in different AWS Regions.

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 2 and answer

Which statement is true regarding Amazon DynamoDB global tables?

Choice	Response
A	Tables are updated manually or through automation tools.
B	Only two tables are active at one time.
C	Select different instance sizes to adjust performance.
D correct	Tables can be in different AWS Regions.

The correct answer is D: Tables can be in different AWS Regions.

When you create a global table, you specify the Regions where you want the table to be available. DynamoDB performs all the necessary tasks to create identical tables in these Regions and to propagate ongoing data changes to all of them.

For more information about Amazon DynamoDB global tables, see "Amazon DynamoDB Global Tables" at https://aws.amazon.com/dynamodb/global-tables/.

DO NOT COPY javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 3

Which statement is true regarding an Amazon Aurora database?

Choice	Response
A	Nine copies of the data are stored across three Availability Zones.
B	Aurora has a limit of five replicas.
C	Aurora is compatible with MySQL or PostgreSQL.
D	Multi-AZ deployments are not required for high availability.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 3 and answer

Which statement is true regarding an Amazon Aurora database?

Choice	Response
A	Nine copies of the data are stored across three Availability Zones.
B	Aurora has a limit of five replicas.
C correct	Aurora is compatible with MySQL or PostgreSQL.
D	Multi-AZ deployments are not required for high availability.

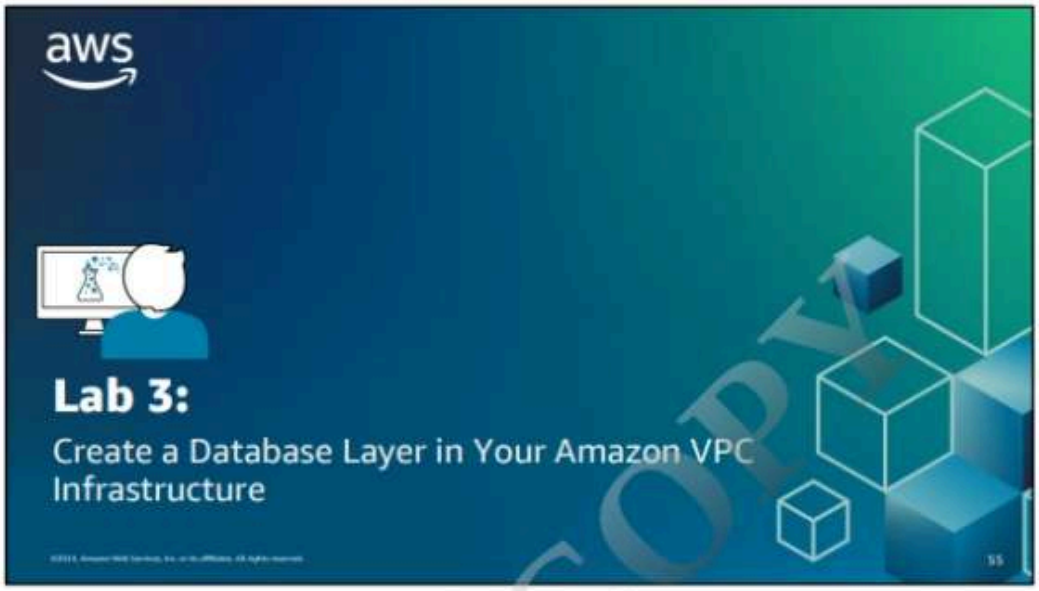
The correct answer is C: Aurora is compatible with MySQL or PostgreSQL.

Aurora is an enterprise-class relational database. It is compatible with MySQL or PostgreSQL relational databases. It is up to five times faster than standard MySQL databases and up to three times faster than standard PostgreSQL databases.

For more information about Aurora storage, see "Amazon Aurora Storage" at https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Overview.StorageReliability.html.

DO NOT COPY
javierartiga.espublico@gmail.com

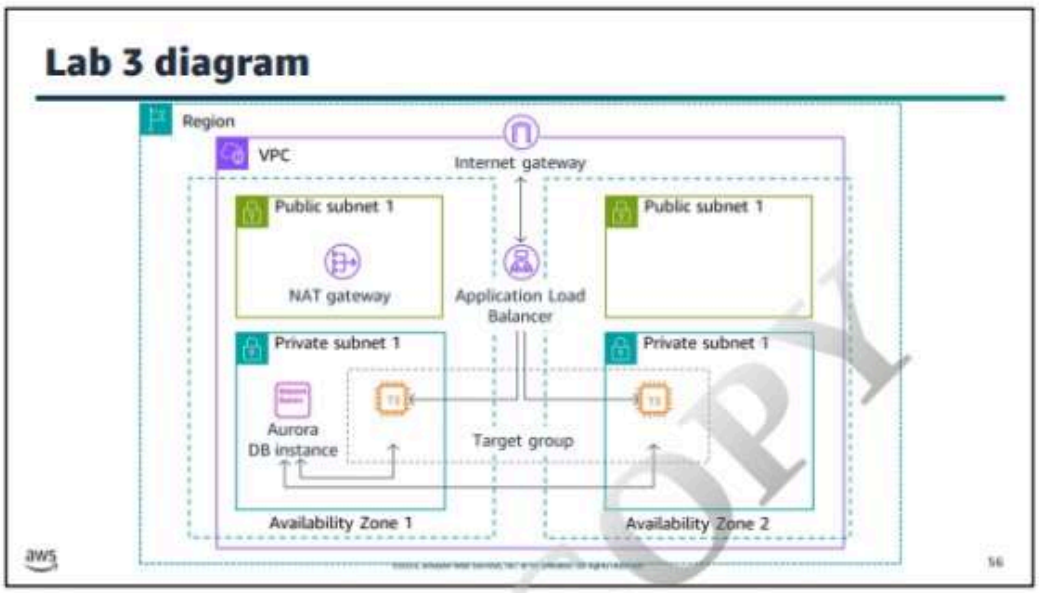
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Lab 3 uses a VPC across two Availability Zones with a private and public subnet in each. Each private subnet contains a T3 EC2 instance. One private subnet contains an Aurora DB instance, which connects to a target group that consists of the two EC2 instances. Inbound traffic routes through an internet gateway into an Application Load Balancer. The load balancer distributes traffic between the EC2 instances.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Lab tasks

Task 1: Create an Amazon RDS database

Task 2: Create and configure an Application Load Balancer

Task 3: Review the Amazon RDS DB instance metadata through the console

Task 4: Test the application connectivity to the database

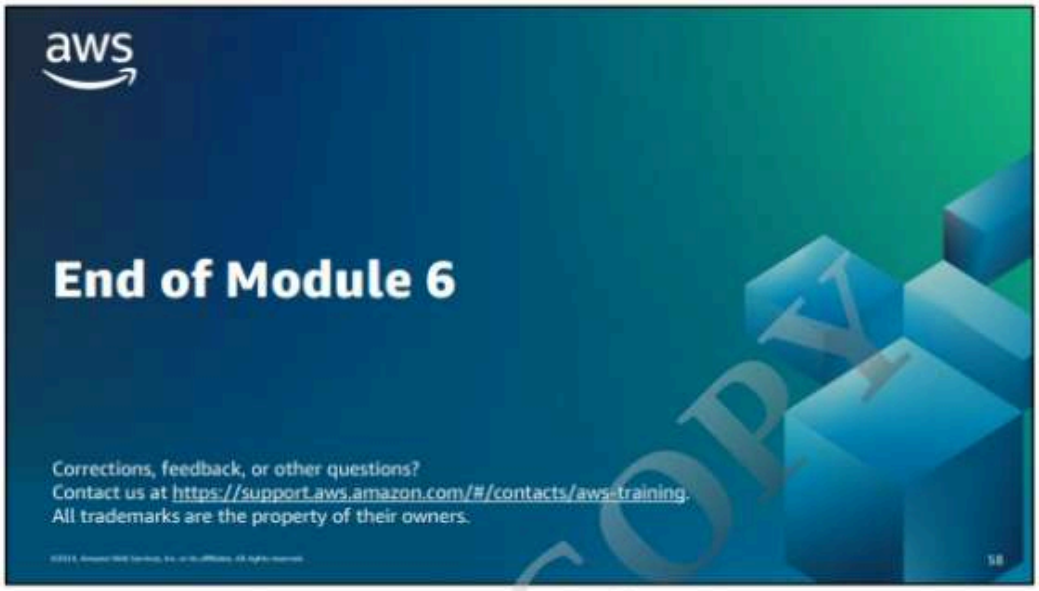
Optional Task: Create an Amazon RDS read replica in a different AWS Region

57

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

**Accessibility note for screen reader users: To make code examples in this presentation more accessible, Start of code and End of code labels have been added before and after blocks of code on slides in this presentation.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Poll question

What factors impact your decisions about making your workloads scalable?

A. Cost

B. Usage patterns

C. Expected growth

D. Criticality of workload

E. All of the above

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Module overview

- Business request
- Monitoring
- Alarms and events
- Load balancing
- Auto scaling
- Present solutions
- Capstone check-in
- Knowledge check
- Lab 4: Configure High Availability in Your Amazon VPC

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

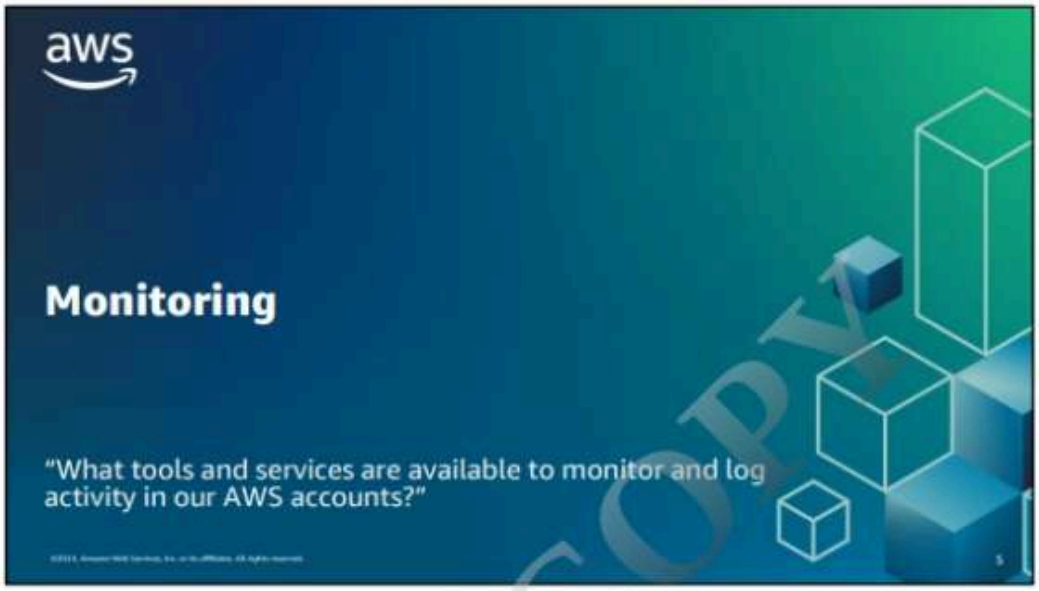
Imagine that your operations manager meets with you to discuss how to monitor resources and scale operations in your Amazon Web Services (AWS) accounts. Here are some questions that they are asking about monitoring and scaling.

At the end of this module, you meet with the operations manager and present some solutions.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

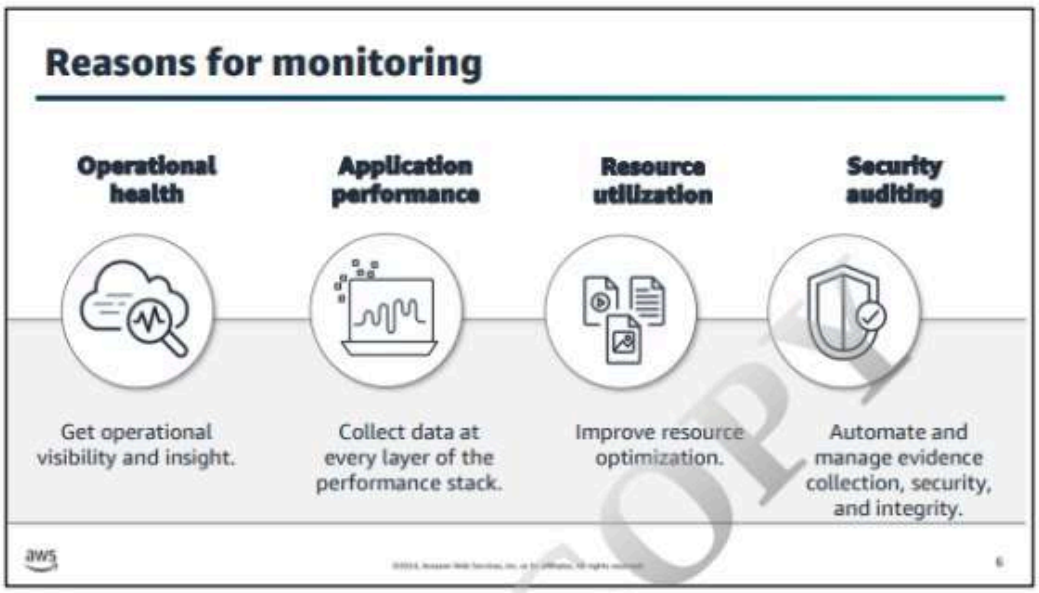


The operations manager asks, "What tools and services are available to monitor and log activity in our AWS accounts?" You and the operations team must examine tools that are used in monitoring activity so that you can help build an easy-to-manage architecture.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Monitoring your environment is one of the most important things to think about when creating architecture. You will always need a way to keep track of how your resources are operating and performing. Monitoring gives you the information to answer the question, Does something need to change?

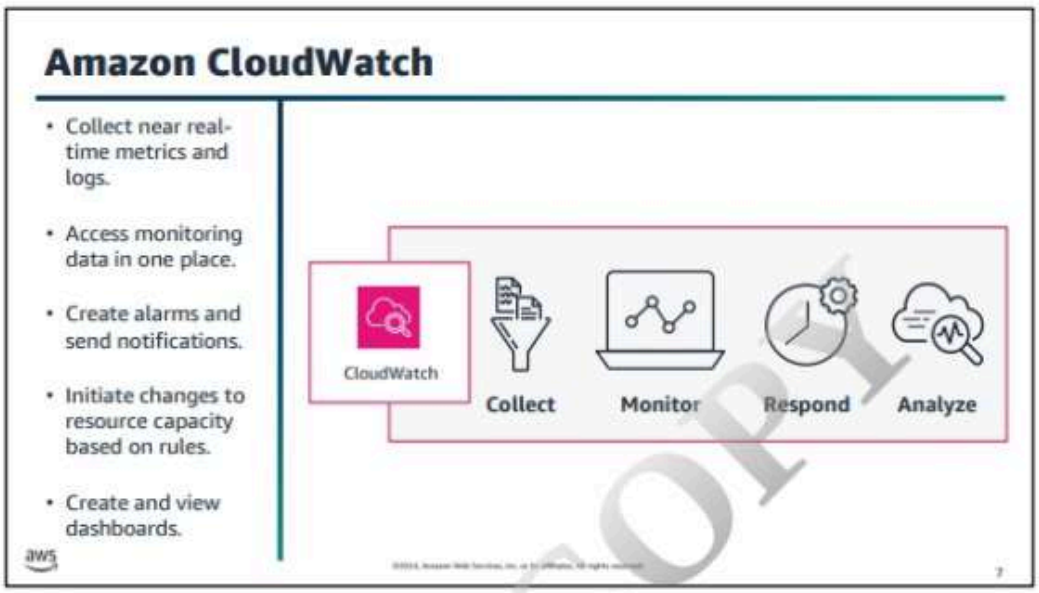
Here are a few points to remember:

- With monitoring, you gather information about resource utilization and application performance. Monitoring measures whether your infrastructure is satisfying demand. It helps you build an architecture that scales up for more demand and pulls back when there is less demand.
- Scaling with the use of monitoring provides a better user experience for your customers and saves you money.
- Monitoring is important for security. With parameters in place, you can see if and when users are accessing parts of your environment, and verify permissions.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Amazon CloudWatch is an AWS service that provides a near real-time stream of system events. The events describe changes to your AWS resources. With CloudWatch, you can respond quickly to operational changes and take corrective action. CloudWatch alarms send notifications or automatically make changes to the resources that you are monitoring based on rules that you define.

For example, you can monitor the CPU usage and disk reads and writes of your Amazon EC2 instances. You can use this data to determine whether you should launch additional instances to handle increased load. You can also use this data to stop underused instances to save money. Besides monitoring the built-in metrics that come with AWS, you can create and monitor custom metrics. With CloudWatch, you gain system-wide visibility into resource use, application performance, and operational health.

You can collect, access, and correlate this data in one place from across all of your AWS resources, applications, and services. CloudWatch also collects from on-premises servers. To optimize performance and resource use, CloudWatch provides automatic dashboards, data with one-second granularity, and up to 15 months of metrics storage and retention.

For more information about CloudWatch, see "What is Amazon CloudWatch?" in the *Amazon CloudWatch User Guide* at https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/WhatIsCloudWatch.html.

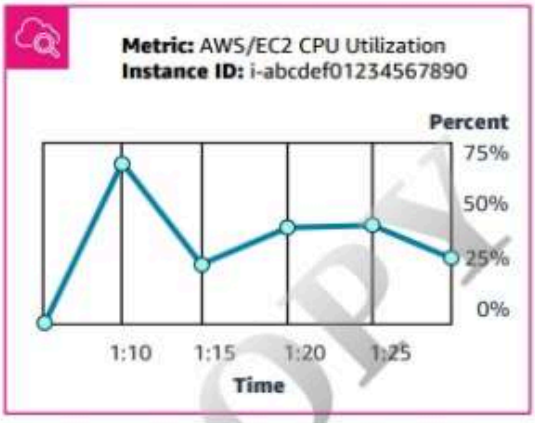
DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

CloudWatch metrics

- Metrics are data about system performance.
- CloudWatch ingests and tracks metrics so that you can search and visualize data.



Metrics are data about the performance of your systems. By default, many services provide you with metrics for resources. For example, these resources might include Amazon EC2 instances, Amazon Elastic Block Store (Amazon EBS) volumes, and Amazon Relational Database Service (Amazon RDS) DB instances. CloudWatch stores data about a metric as a series of data points. Each data point has an associated timestamp.

You can publish your own metrics to CloudWatch. Use the AWS Management Console to view statistical graphs of your published metrics. You can also turn on detailed monitoring for some resources, such as your Amazon EC2 instances, or publish your own application metrics. Amazon CloudWatch can load all of the metrics in your account (both AWS resource metrics and application metrics that you provide) for search, graphing, and alarms. Metric data is kept for 15 months, so you can view both up-to-the-minute data and historical data.

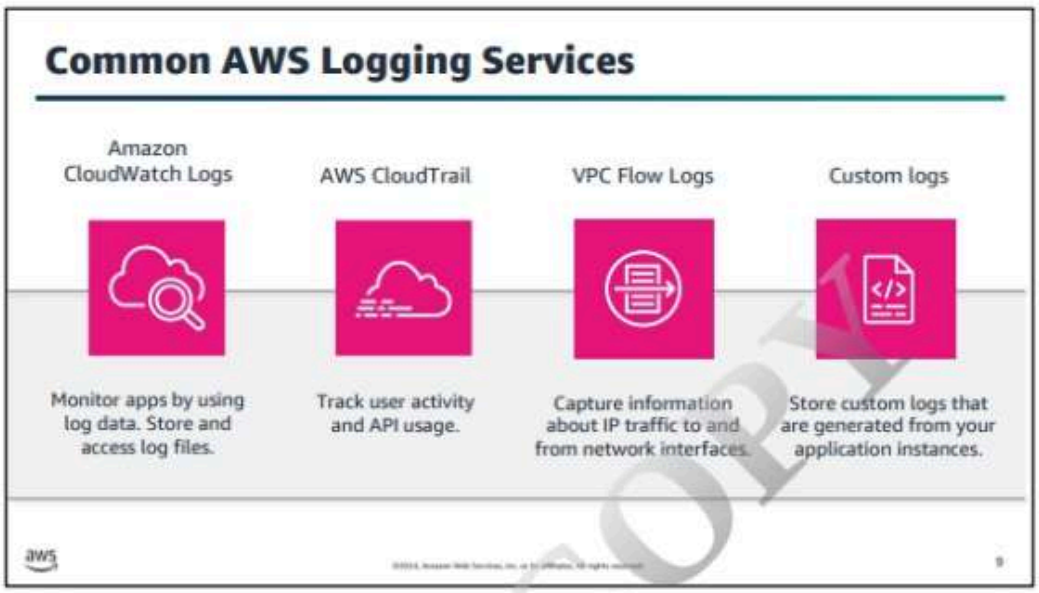
A CloudWatch metric includes the following components:

- A *namespace* is a container for CloudWatch metrics. Metrics in different namespaces are isolated from each other so that metrics from different applications are not mistakenly aggregated into the same statistics. You can specify a namespace name when you create a metric. The AWS namespaces use the following naming convention: AWS/*service*. In the example, the namespace is AWS/EC2.
- A *metric* represents a time-ordered set of data points that are published to CloudWatch. Think of a metric as a variable to monitor, and the data points represent the values of that variable over time. Each metric data point must be associated with a *timestamp*. If you do not provide a timestamp, CloudWatch creates a timestamp for you based on the time that the data point was received.
- A *dimension* is a name-value pair that uniquely identifies a metric. You can assign up to 10 dimensions to a metric. Every metric has specific characteristics that describe it. You can think of dimensions as categories for those characteristics. In the example, the dimension is the instance ID.

To graph metrics in the console, you can use CloudWatch Metrics Insights, a high-performance SQL query engine. Use it to identify real-time trends and patterns within all of your metrics.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Plan for logging as you build. Review the following services to understand how they support logs:

- **Amazon CloudWatch Logs** monitor, store, and access your log files from Amazon EC2 instances, AWS CloudTrail, Amazon Route 53, and other resources.
- **AWS CloudTrail** provides an event history of your account activity, including actions taken through the console, AWS SDK, command line interface (CLI), and AWS services. This event history simplifies security analysis, resource change tracking, and troubleshooting. CloudTrail facilitates governance, compliance, and operational and risk auditing. With CloudTrail, you can continuously monitor, log, and retain account activity that is related to actions across your AWS infrastructure.
- **VPC Flow Logs** captures information about the IP traffic that goes to and from network interfaces in your virtual private cloud (VPC).
- **Load balancing** provides access logs that capture detailed information about requests that are sent to your load balancer. You can use **custom logs** from your applications.

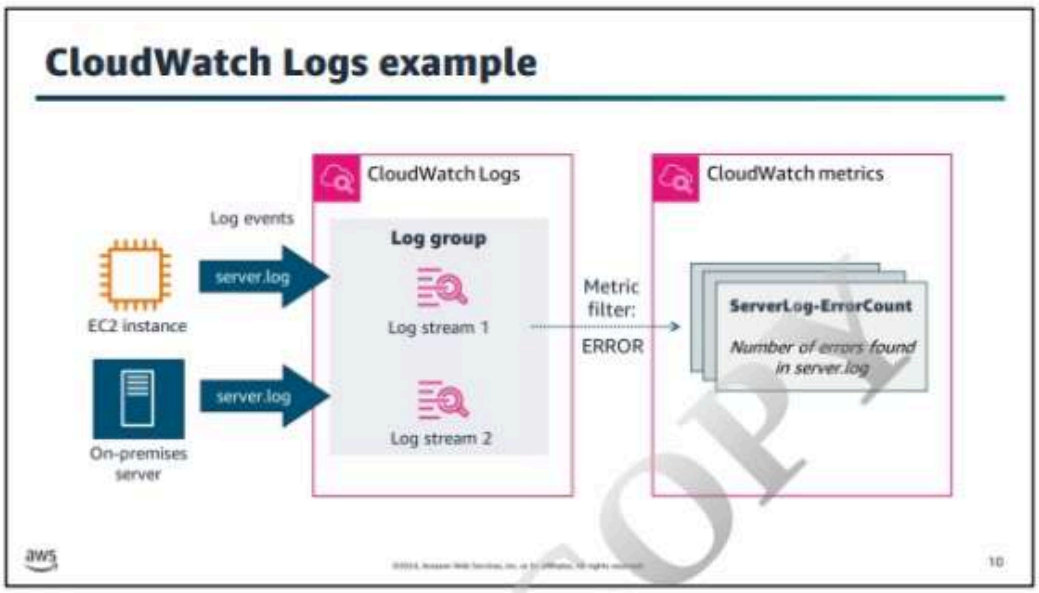
For more information about logging and events, see "Logging and Events" in *AWS Security Incident Response Guide* in the *AWS Technical Guide* at https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/logging-and-events.html.

For more information about application log files, see "Store and Monitor OS & Application Log Files with Amazon CloudWatch" in the *AWS News Blog* at https://aws.amazon.com/blogs/aws/cloudwatch-log-service/.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



The example shows two identical web servers that record data to the same log file, which is called server.log. One server is in Amazon EC2, and the other is a virtual machine on premises. Both servers are able to publish events in the log file to a *log stream*. A log stream is a sequence of log events that share the same source. Each separate source of logs in CloudWatch Logs makes up a separate log stream.

Multiple log streams can be collected in a single *log group*. A log group shares the same retention, monitoring, and access control settings across all streams. You can define log groups and specify which streams to put into each group. There is no limit to the number of log streams that can belong to one log group.

After you create a log group, you can use *metric filters* to search for and match terms, phrases, or values in your log events. When a metric filter finds one of the terms, phrases, or values in your log events, you can increment the value of a CloudWatch metric. For example, you can count occurrences of a single term, such as the word *error*, in a metric called *LogFile-ErrorCount*.

Metrics can be monitored and invoke alarms, which is covered later in this module.

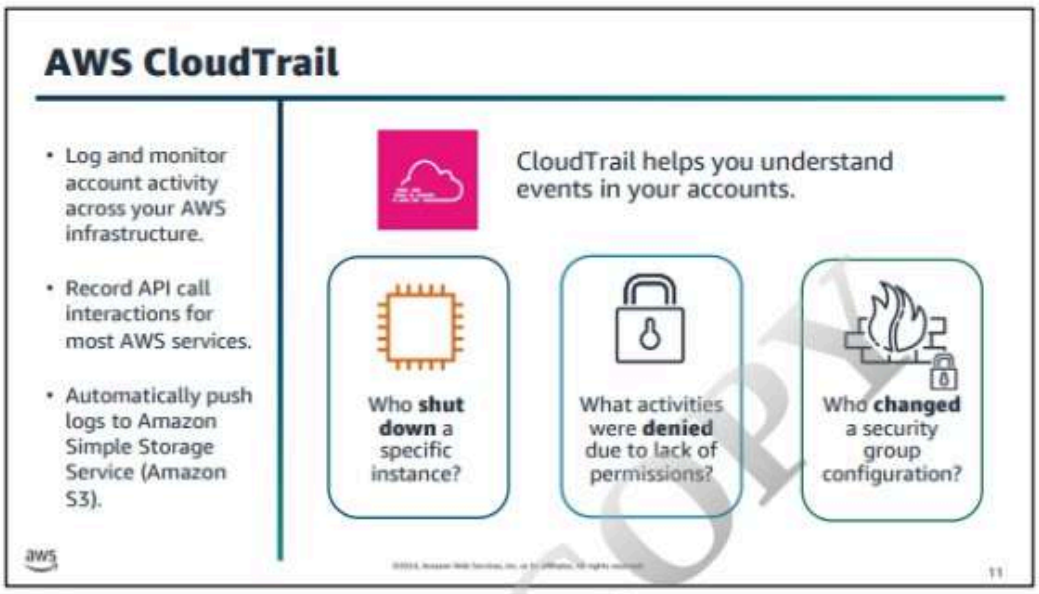
For more information about CloudWatch log functionality, see "Working with log groups and log streams" in the *Amazon CloudWatch Logs User Guide* at https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/Working-with-log-groups-and-streams.html.

To learn more about collecting metrics and logs, see "Collect metrics, logs, and traces with the CloudWatch agent" in *Amazon CloudWatch User Guide* at https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Install-CloudWatch-Agent.html.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



AWS CloudTrail provides insight into who did what and when by tracking user activity and API usage. With CloudTrail, you can get a history of AWS API calls in your account. These calls can be made through the console, AWS SDKs, CLI, and higher-level AWS services. CloudTrail records the AWS API calls and delivers the log files to you. The information includes the source IP address and identity of the API caller. It also includes the time of the call, the request parameters, and the response elements that the AWS service returns.

The AWS API call history that CloudTrail produces can facilitate security analysis, tracking of resource changes, and compliance auditing.

You turn on CloudTrail on a per-Region basis. If you use multiple Regions, you can choose where log files are delivered for each Region. CloudTrail saves the logs in your designated Amazon Simple Storage Service (Amazon S3) bucket. You might have a separate S3 bucket for each Region, or you can aggregate log files from all Regions into a single S3 bucket.

CloudTrail helps you answer questions that require detailed analysis. Store your CloudTrail API usage logs in an S3 bucket. You can analyze those logs later to answer compelling questions, such as the following:

- Why was a long-running instance terminated and who terminated it? (*Organizational traceability and accountability*)
- Who changed a security group configuration? (*Accountability and security auditing*)
- What activities were denied due to lack of permissions? (*Potential internal or external attack against the network*)

For more information, see "CloudTrail supported services and integrations" in the *AWS CloudTrail User Guide* at http://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-supported-services.html.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Example: CloudTrail log (1 of 4)

```
{
    "Records": [{
        "eventVersion": "1.0",
        "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::123456789012:user/Alice",
            "accountId": "123456789012",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
        },
```

Who made the request?

This example log file shows that an AWS Identity and Access Management (IAM) user named Alice called the EC2 StopInstances action. Alice used the ec2-stop-instances command in AWS Command Line Interface (AWS CLI). In this section of the log, you get information about who made the request.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Example: CloudTrail log (2 of 4)

```
"requestParameters": {
        "instancesSet": {
            "items": [{
                "instanceId": "i-abcdefg01234567890"
            }]
        },
        "force": false
    },
```

What was the focus of the request?

In this section of the script, you get information about the focus of the request. In this case, the focus of the request was an instance, and you can see the instance ID: i-abcdefg01234567890.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Example: CloudTrail log (3 of 4)

```
"eventTime": "2018-03-06T21:01:59Z",
"eventSource": "ec2.amazonaws.com",
"eventName": "StopInstances",
"awsRegion": "us-west-2",
"sourceIPAddress": "205.251.233.176",
"userAgent": "ec2-api-tools 1.6.12.2",
```

When did the request occur?

What was the API call?

Where did it occur?

14

This section of the script can tell you when the API call occurred, what the API call was (StopInstances), and in what Region it occurred.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Example: CloudTrail log (4 of 4)

```
"responseElements": {
    "instancesSet": {
        "items": [{
            "instanceId": "i-abcdefg01234567890",
            "currentState": {
                "code": 64,
                "name": "stopping"
            },
            "previousState": {
                "code": 16,
                "name": "running"
            }
```

What was the response?

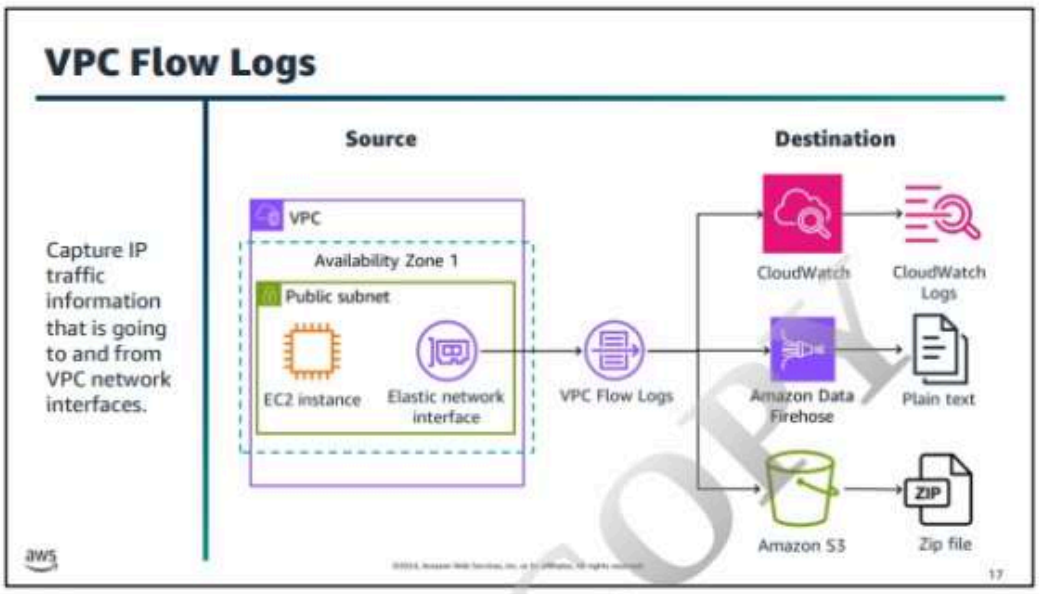
15

In this section of the script, you get information about the response. In this case, the instance was stopped.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



VPC Flow Logs captures IP traffic information to and from VPC network interfaces.

- Flow logs can be configured to record traffic per VPC, subnet, or network interface.
- You can view information about your flow logs in the Amazon EC2 and Amazon Virtual Private Cloud (Amazon VPC) consoles by choosing the **Flow Logs** tab for a specific resource.
- Flow logs are turned off by default. You must opt in to collect flow log data.

Flow logs are published to either Amazon S3 buckets, CloudWatch log groups, or Amazon Data Firehose delivery streams. Data is collected outside the path of your network traffic, and therefore does not affect network throughput or latency. Flow logs can help you perform several tasks, such as the following:

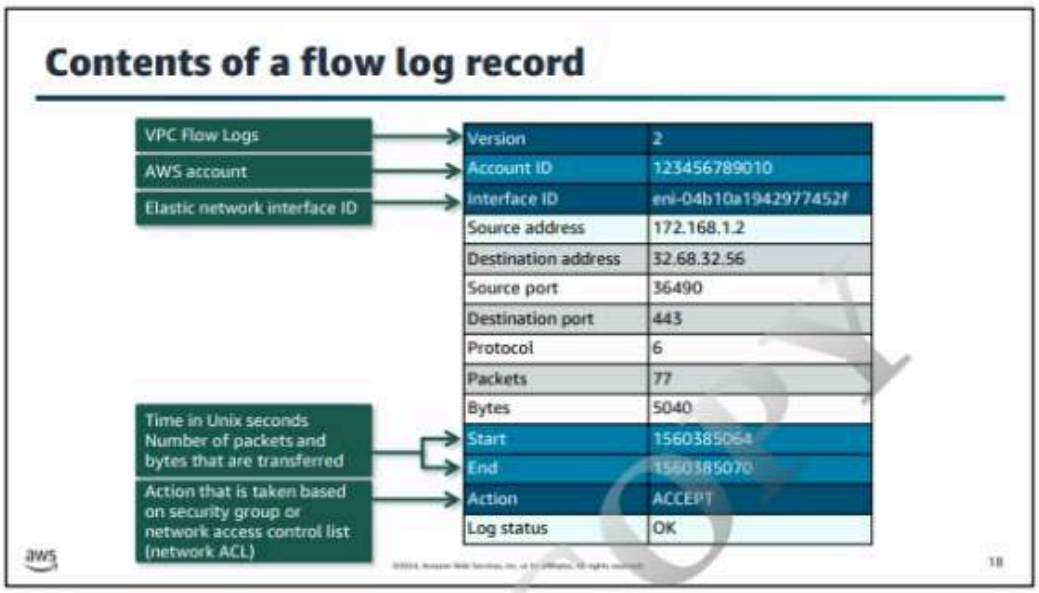
- Diagnose overly restrictive security group rules.
- Monitor the traffic that is reaching your instance.
- Determine the direction of the traffic to and from the network interfaces.

For more information about flow logs, see "Logging IP traffic using VPC Flow Logs" in the *Amazon Virtual Private Cloud User Guide* at https://docs.aws.amazon.com/vpc/latest/userguide/flow-logs.html.

To learn about improving security by analyzing VPC flow logs, see "Improve security by analyzing VPC flow logs with Amazon CloudWatch Contributor Insights" in the *AWS Cloud Operations & Migrations Blog* at https://aws.amazon.com/blogs/mt/improve-security-by-analyzing-vpc-flow-logs-with-amazon-cloudwatch-contributor-insights/.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Each flow log record captures a network IP traffic flow for a specific capture window and contains five different values, also known as a 5-tuple. A record includes different components of IP flow, such as the source, destination, and protocol. You can create alarms that will activate if certain types of traffic are detected, and metrics to help you identify trends and patterns.

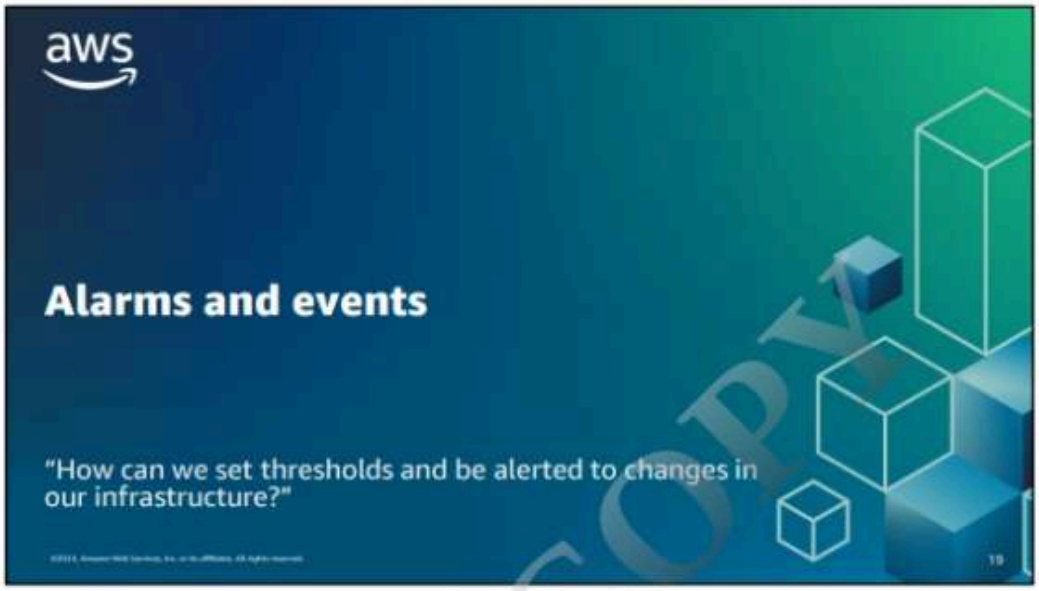
The information includes an ACCEPT or REJECT action, based on security group and network access control list (network ACL) rules. It also includes source and destination IP addresses, ports, and the Internet Assigned Numbers Authority (IANA) protocol number. In addition, it includes packet and byte counts (a time interval during which the flow was observed).

Flow logs don't capture everything in your network. VPC logging does not include DNS traffic or Dynamic Host Configuration Protocol (DHCP) requests and responses. If you're running your own DNS server, you can log request resolution traffic. But many users rely on internal AWS DNS servers, and VPC Flow Logs will not capture activity between the servers and AWS DNS services. DHCP traffic is also not recorded.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



The operations manager asks, "How can we set thresholds and be alerted to changes in our infrastructure?" The operations team must examine AWS services that are used to help automate actions based on events.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

CloudWatch alarms

1. Identify the CloudWatch metric.
2. Create your alarms based on metrics.
3. Define the actions to take when your metric's threshold is exceeded.

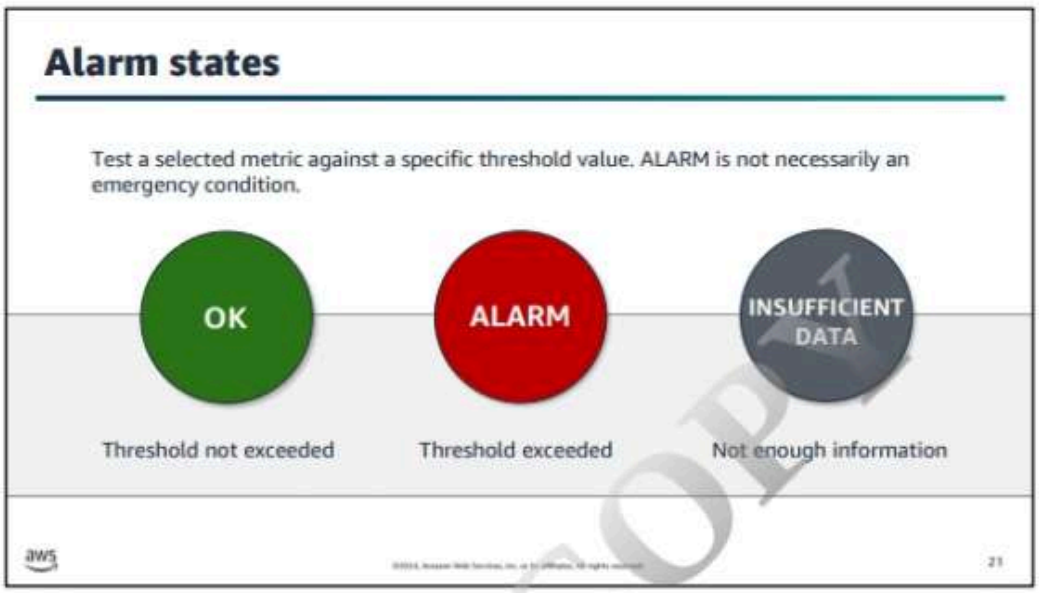
20

A metric alarm watches a single CloudWatch metric. The alarm performs one or more actions based on the value of the metric relative to a threshold over a number of time periods. The action can be an Amazon EC2 action, an auto scaling action, or a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



An alarm has three possible states:

- **OK** – The metric is within the defined threshold.
- **ALARM** – The metric is outside the defined threshold.
- **INSUFFICIENT_DATA** – The alarm has started, the metric is not available, or not enough data is available for the metric to determine the alarm state.

ALARM is only a name that is given to the state and does not necessarily signal an emergency condition that requires immediate attention. It means that the monitored metric is equal to, greater than, or less than a specified threshold value. You could, for example, define an alarm that tells you when your CPU utilization for a given EC2 instance is too high. You might process this notification programmatically to suspend a CPU-intensive job on the instance. You can also send a notification to notify the application owner to take action.

INSUFFICIENT_DATA can be returned when no data exists for a given metric. An example is the depth of an empty Amazon Simple Queue Service (Amazon SQS) queue. It can also be an indicator that something is wrong in your system.

DO NOT COPY javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Alarm components

Metric name	Instance ID	Statistic	Period	Data points to alarm
CPUUtilization	i-abcdef012345	Average	5 minutes	2 of 2

To create an alarm based on a metric math expression, choose one or more CloudWatch metrics to use in the expression. Then, specify the expression, threshold, and evaluation periods.

When configuring an alarm in Amazon CloudWatch, there are several key components that you need to understand and define. These components determine how CloudWatch evaluates the metric data and triggers the alarm based on the specified conditions.

- *Statistics* are metric data aggregations over specified periods of time. For a metric alarm, it is evaluated on one statistic. Some of the most common statistics that you can choose are sample count, sum, average, maximum, minimum, and percentile.
- *Period* is the length of time to evaluate the metric or expression to create each individual data point for an alarm. It is expressed in seconds. If you choose one minute as the period, the alarm evaluates the metric once per minute.
- *Evaluation Periods* is the number of the most recent periods, or data points, to evaluate when determining the alarm state. In the example table, this number is the second number in the *2 of 2* value.
- *Datapoints to Alarm* is the number of data points within the Evaluation Periods that must be breaching to cause the alarm to reach ALARM state. The breaching data points don't need to be consecutive. However, they must all be within the last number of data points equal to the Evaluation Period.

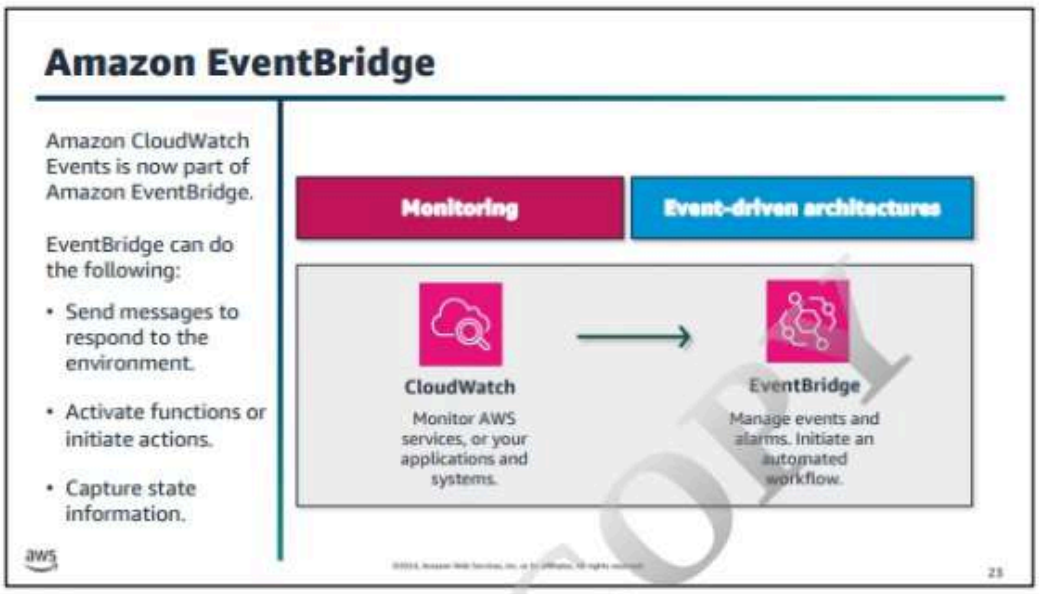
In the example, the alarm threshold is set to 25 percent and the minimum breach is two periods. That is, the alarm state changes, and it invokes its action only when the threshold is breached for two consecutive periods. You control this value. You can also set how the alarm treats missing data points. In some cases, you might want to build your alarm to go to the INSUFFICIENT_DATA state for missing data rather than an ALARM state.

In the metric graph, this circumstance happens with the third through fourth and fifth periods. The alarm's state is set to ALARM at 1:20 PM. At period six, the value dips below the threshold and the state reverts to OK at 1:25 PM.

For more information about CloudWatch alarms, see "Using Amazon CloudWatch alarms" in the *Amazon CloudWatch User Guide* at https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/AlarmThatSendsEmail.html.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Amazon EventBridge removes the friction of writing *point-to-point* integrations. You can access changes in data that occur in both AWS and software as a service (SaaS) applications through a highly scalable, central stream of events.

EventBridge is the preferred way to manage your events that are captured in CloudWatch. CloudWatch Events and EventBridge are the same underlying service and API, but EventBridge provides more features. Changes that you make in either CloudWatch or EventBridge will appear in each console.

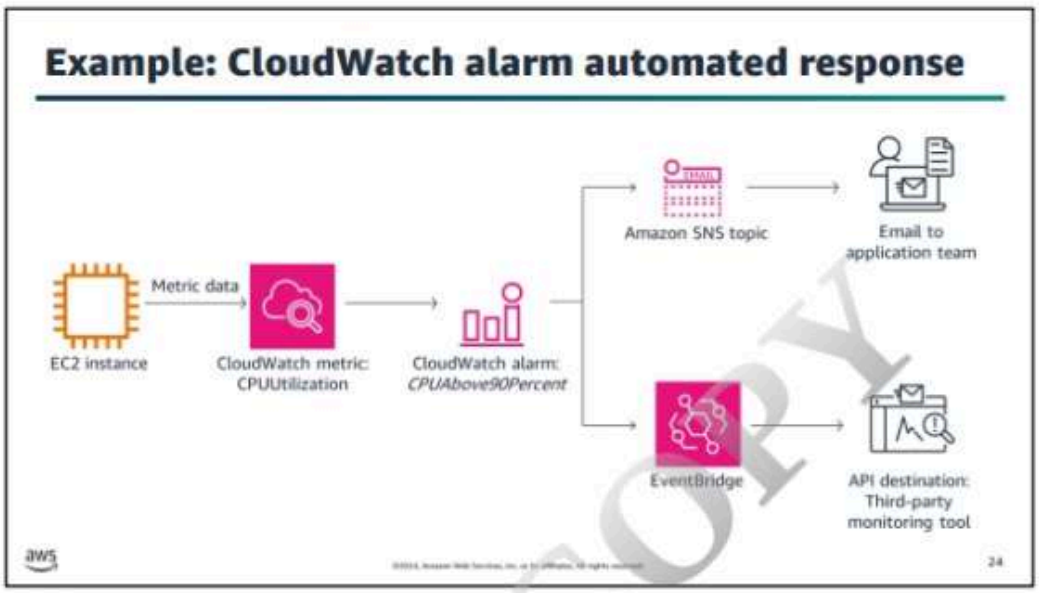
With EventBridge, you get a programming model where event publishers are decoupled from event subscribers. You can build loosely coupled, independently scaled, and highly reusable event-driven applications.

Because it's fully managed, it handles everything from event ingestion and delivery to security, authorization, and error-handling, so you can build scalable, event-driven applications. EventBridge is serverless, so you have no infrastructure to manage and you pay for only the events that your own applications or SaaS applications generate.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



In the example, an EC2 instance reports the CPUUtilization metric data to CloudWatch. A custom alarm is created and configured, called *CPUAbove90Percent*, so that you will know when the EC2 instance is being overused.

EventBridge rules are built to notify your support team when the CPUAbove90Percent alarm is in ALARM state, so that they can investigate and take action. EventBridge takes two actions. It sends an email to subscribed recipients by using the Amazon SNS topic, and it sends a rich notification to the operation team's third-party monitoring tool.

Think about how you could take this further:

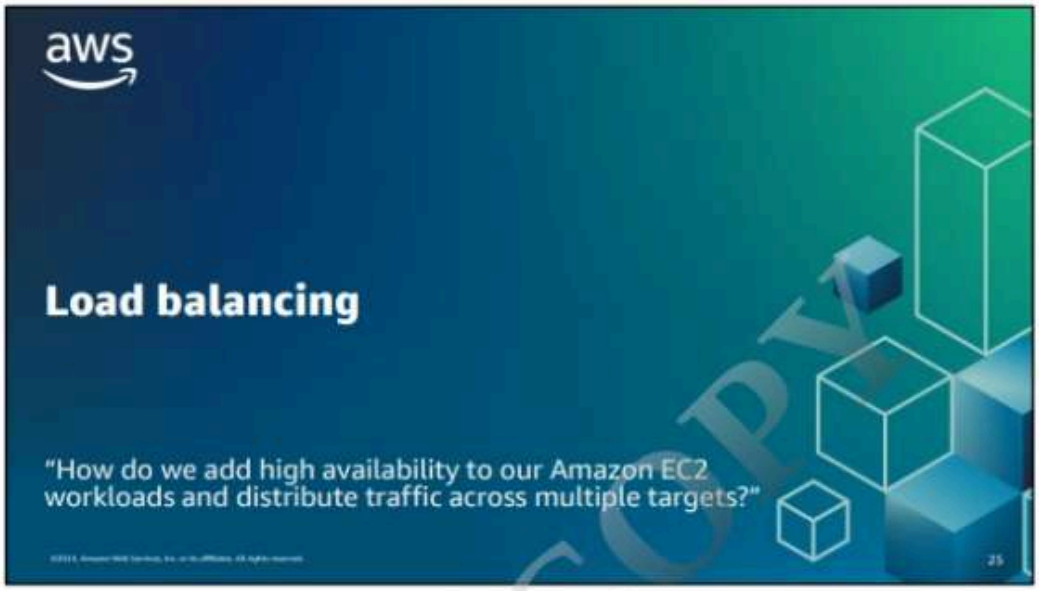
- What other actions could you invoke by using EventBridge that could automate a response to high CPU utilization on the EC2 instance?
- Are tools available in AWS services that can help you redirect traffic while you take action?
- How could you scale out to prevent an event that is caused by overutilized CPU?

In this course, you learn how to build architectures that support your operations team to maintain a healthy environment.

For more information, see Amazon EventBridge integrations at https://aws.amazon.com/eventbridge/integrations/.

DO NOT COPY javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

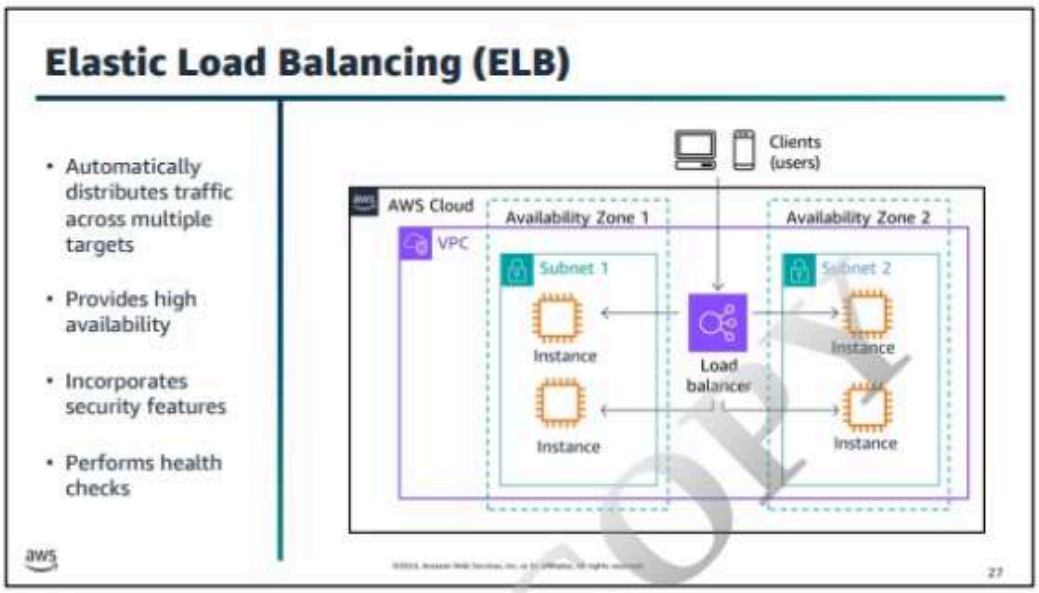


The operations manager asks, "How do we add high availability to our Amazon EC2 workloads and distribute traffic across multiple targets?" The team needs to examine the AWS services that are used in event-driven infrastructure, or applications, to monitor and manage events.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Elastic Load Balancing (ELB) makes up one of the most widely used AWS service categories. It has been adopted by organizations of all sizes, in all geographies, and across every industry. ELB load balancers are the only load balancers available on AWS that natively connect users to your EC2 instances, container deployments, and AWS Lambda functions.

Some key feature sets of ELB include the following:

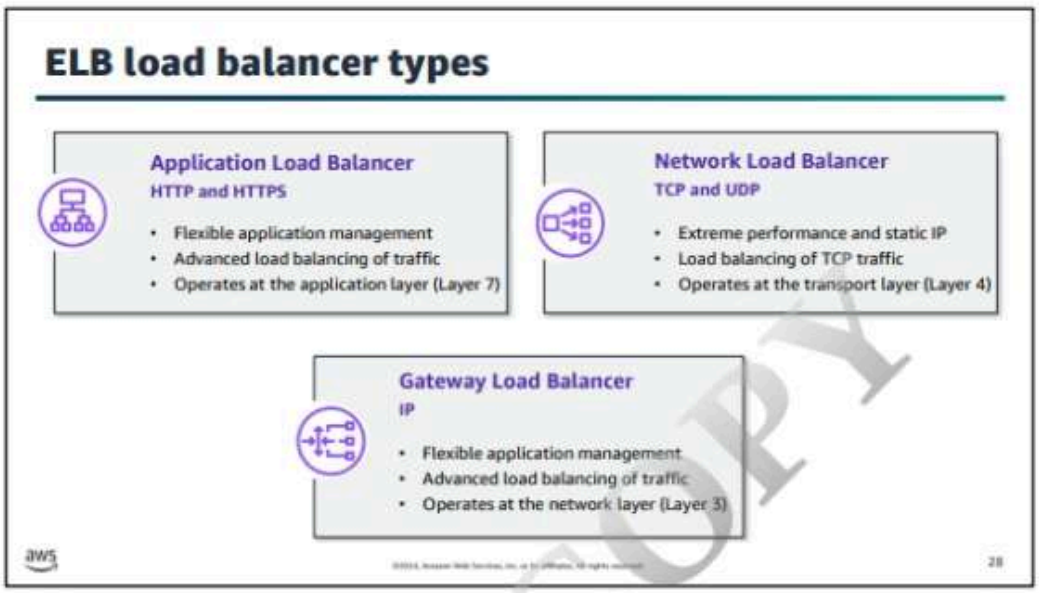
- **High availability** – ELB automatically distributes your traffic across multiple targets in a single Availability Zone or multiple Availability Zones. Examples of targets include EC2 instances, containers, and IP addresses.
- **Layer 4 or Layer 7 HTTP and HTTPS load balancing** – You can load balance your HTTP or HTTPS applications for Layer 7-specific features. Alternatively, you can use strict Layer 4 load balancing for applications that rely purely on the TCP.
- **Security features** – Use Amazon Virtual Private Cloud (Amazon VPC) to create and manage security groups that are associated with load balancers to provide additional networking and security options. You can also create an internal (non-internet-facing) load balancer.
- **Health checks** – ELB load balancers can detect unhealthy targets, stop sending traffic to them, and spread the load across the remaining healthy targets.
- **Monitoring operations** – To monitor real-time application performance, ELB integrates with CloudWatch metrics and provides request tracing.

In this example, the load balancer receives requests from desktop and mobile clients (users). There are two subnets in the same VPC, with two EC2 instances each. Each subnet is in a separate Availability Zone. All four EC2 instances are registered to the same load balancer and receive traffic.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



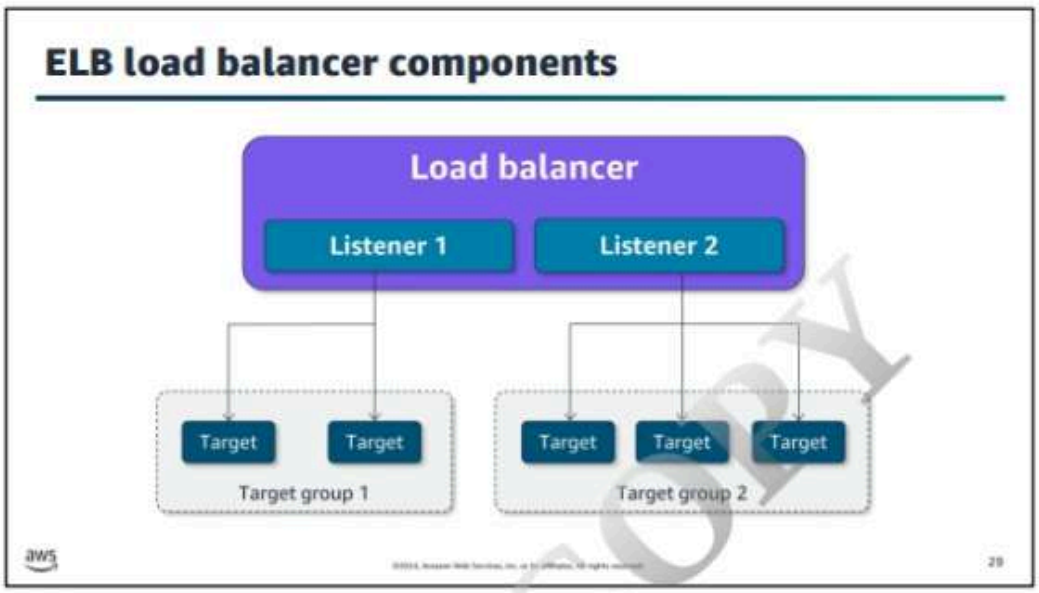
Types of load balancers available for you to use include the following:

- **Application Load Balancer** – This load balancer functions at the application layer (Layer 7) of the Open Systems Interconnection (OSI) model. Application Load Balancer supports content-based routing, applications that run in containers, and open standard protocols (WebSocket and HTTP/2). This type of balancer is ideal for advanced load balancing of HTTP and HTTPS traffic.
- **Network Load Balancer** – This load balancer is designed to handle tens of millions of requests per second while maintaining high throughput at ultra-low latency. Network Load Balancer operates at the transport layer (Layer 4), and routes connections to targets based on IP protocol data. Targets include EC2 instances, containers, and IP addresses. It is ideal for balancing TCP and UDP traffic.
- **Gateway Load Balancer** – You can use this load balancer to deploy, scale, and manage your third-party virtual appliances. It provides one gateway for distributing traffic across multiple virtual appliances, and scales them up or down based on demand. This distribution reduces potential points of failure in your network and increases availability. Gateway Load Balancer passes all Layer 3 traffic transparently through third-party virtual appliances. It is invisible to the source and destination.
- **Classic Load Balancer** – This previous-generation load balancer is not discussed deeply in this course.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



An ELB load balancer distributes your incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones. In this way, it increases the availability of your application. Your load balancers are configured with *listeners*. You can have more than one listener per load balancer. Listeners have different functions for different types of load balancers. Your load balancer can forward requests to one or more target groups based on rules and settings that you define.

Each *target group* routes requests to one or more registered targets (for example, EC2 instances) by using the specified protocol and port number. You can register a target with multiple target groups.

In the example, a load balancer has two listeners. Listener 1 has one target group called *target group 1*, with two registered targets. A second listener in the same load balancer has one target group with three different registered targets.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

ELB common features

Feature	Application Load Balancer	Network Load Balancer	Gateway Load Balancer
Health checks	Yes	Yes	Yes
CloudWatch metrics	Yes	Yes	Yes
Logging	Yes	Yes	Yes
SSL offloading	Yes	Yes	
Connection draining	Yes	Yes	Yes
Preserve source IP address	Yes	Yes	Yes
Static IP address	**	Yes	
AWS Lambda functions as a target	Yes		
Redirects	Yes		
Fixed-response actions	Yes		

10

The table in the graphic outlines support for a set of load balancer features.

If you need flexible application management, consider an Application Load Balancer. If you need extreme performance and a static IP, consider a Network Load Balancer. If you need to manage your third-party virtual appliances, consider a Gateway Load Balancer.

**Elastic Load Balancing now supports forwarding traffic directly from Network Load Balancer to Application Load Balancer. With this feature, you can use AWS PrivateLink and expose static IP addresses for applications that are built on Application Load Balancer.

For more information and a full, up-to-date comparison of ELB products, see Elastic Load Balancing features at https://aws.amazon.com/elasticloadbalancing/features/?nc=sn&loc=2.

Learn more about exposing static IP addresses for applications that are built on Application Load Balancer. For more information, see "Application Load Balancer now enables AWS PrivateLink and static IP addresses by direct integration with Network Load Balancer" at https://aws.amazon.com/about-aws/whats-new/2021/09/application-load-balancer-aws-privatelink-static-ip-addresses-network-load-balancer/.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

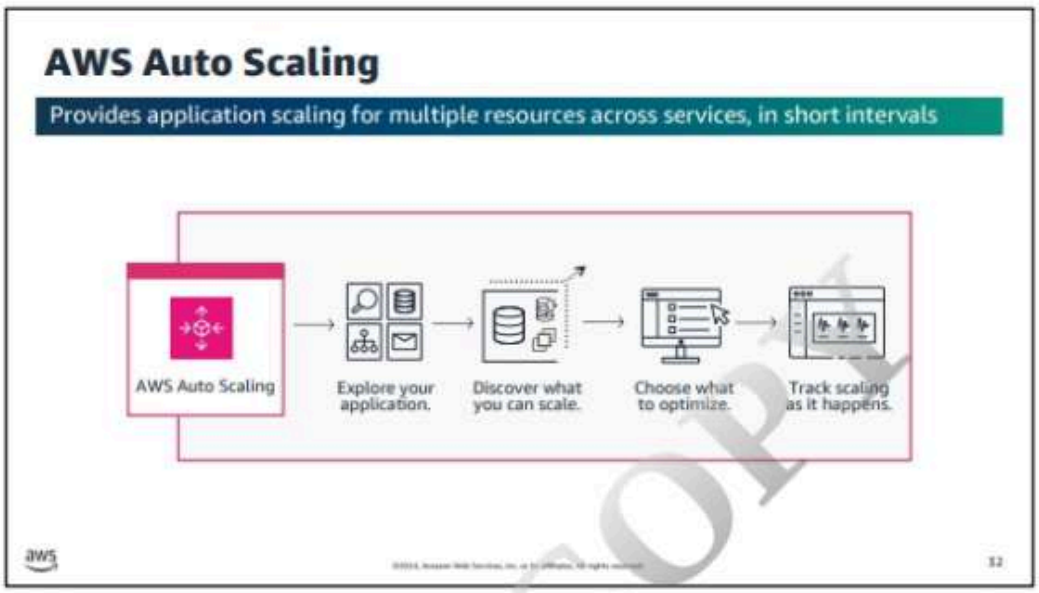
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

The operations manager asks, "How can we dynamically increase and decrease capacity to meet changing demand?" Your operations team needs to compare auto scaling features to determine best practices.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, you can set up application scaling for multiple resources across multiple services in minutes.

The service provides a powerful user interface that you can use to build scaling plans for resources. For example, these resources might include EC2 instances, Spot Fleets, and other compute and database services. AWS Auto Scaling helps you with scaling by providing recommendations for you to optimize performance and costs, or to balance between them.

AWS Auto Scaling provides application scaling for multiple resources, such as Amazon EC2, Amazon DynamoDB, Amazon Aurora, and many more, across multiple services in short intervals.

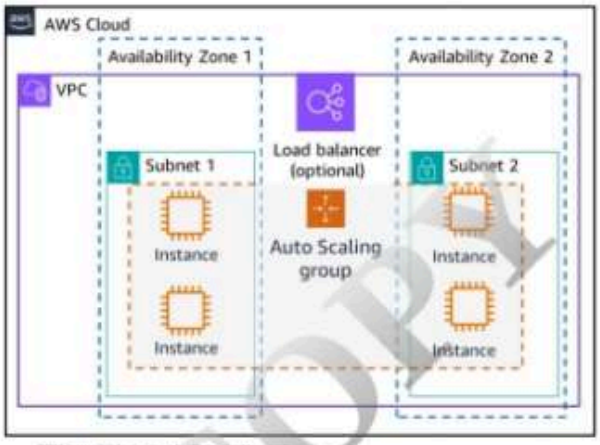
DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Amazon EC2 Auto Scaling

- Helps you control EC2 instances available to handle the load for your application
- Launches or terminates your AWS resources based on specified conditions
- Registers new instances with load balancers, when specified



With Amazon EC2 Auto Scaling, you can build scaling plans that automate how groups of different EC2 resources respond to changes in demand. You can optimize availability, costs, or a balance of both.

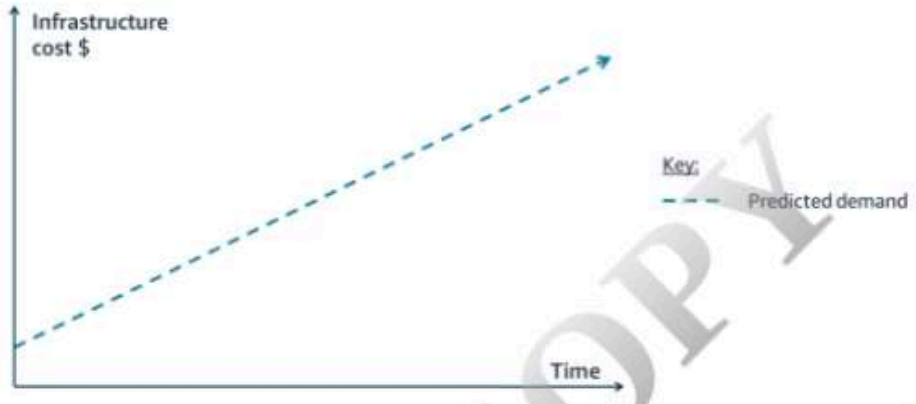
If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases. Amazon EC2 Auto Scaling integrates with ELB so that you can attach one or more load balancers to an existing Amazon EC2 Auto Scaling group. After you attach the load balancer, it automatically registers the instances in the group and distributes incoming traffic across the instances.

In the example, one VPC has two subnets in two separate Availability Zones. Two EC2 instances are launched in each subnet as part of one single Auto Scaling group. An existing load balancer is shown separately, but registering the Auto Scaling group is optional.

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Elasticity: Scaling in and out (1 of 6)



Cloud computing addresses the problem of overutilization and underutilization of resources through the concept of scaling. What does this concept mean? As your business grows and demand for your applications increases, your infrastructure costs will also increase over time. An elastic infrastructure can intelligently expand and contract as its capacity needs change.

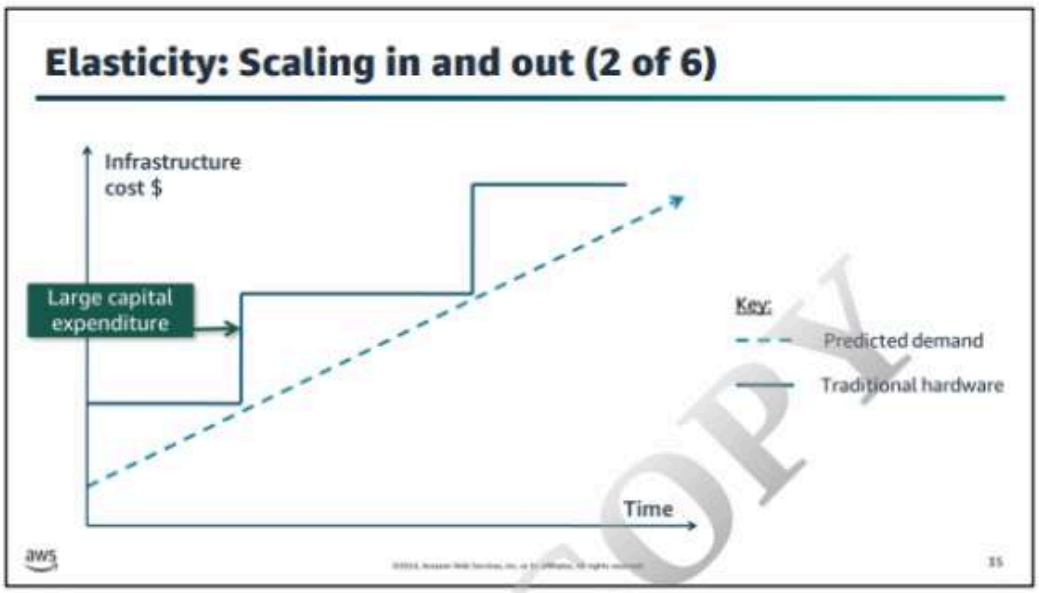
Examples of scaling include the following:

- Increase the number of web servers when traffic spikes.
- Lower write capacity on your database when that traffic goes down.
- Handle the day-to-day fluctuation of demand throughout your architecture.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

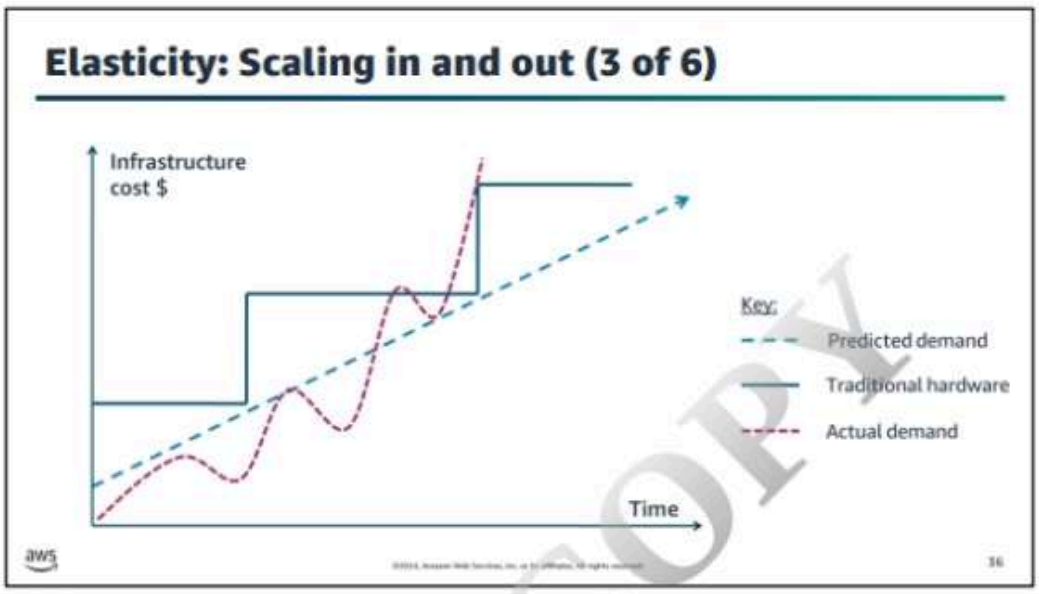
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



In a traditional IT procurement model, to cope with this increase in demand, businesses must regularly invest up front in large purchases of infrastructure. They would continue to use this infrastructure until they needed more capacity, and then make another large investment.

DO NOT COPY
javierartiga.espublico@gmail.com

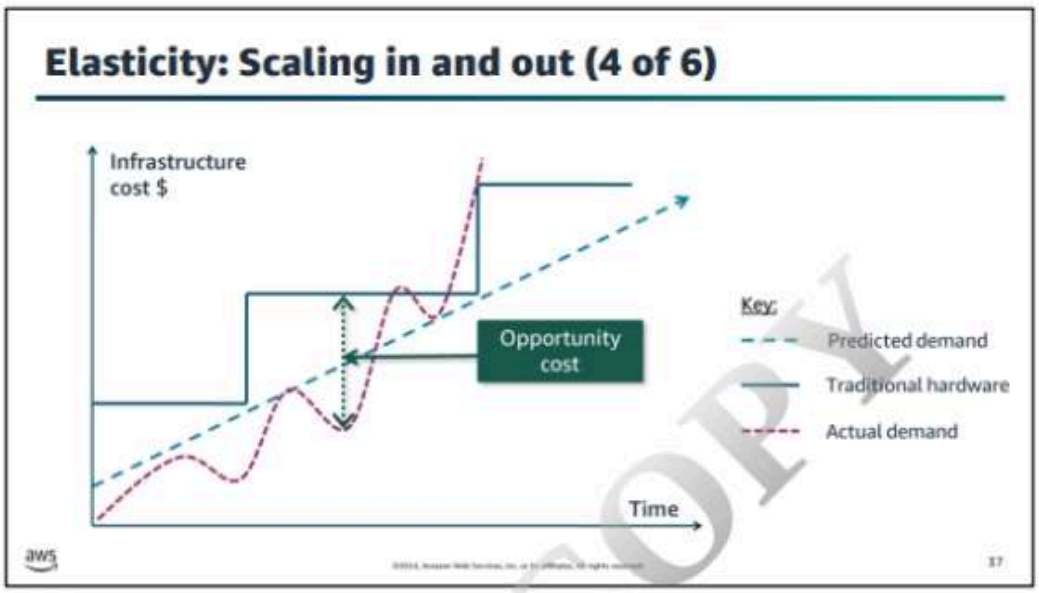
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



However, demand is rarely linear or predictable. It is far more likely that workloads are spiky and variable.

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

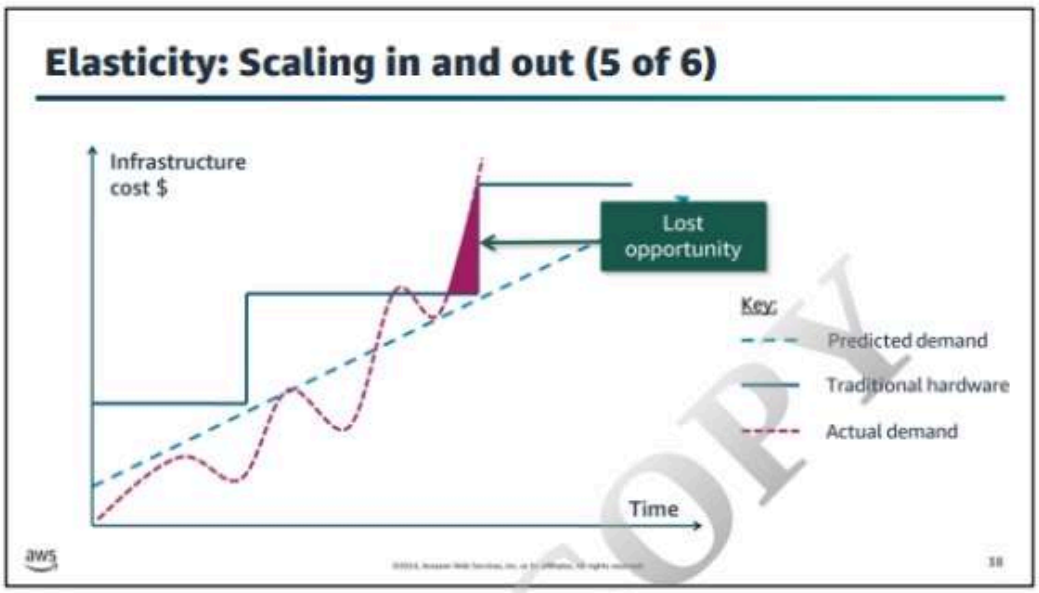


A big opportunity cost exists for businesses that have invested in high levels of infrastructure that aren't used when demand is low. Consider how businesses could have reallocated these upfront costs if they had known these demand patterns.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

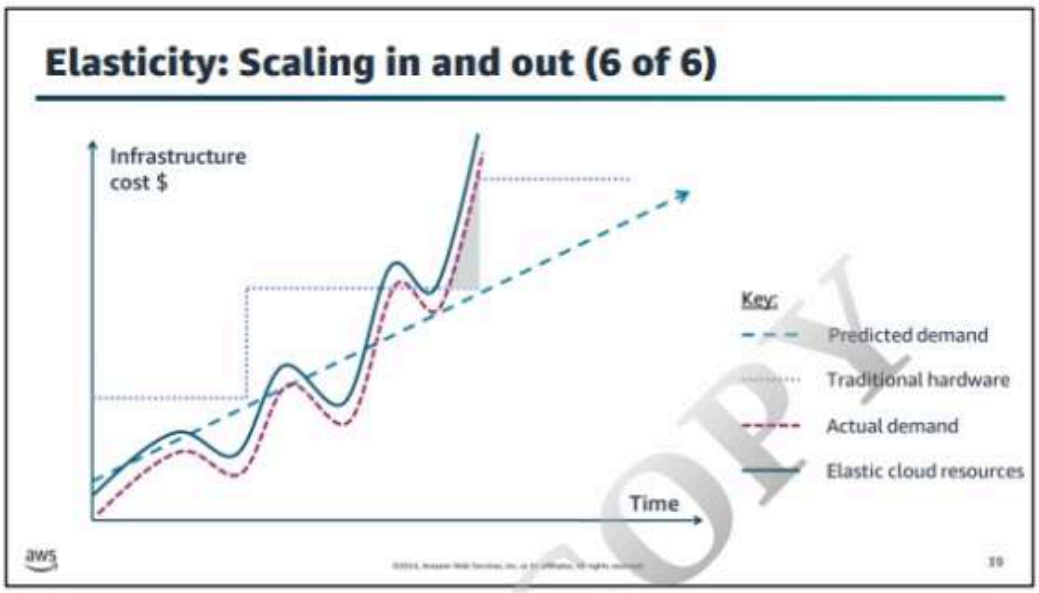


The alternative is that a business underprovisions its infrastructure to save money or because it couldn't have predicted a peak. It experiences a lost opportunity when demand outstrips the available environment. This situation can create a negative user experience. Consider the feeling of a website that crashes when you are trying to purchase newly released tickets to a popular show.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

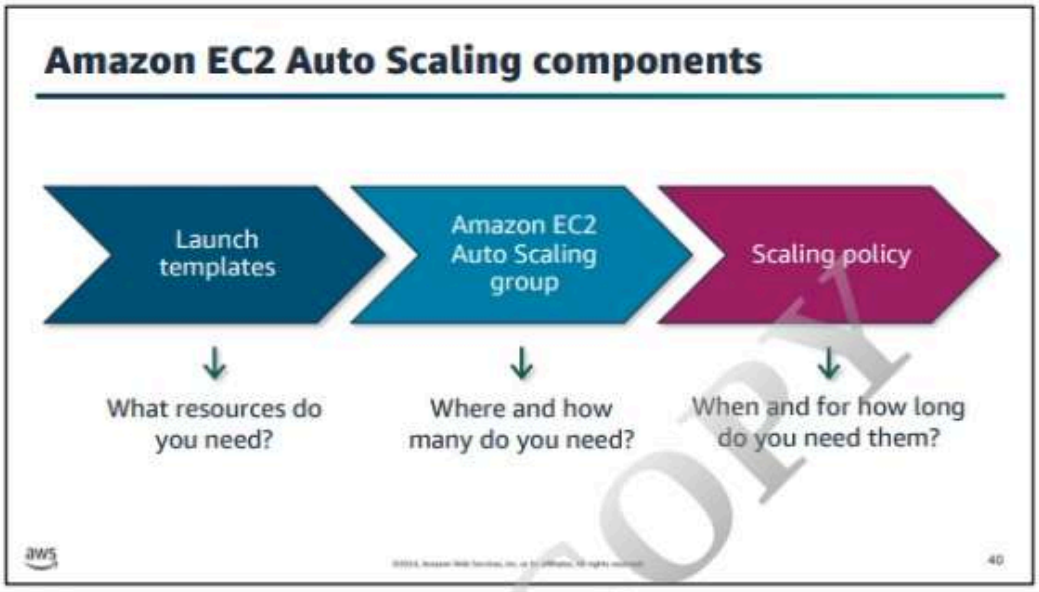


Using cloud technology, you can fit supply to demand through elastic cloud resources. Unlike traditional infrastructure, you don't need to provision resources months in advance. You don't need to keep resources that are not used, and you don't need to worry as much about an inability to forecast customer demand.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

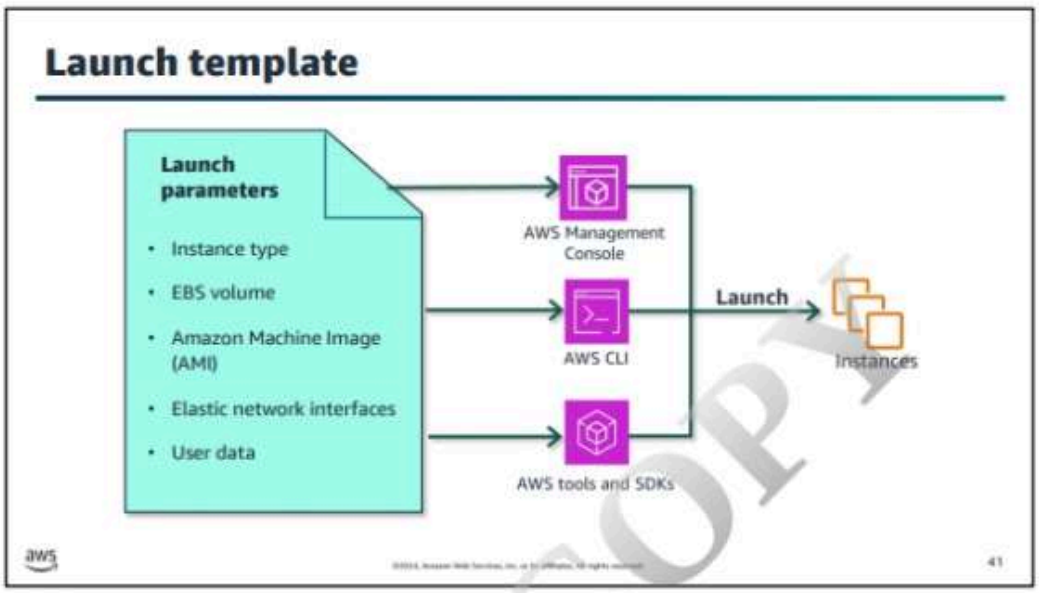


Amazon EC2 Auto Scaling helps you to have the correct number of Amazon EC2 instances available to handle application load. You create collections of EC2 instances, called *Auto Scaling groups*. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling manages your group to never go below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling manages your group to never go above this size.

The graphic shows how to set up a group from start to finish. You learn about each of the three steps in the following slides.

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Before you can create an Auto Scaling group using a *launch template*, you must create a launch template. This template includes the parameters required to launch an EC2 instance, such as the ID of the Amazon Machine Image (AMI) and an instance type. A launch template provides full functionality for Amazon EC2 Auto Scaling, and it also provides newer features of Amazon EC2. These features can include the current generation of Amazon EBS Provisioned IOPS volumes (io2), EBS volume tagging, T2 Unlimited instances, Elastic Inference, and Dedicated Hosts.

You are strongly encouraged to create Auto Scaling groups from launch templates to get the latest features from Amazon EC2. A *launch configuration* is an instance configuration template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you must specify information about the EC2 instances to launch. Include the AMI, instance type, key pair, security groups, and block device mapping.

Alternatively, you can create a launch configuration by using attributes from a running EC2 instance.

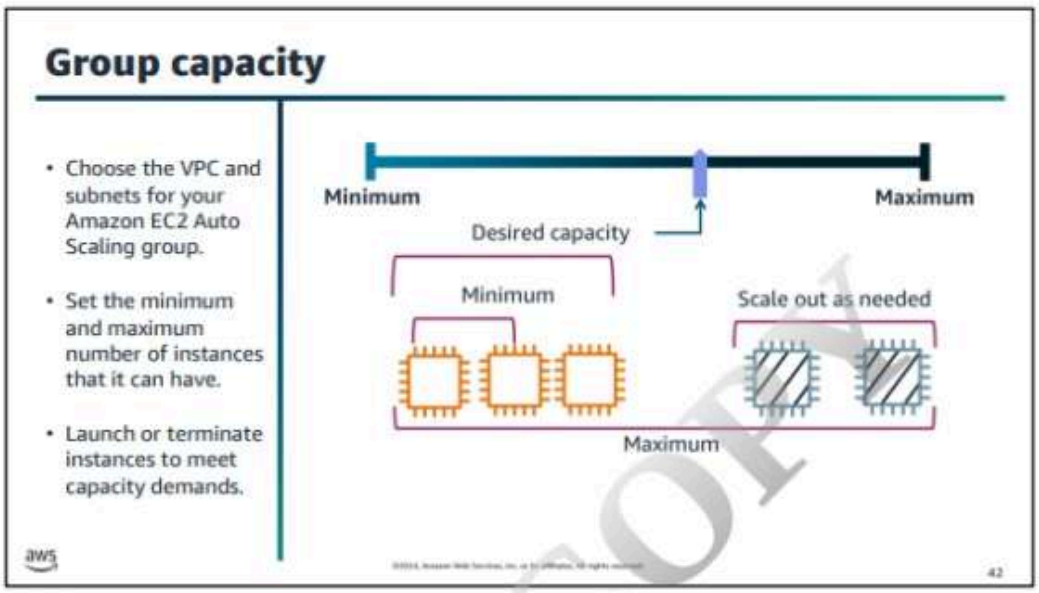
For more information about templates, see "Create a launch template for an Auto Scaling group" in the *Amazon EC2 Auto Scaling User Guide* at https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-launch-template.html.

For more information about launch configurations, see "Create a launch configuration" in the *Amazon EC2 Auto Scaling User Guide* at https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-lc-with-instanceID.html.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



A group contains a collection of EC2 instances that are treated as a logical grouping for automatic scaling and management. With a group, you can also use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies. You can specify the minimum number of instances in each group, and Amazon EC2 Auto Scaling controls the group to never go below this size.

You can specify the desired capacity, either when you create the group or at any time afterward. Then, Amazon EC2 Auto Scaling manages your group to have this number of instances. If you specify scaling policies, Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

For example, this group has a minimum size of two instances, a desired capacity of three instances, and a maximum size of five instances. The scaling policies that you define adjust the number of instances, within your minimum and maximum number, based on the criteria that you specify.

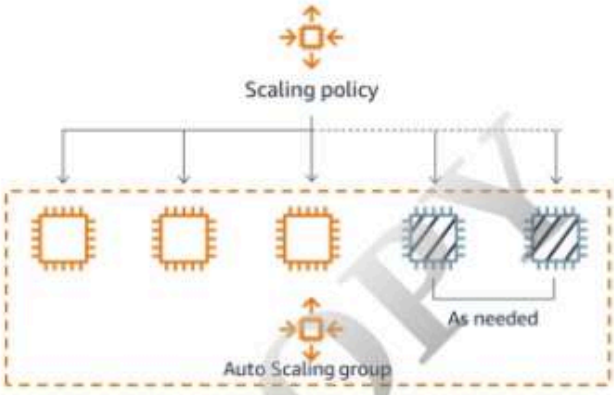
For more information, see "Auto Scaling groups" in the *Amazon EC2 Scaling User Guide* at https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html.

DO NOT COPY javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Scaling methods

- Fixed
- Manually
- Scheduled
- Dynamically
- Proactively



41

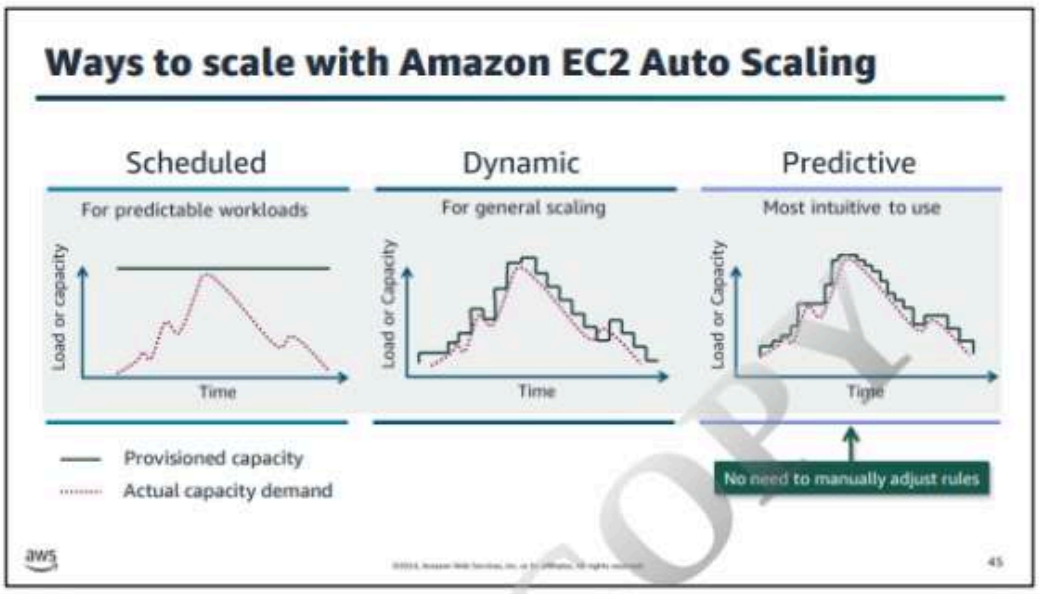
You can use the following tools to invoke scaling in your groups:

- **Maintain a fixed number of instances** – You can configure your group to maintain a specified number of running instances at all times. If an instance becomes unhealthy, the group terminates the unhealthy instance and launches another instance to replace it.
- **Scale dynamically based on demand** – A scaling policy instructs Amazon EC2 Auto Scaling to track a specific CloudWatch metric. It defines which action to take when the associated CloudWatch alarm is in the ALARM state.
- **Scale based on a schedule** – You can scale by schedule. Actions are performed automatically as a function of time and date. Scaling by schedule is useful when you know exactly when to increase or decrease the number of instances in your group.
- **Scale manually** – Manual scaling is the most basic way to scale your resources. You specify only the change in the maximum, minimum, or desired capacity of your group. Amazon EC2 Auto Scaling manages the process of creating or terminating instances to maintain the updated capacity.
- **Scale proactively** – You can also combine predictive scaling and dynamic scaling (proactive and reactive approaches, respectively) to scale your EC2 capacity faster. Use predictive scaling to increase the number of EC2 instances in your Auto Scaling group in advance of daily and weekly patterns in traffic flows.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



With *scheduled scaling*, you can scale your application before known load changes. For example, every week, the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday. You can plan your scaling activities based on the known traffic patterns of your web application.

With *dynamic scaling*, you define how to scale the capacity of your Amazon EC2 Auto Scaling group in response to changing demand. For example, suppose that you have a web application that currently runs on two instances. You want the CPU utilization of the group to stay at about 50 percent when the load on the application changes. As a result, you have extra capacity to handle traffic spikes without maintaining an excessive number of idle resources.

Use *predictive scaling* to increase the number of EC2 instances in your group in advance of daily and weekly patterns in traffic flows. Predictive scaling is well suited for situations where you have the following:

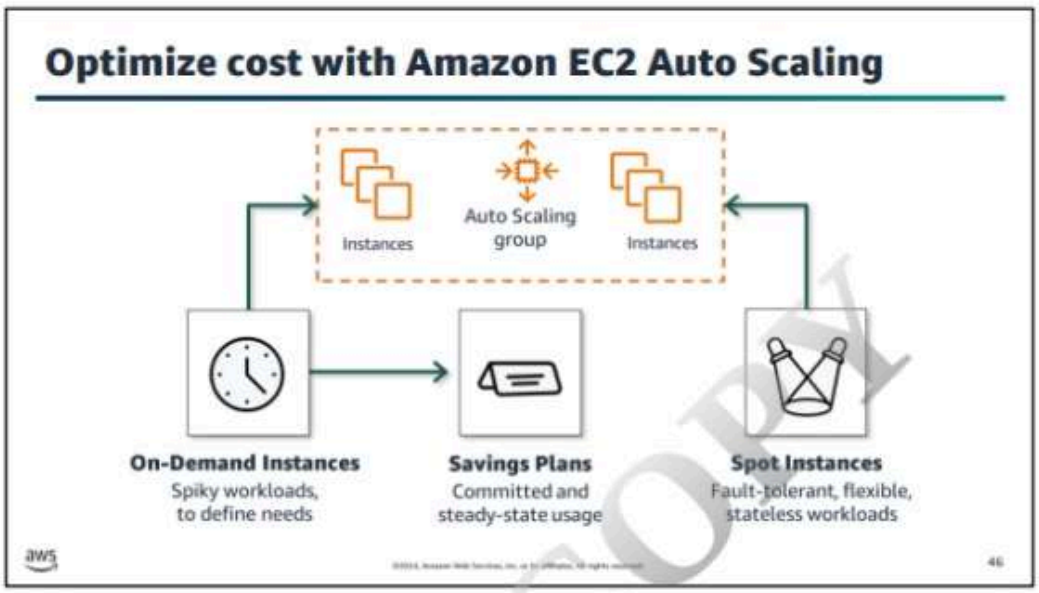
- Cyclical traffic, such as high use of resources during regular business hours and low use of resources during evenings and weekends
- Recurring on-and-off workload patterns, such as batch processing, testing, or periodic data analysis
- Applications that take a long time to initialize, which causes a noticeable latency impact on application performance during scale-out events

It is recommended that you scale out early and fast, while you scale in slowly over time.

NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Amazon EC2 Auto Scaling supports multiple purchasing options within the same group. You can launch and automatically scale a fleet of On-Demand Instances and Spot Instances within a single Auto Scaling group. You can receive discounts for using Spot Instances. In addition, you can use Reserved Instances or a Savings Plan to receive discounted rates of the regular On-Demand Instance pricing. All of these factors combined help you to optimize your cost savings for EC2 instances, while making sure that you obtain the desired scale and performance for your application.

Using *Amazon EC2 Fleet*, you can define a combination of EC2 instance types to make up the desired capacity of your group. This capacity is defined as a percentage of each type of purchasing option. Amazon EC2 Auto Scaling will maintain the desired cost optimization as your group scales in or out. Groups that are made up of mixed fleets still support the same lifecycle hooks, instance health checks, and scheduled scaling as a single-fleet group.

To learn more about optimizing cost, see "Auto Scaling groups with multiple instance types and purchase options" in the *Amazon EC2 Auto Scaling User Guide* at https://docs.aws.amazon.com/autoscaling/ec2/userguide/asg-purchase-options.html.

DO NOT COPY
javierartiga.espublico@gmail.com

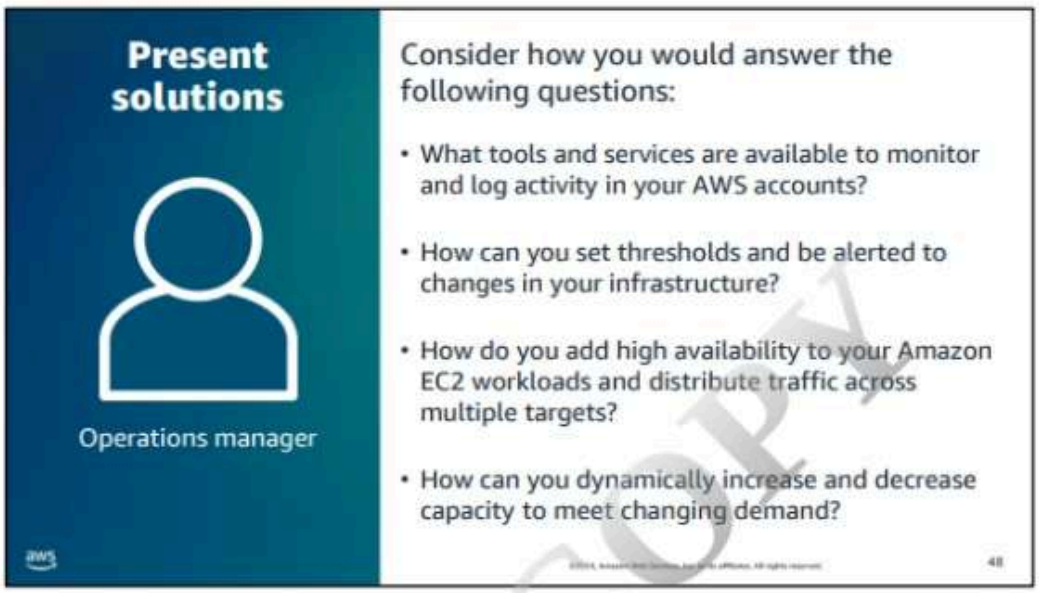
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Imagine that you are now ready to talk to the operations manager and present solutions that meet their architectural needs. Think about how you would answer the questions from the beginning of the lesson.

Your answers should include the following solutions:

- Use tools such as CloudWatch metrics, CloudWatch Logs, CloudTrail, and VPC Flow Logs to monitor and log activity in your accounts.
- Configure CloudWatch alarms for existing metrics and use EventBridge to take action when a metric is in the ALARM state.
- Explore options available to you in Elastic Load Balancing. Choose either an Application Load Balancer, a Network Load Balancer, or a Gateway Load Balancer based on your use case.
- Prepare for changes in demand by using AWS Auto Scaling. For compute, use Amazon EC2 Auto Scaling to invoke scaling based on the expected demand for that workload. Use launch templates to build your configuration to support scaling in and out.

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Module review

In this module, you learned about the following:

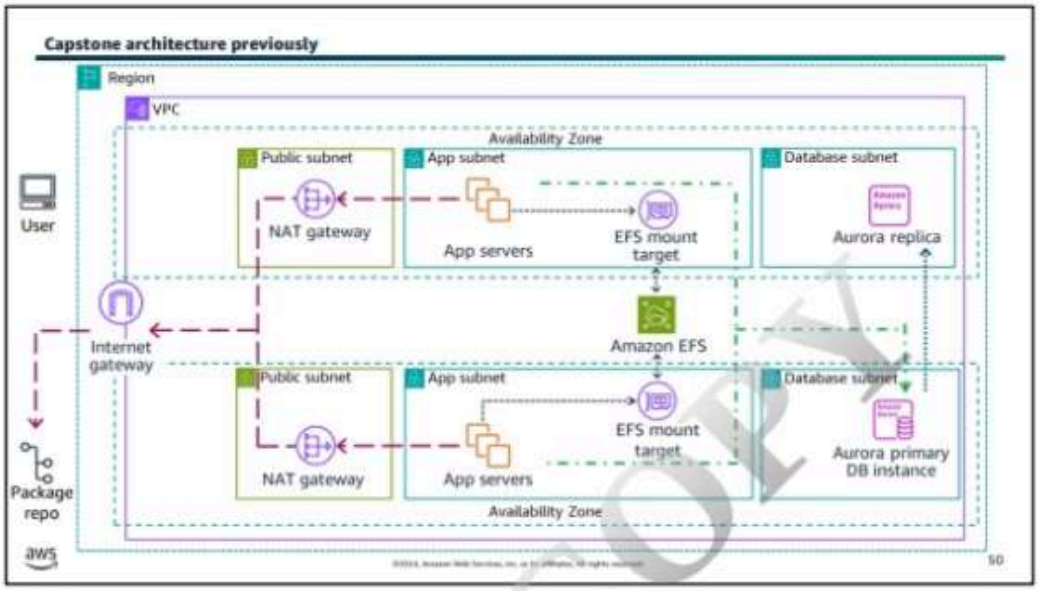
- ✓ Monitoring
- ✓ Alarms and events
- ✓ Load balancing
- ✓ Auto scaling

Next, you review the following:

- Capstone check-in
- Lab introduction
- Knowledge check

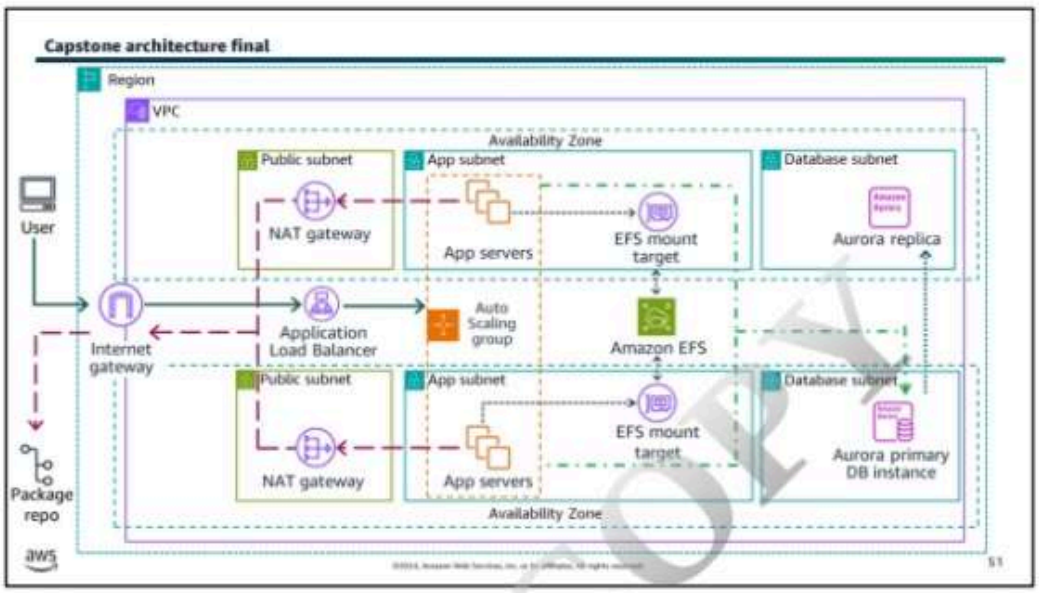
DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



You notice that an ELB load balancer is added to balance the WordPress traffic across multiple EC2 instances. This arrangement also gives you fault tolerance if one of your EC2 instances fails a health check for application traffic.

Your application servers that were added in the Compute module have been placed in an Amazon EC2 Auto Scaling group. This provision adds scalability to your application.

As your Auto Scaling group scales out, the launch template will use your AMI to launch application servers. User data in the launch template is processed during instance launch. You can use it to mount your Amazon Elastic File System (Amazon EFS) automatically at instance launch. The EC2 instances in the Auto Scaling group will launch with security groups that allow incoming traffic from the Amazon Aurora primary DB instance.

You have now learned about all of the services used in the capstone lab. You will revisit this architecture again on Day 3 to strengthen your knowledge on these topics.

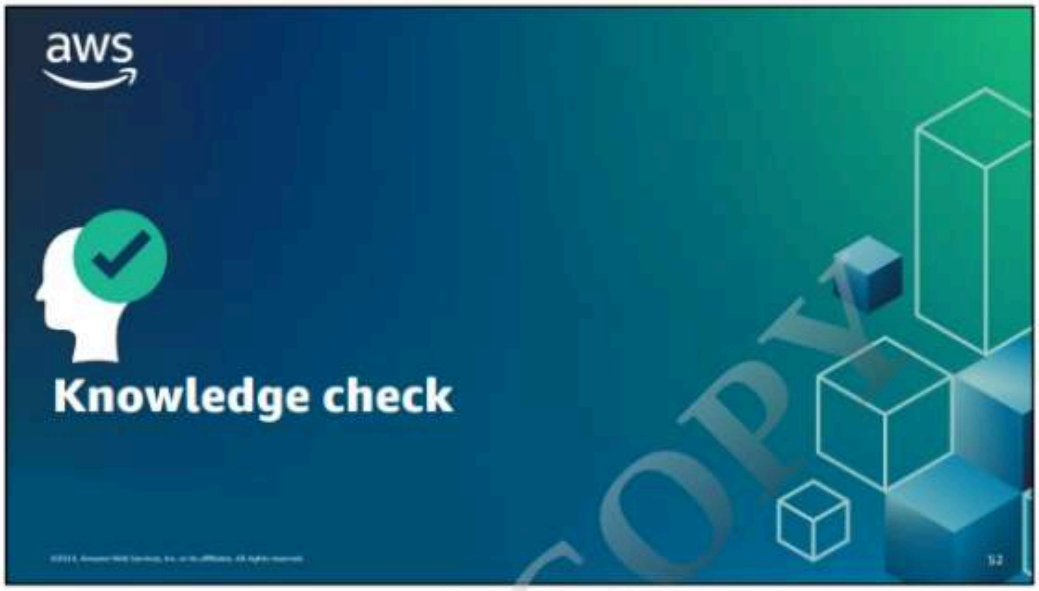
Consider how you could continue expanding on this architecture by reviewing the following questions:

- How would you automate the creation of resources?
- How would you duplicate this deployment in more than one account without manual effort?
- Can you use containers instead of virtual machines to control the application?
- Can you integrate this application with your on-premises network?
- Do you need more than one account to be able to access this application?
- What serverless strategies can you use to help with easy management of the application?
- What is your disaster recovery strategy for this environment?

DO NOT COPY javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 1

Which option is a valid target for an Application Load Balancer?

Choice	Response
A	An Amazon EC2 instance
B	An Availability Zone
C	An Amazon S3 bucket
D	A virtual private network (VPN) connection

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 1 and answer

Which option is a valid target for an Application Load Balancer?

Choice	Response
A correct	An Amazon EC2 instance
B	An Availability Zone
C	An Amazon S3 bucket
D	A virtual private network (VPN) connection

The correct answer is A, Amazon EC2 instance.

A load balancer distributes incoming application traffic across multiple targets, such as EC2 instances, in multiple Availability Zones to increase the availability of your application.

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 2

An application support team has an application with unpredictable traffic patterns that runs on at least two instances. They want the CPU utilization to stay at about 75 percent. Which Amazon EC2 Auto Scaling strategy should they choose?

Choice	Response
A	Scheduled
B	Dynamic
C	Predictive
D	Manual

55

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 2 and answer

An application support team has an application with unpredictable traffic patterns that runs on at least two instances. They want the CPU utilization to stay at about 75 percent. Which Amazon EC2 Auto Scaling strategy should they choose?

Choice	Response
A	Scheduled
B correct	Dynamic
C	Predictive
D	Manual

The answer is B, dynamic.

This option gives you extra capacity to handle traffic spikes without maintaining an excessive number of idle resources.

Predictive scaling would be best for cyclical traffic, recurring on-and-off workload patterns, or applications that take a long time to initialize.

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 3

Which service can invoke actions based on data from account resources and supported third-party management services?

Choice	Response
A	Amazon CloudWatch Logs
B	Amazon EventBridge
C	AWS CloudTrail
D	Amazon EC2 Auto Scaling

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 3 and answer

Which service can invoke actions based on data from account resources and supported third-party management services?

Choice	Response
A	Amazon CloudWatch Logs
B correct	Amazon EventBridge
C	AWS CloudTrail
D	Amazon EC2 Auto Scaling

The correct answer is B, Amazon EventBridge.

EventBridge builds upon and extends CloudWatch Events. It uses the same service API and endpoint, and the same underlying service infrastructure. EventBridge reacts to events that other AWS services generate.

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 4

Which options are valid alarm states in Amazon CloudWatch? (Select TWO.)

Choice	Response
A	READY
B	ALERT
C	ALARM
D	INSUFFICIENT_DATA
E	FAILED

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 4 and answer

Which options are valid alarm states in Amazon CloudWatch? (Select TWO.)

Choice	Response
A	READY
B	ALERT
C correct	ALARM
D correct	INSUFFICIENT_DATA
E	FAILED

The correct answers are C and D, ALARM and INSUFFICIENT_DATA.

Amazon CloudWatch alarms have three possible states: OK, ALARM, and INSUFFICIENT_DATA.

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 5

Which options are use cases for AWS CloudTrail data? (Select TWO.)

Choice	Response
A	Provide real-time observability of AWS resources.
B	Store log data as a record of account usage.
C	Log events for a particular service or application.
D	Capture root login failures.
E	Collect metric data that measures CPU utilization.

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Knowledge check question 5 and answer

Which options are use cases for AWS CloudTrail data? (Select TWO.)

Choice	Response
A	Provide real-time observability of AWS resources.
B correct	Store log data as a record of account usage.
C	Log events for a particular service or application.
D correct	Capture root login failures.
E	Collect metric data that measures CPU utilization.

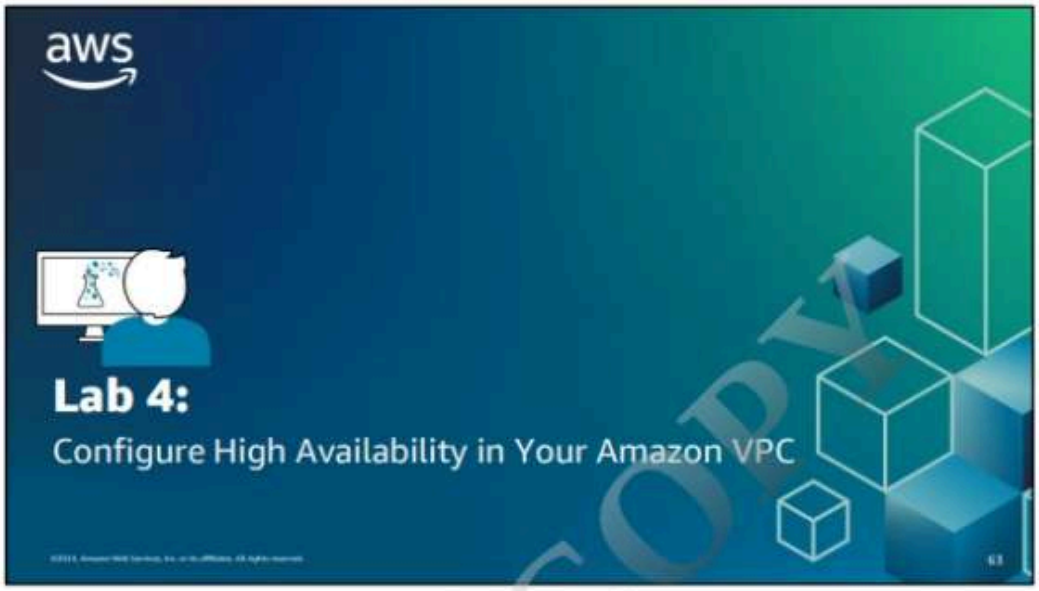
The correct answers are B and D, store log data as a record of account usage and capture root login failures.

CloudTrail provides insight into who is doing what and when, by tracking user activity and API usage. CloudTrail does the following tasks:

- Monitors and logs account activity across your AWS infrastructure
- Records API call interactions for most AWS services
- Automatically pushes logs to Amazon S3
- Captures root login failures

DO NOT COPY javierartiga.espublico@gmail.com

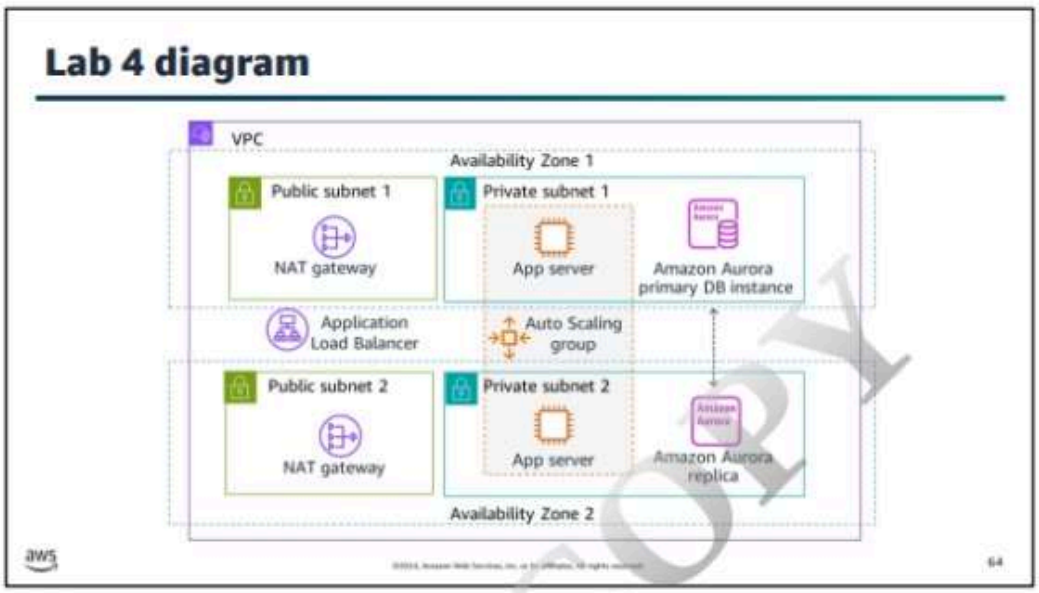
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



Lab 4 uses a VPC across two Availability Zones with a public and private subnet in each. Each private subnet contains an app server in the same Auto Scaling group. Private subnet 1 has an Aurora primary DB instance, and private subnet 2 has an Aurora replica. The primary DB instance and replica communicate with each other across Availability Zones. Each public subnet has a NAT gateway. An Application Load Balancer is in the VPC.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Lab tasks

Task 1: Inspect your existing lab environment.

Task 2: Create a launch template.

Task 3: Create an Auto Scaling group.

Task 4: Test your application.

Task 5: Test high availability of your application tier.

Task 6: Configure high availability of the database tier.

Task 7: Make the NAT gateway highly available.

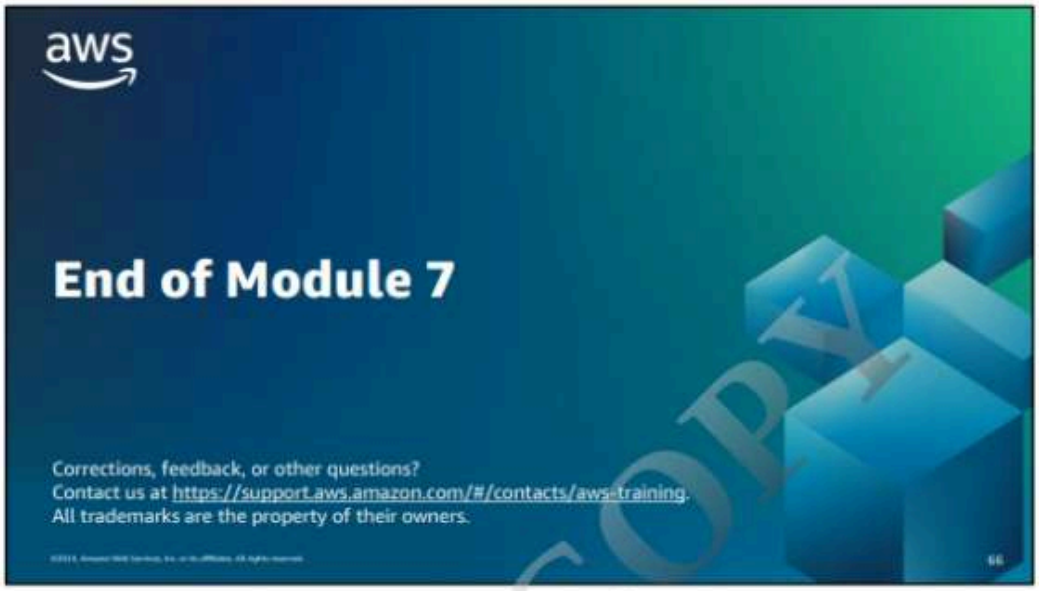
Task 8: Test the high availability of the Aurora database.

65

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

**Accessibility note for screen reader users: To make code examples in this presentation more accessible, Start of code and End of code labels have been added before and after blocks of code on slides in this presentation.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Poll question

Have you used any of these resource provisioning tools?

A. AWS CloudFormation

B. AWS Elastic Beanstalk

C. Others (for example, Terraform or OpenStack Heat)

D. Not yet

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Module overview

- Business request
- AWS CloudFormation
- Infrastructure management
- Amazon Q
- Present solutions
- Knowledge check

DO NOT COPY
javierartiga.espublico@gmail.com

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Imagine that your chief technology officer meets with you to discuss how to automate deployments and operations in the cloud. Here are some questions that they are asking.

At the end of this module, you meet with the chief technology officer and present some solutions.

DO NOT COPY
javierartiga.espublico@gmail.com

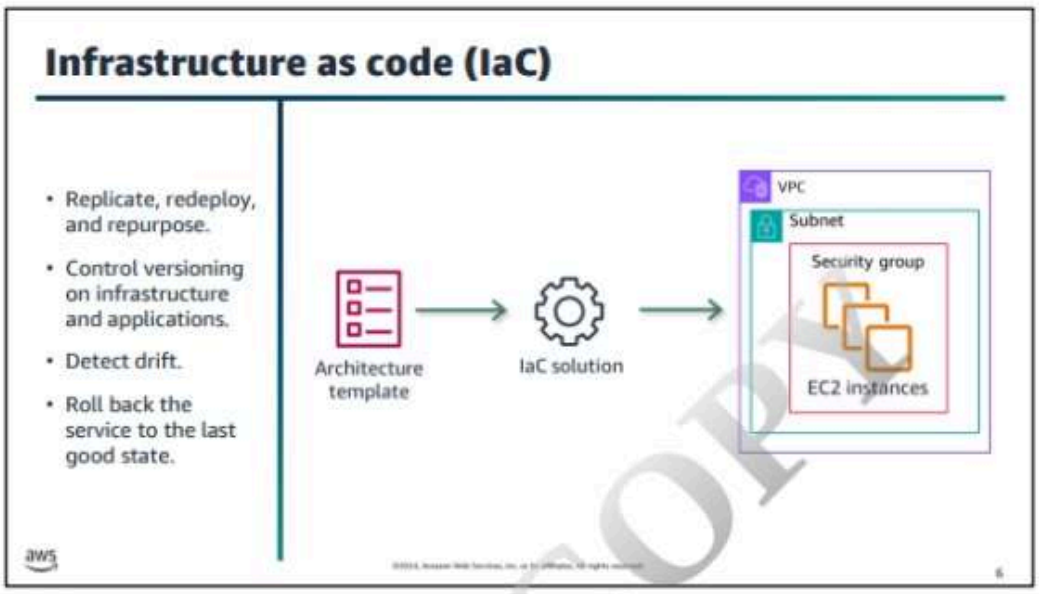
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

The chief technology officer asks, "How can we simplify our cloud infrastructure build?" The infrastructure team would like to create environments that they can deploy, update, and take down in an uncomplicated way. The company wants you to identify tools that automate infrastructure deployment.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



You can simplify the deployment of your Amazon Web Services (AWS) resources by using infrastructure as code (IaC). With IaC, you use code to define, deploy, configure, update, and remove infrastructure.

A *template* is a text file that describes and defines the resources to be deployed in your environment. This template is then processed by an engine that provisions the specified resources. With a template, you can do the following:

- Define an entire application stack (all resources required for your application) in a JSON or YAML template file. Treat the template as code and manage it using a version control system.
- Define runtime parameters for a template, such as the Amazon Elastic Compute Cloud (Amazon EC2) instance size, and Amazon EC2 key pair.

The IaC solution provisions the resources that are defined in the template. IaC has the following benefits:

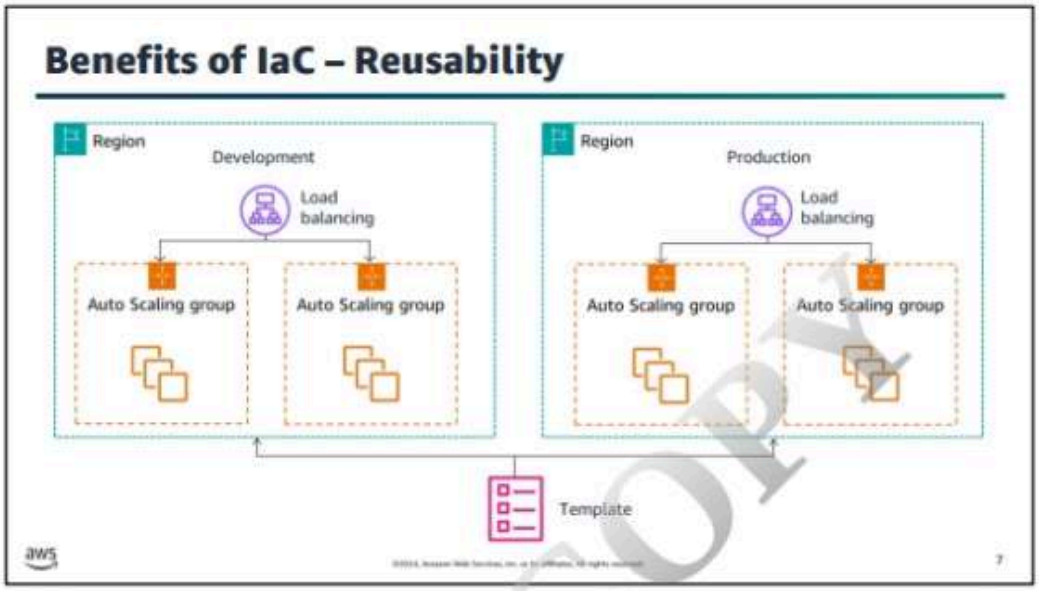
- **Speed and safety** – Your infrastructure is built programmatically, which makes it faster than manual deployment and makes errors less likely.
- **Reusability** – You can organize your infrastructure into reusable modules.
- **Documentation and version control** – Your templates document your deployed resources, and version control provides a history of your infrastructure over time. You can also roll back to a previous working version of your infrastructure in the event of error.
- **Validation** – You perform code review on your templates, which decreases the chances of errors.

Learn more about creating a continuous integration and delivery (CI/CD) pipeline in the AWS Cloud. For more information, see "Complete CI/CD with AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, and AWS CodePipeline" in the *AWS DevOps Blog* at https://aws.amazon.com/blogs/devops/complete-ci-cd-with-aws-codecommit-aws-codebuild-aws-codedeploy-and-aws-codepipeline/.

You can initiate this same solution by using GitHub instead of AWS CodeCommit. For more information, see "GitHub connections" in the *AWS CodePipeline User Guide* at https://docs.aws.amazon.com/codepipeline/latest/userguide/connections-github.html.

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.



If you build infrastructure with code, you gain the benefits of repeatability and reusability while building your environments. Build the same complex environments with one template, or a combination of templates. This example uses the architecture template to create identical resources in different AWS Regions. One is the development environment, and the other is the production environment.

For builds to match a specific context, create environments that are dependent on conditions. For example, a template can be designed so that different Amazon Machine Images (AMIs) are used in the development or the production environments.

DO NOT COPY
javierartiga.espublico@gmail.com

© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.