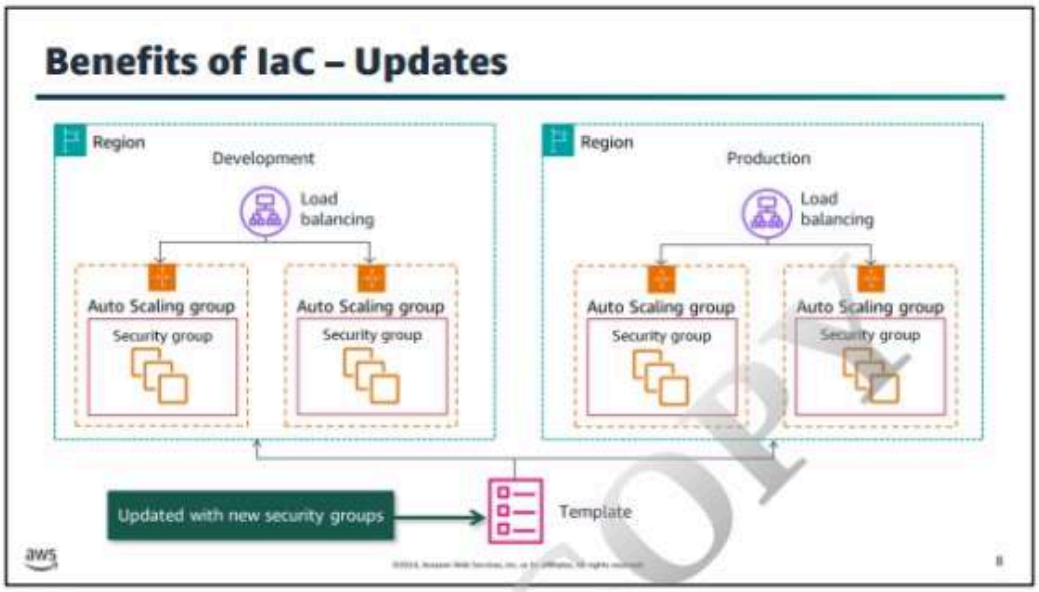
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

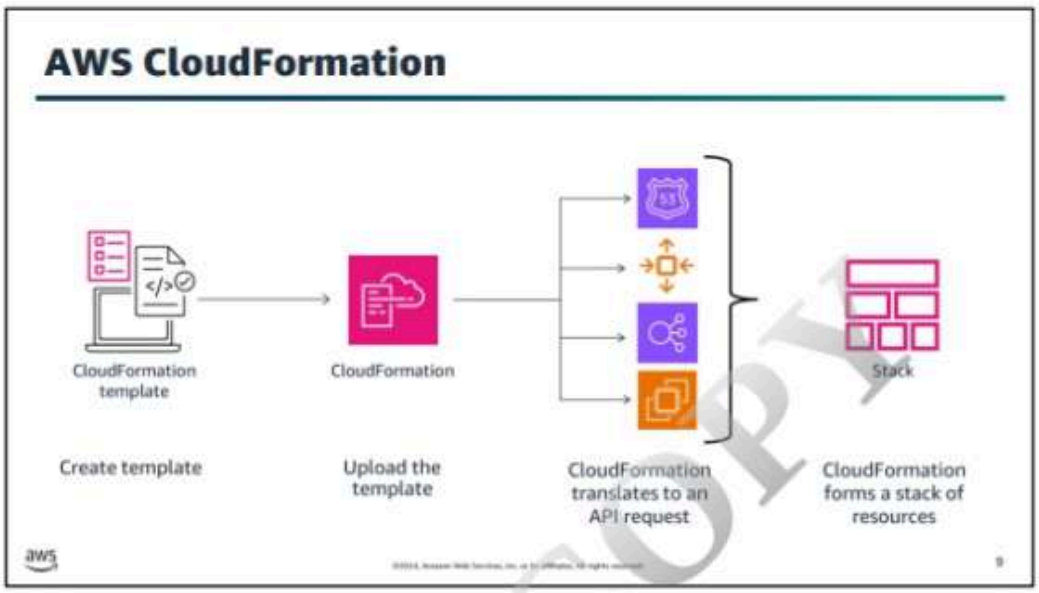


In this scenario, the template has been updated to add new security groups to the instance stacks. With this one change to the template that is used to launch these environments, all environments add the new security group resource. This feature makes resource maintenance less complicated, provides consistency, and reduces effort through parallelization.

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.



Essentially, AWS CloudFormation is an API wrapper. When you create an EC2 instance in the AWS Management Console, you initiate an API call to the Amazon EC2 service. The information that you enter through the wizard is passed on as parameters. The resources that you define in your CloudFormation template become API calls to AWS services, just like in the AWS Management Console. CloudFormation uses those APIs and manages the dependencies and relationships.

You can author your CloudFormation template with any code editor, and then check it in to a version control system such as GitHub. Review your files before deploying. CloudFormation is available in all AWS Regions, and you pay for only the resources that you use.

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.

Understanding CloudFormation

- Written as JSON or YAML
- Describes the resources to be created or modified
- Treated as source code:
  - Code reviewed
  - Version controlled

Template

JSON

```
{
  "Resources" : {
    "HelloBucket" : {
      "Type" : "AWS::S3::Bucket"
    }
  }
}
```

YAML

```
Resources:
 HelloBucket:
  Type: AWS::S3::Bucket
```

10

Additional points for a CloudFormation template are as follows:

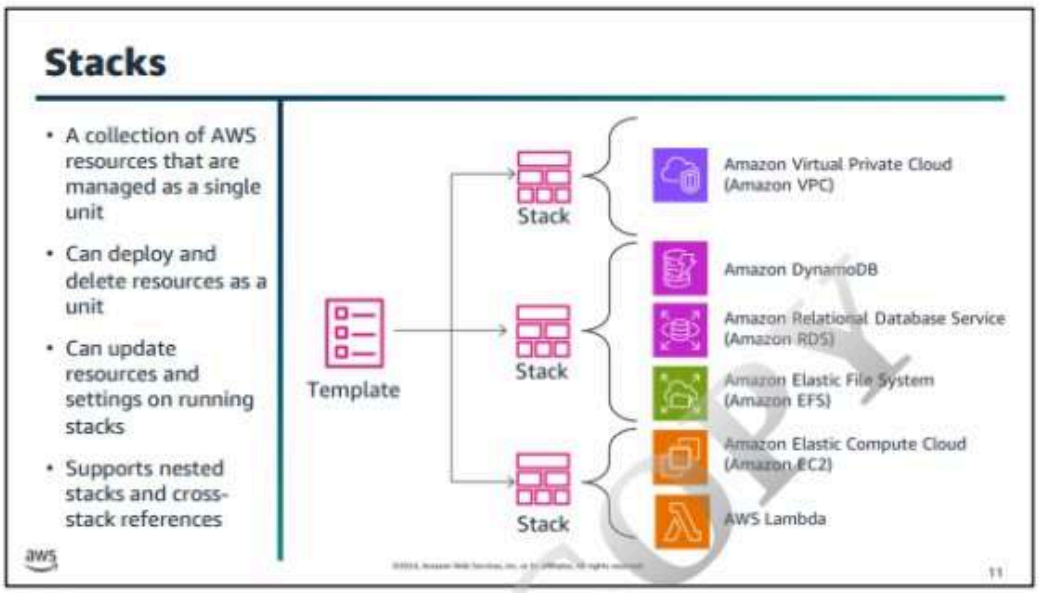
- You can manage a template by using your choice of version control—for example, Git or Subversion (SVN).
- Define an entire application stack (all resources that your application requires) in a JSON template file.
- Define runtime parameters for a template (EC2 instance size, Amazon EC2 key pair, and others).
- If you created an AWS resource outside CloudFormation management, you can bring this existing resource into CloudFormation management by using *resource import*. Resource import is a manual process where you describe the existing resources in your CloudFormation template and then import them into a stack.

YAML-formatted CloudFormation templates follow the same anatomy as existing JSON formatted templates and support all the same features.

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.



All resources in a stack are defined by the stack's CloudFormation template. You can manage a collection of resources by creating, updating, or deleting stacks. For example, a stack can include all resources that are required to run a web application, including a web server, database, and networking rules. If you no longer require that web application, you can delete the stack, which deletes all of its related resources.

CloudFormation treats the stack resources as a single unit. They must all be created or deleted successfully for the stack to be created or deleted. If a resource can't be created, CloudFormation rolls the stack back and deletes any resources that were created. If a resource can't be deleted, CloudFormation retains any remaining resources until the entire stack can be successfully deleted.

When you must make changes to a stack's settings or change its resources, you update the stack instead of deleting it and creating a new stack. To change a running stack, submit the changes that you want to make by providing a modified template, new input parameter values, or both. CloudFormation generates a change set by comparing your stack with the changes that you submitted.

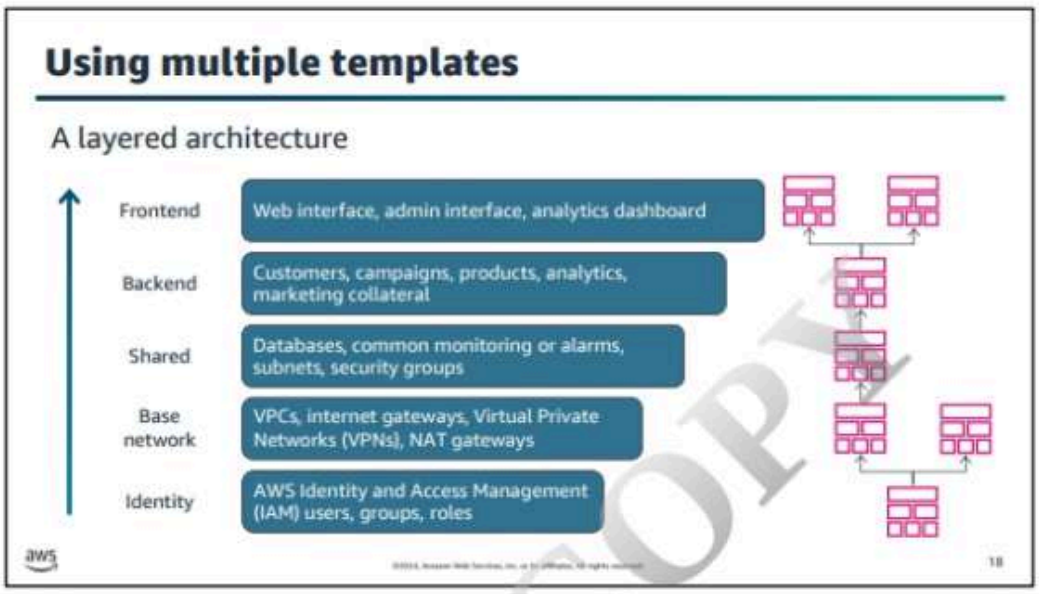
You can create cross-stack references so that you can share outputs from one stack with another stack.

For more information about cross-stack references, see the walkthrough, "Refer to resource outputs in another AWS CloudFormation stack" in the *AWS CloudFormation User Guide* at https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/walkthrough-crossstackref.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 layered architecture organizes stacks into multiple horizontal layers that build on top of one another. Each layer has a dependency on the layer directly under it. You can have one or more stacks in each layer, but within each layer, your stacks should have AWS resources with similar lifecycles and ownership.

For more information about a layered architecture, see "AWS CloudFormation best practices" in the *AWS CloudFormation User Guide* at https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/best-practices.html.

DO NOT COPY javierartiga.espublico@gmail.com

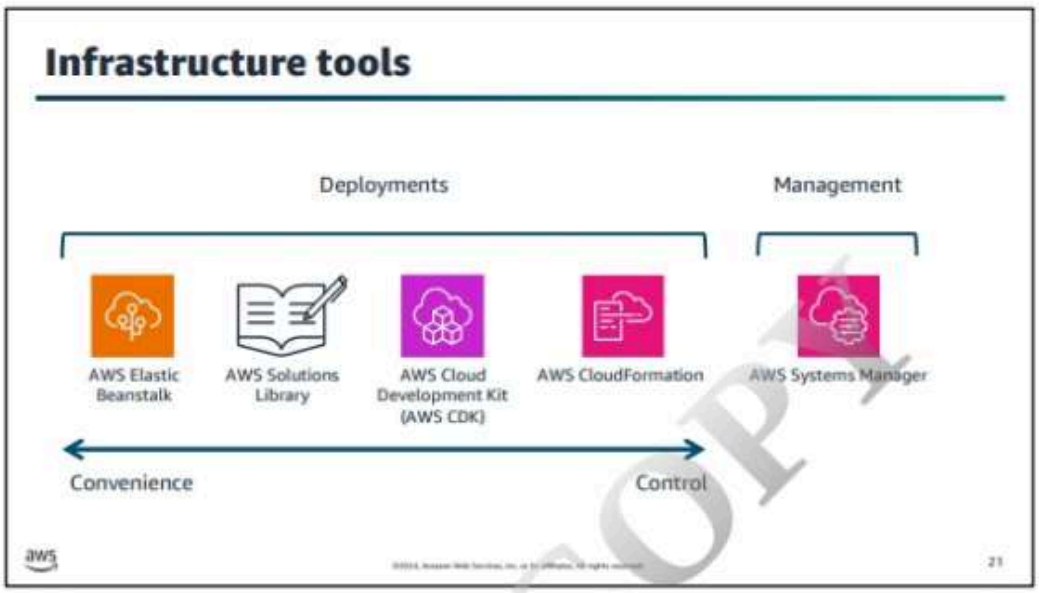
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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 chief technology officer asks, "How can we deploy, maintain, and scale applications in the cloud?" The operations team is looking for tools that can automate the deployment of infrastructure and help them manage resources after the resources are deployed.

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.



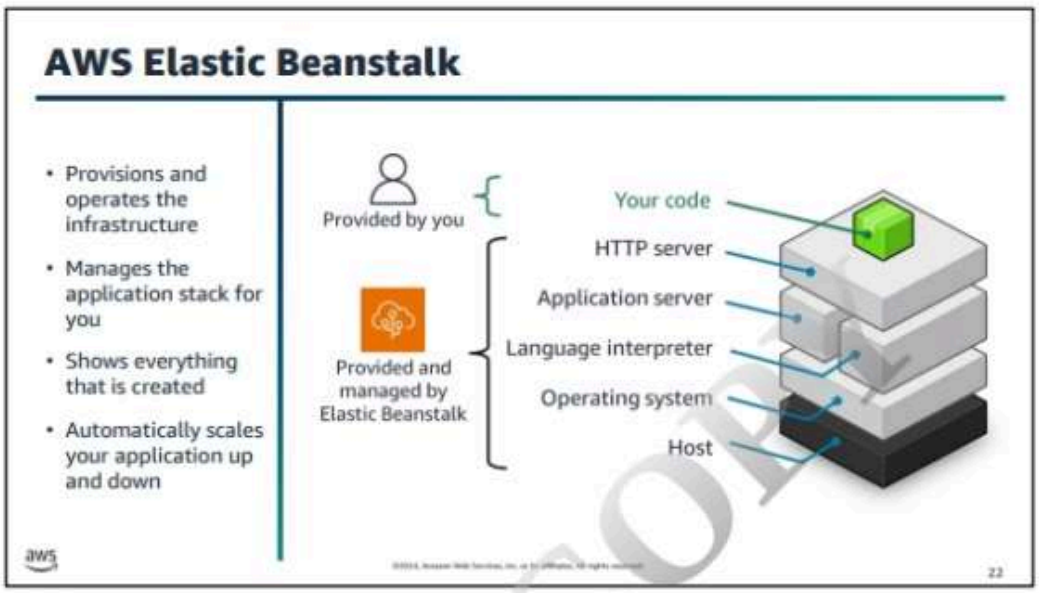
When choosing infrastructure deployment tools, you must find a balance between convenience and control. Some tools give you complete control and have you choose every component and configuration. Though you can customize your deployment to fit your business needs, this method requires greater expertise and more resources to manage and maintain. Other tools are designed for convenience and include preconfigured infrastructure templates for common solutions. Though these tools are less complicated to use and require less maintenance, you do not always have the ability to customize your infrastructure components.

You can automate your infrastructure deployments by using the following tools:

- **AWS Elastic Beanstalk** – Elastic Beanstalk integrates with developer tools and provides a one-stop experience to manage the application lifecycle. Elastic Beanstalk provisions and manages application infrastructure to support your application.
- **AWS Solutions Library** – AWS Solutions Library carries solutions that AWS and AWS Partners have built for a broad range of industry and technology use cases. These solutions include the tools that you need to get started quickly, such as CloudFormation templates, scripts, and reference architectures.
- **AWS Cloud Development Kit (AWS CDK)** – AWS CDK is an open-source software development framework to model and provision your application resources by using common programming languages. AWS CDK simplifies the creation and deployment of CloudFormation templates. It offers infrastructure components and groups of components that are preconfigured according to best practices. However, you can still customize your components and their settings.
- **AWS CloudFormation** – With CloudFormation, you define every resource and its configuration. You have granular control over every component of your infrastructure.
- **AWS Systems Manager** – With Systems Manager, you can view and control your infrastructure on AWS. You can automate or schedule a variety of maintenance and deployment tasks.

© 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.

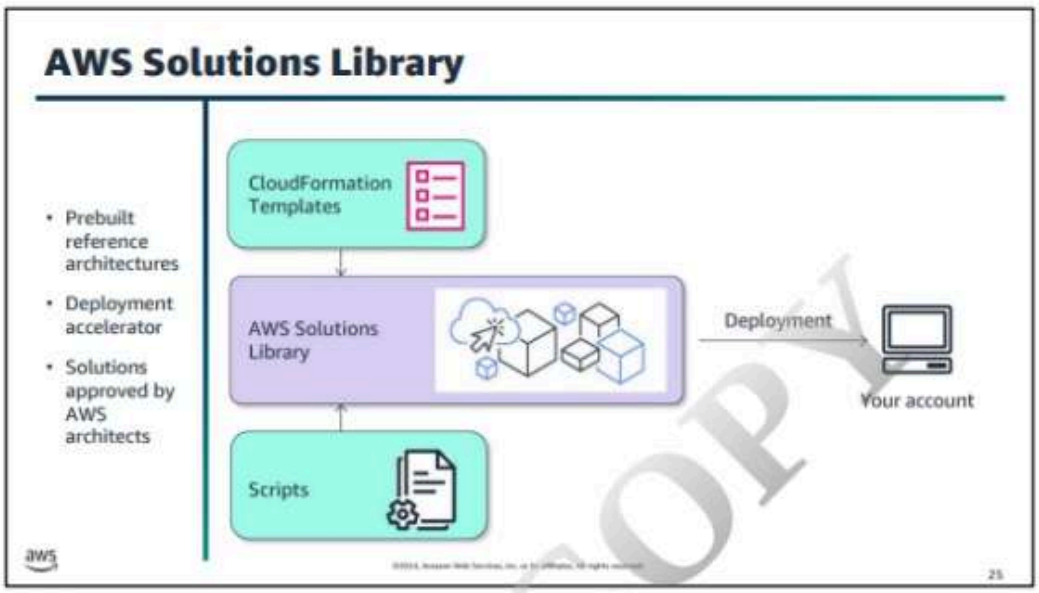


The goal of Elastic Beanstalk is to help developers deploy and maintain scalable web applications and services in the cloud without worrying about underlying infrastructure. Elastic Beanstalk configures each EC2 instance in your environment with the components necessary to run applications for the selected application type. You don't need to worry about logging into instances to install and configure your application stack. With Elastic Beanstalk, you can provision infrastructure to support common application designs, such as web applications and worker services.

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.

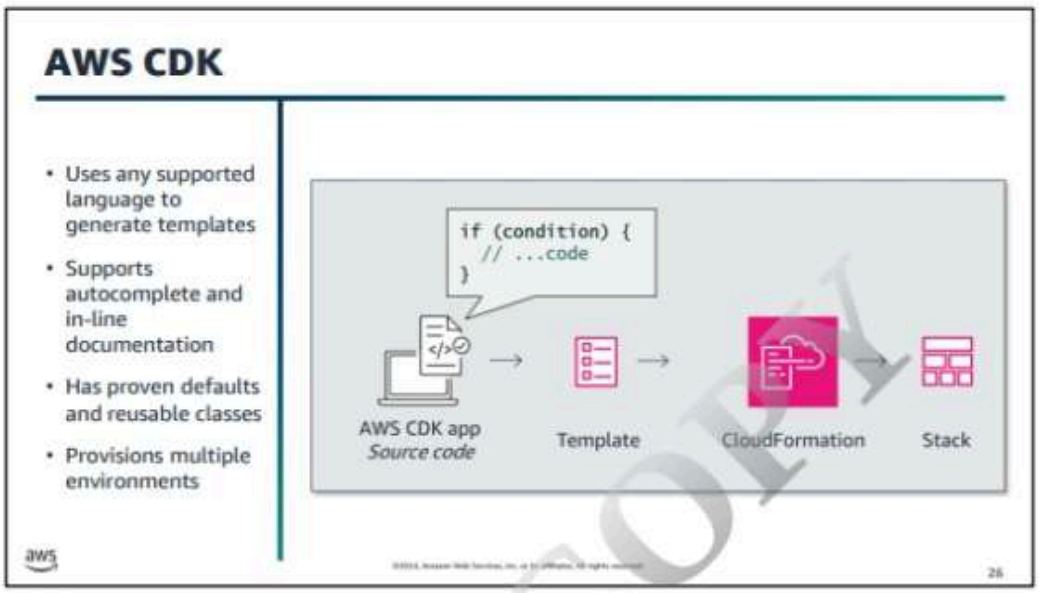


AWS Solutions Library helps you solve common problems and build faster by using AWS. Solutions are vetted by AWS architects and designed to be operationally effective, reliable, secure, and cost efficient. Many AWS solutions come with prebuilt CloudFormation templates. The templates can also include a detailed architecture, a deployment guide, and instructions for automated and manual deployment. You will be charged for the resources that you use to create and run this environment.

For more information, see AWS Solutions Library at https://aws.amazon.com/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.

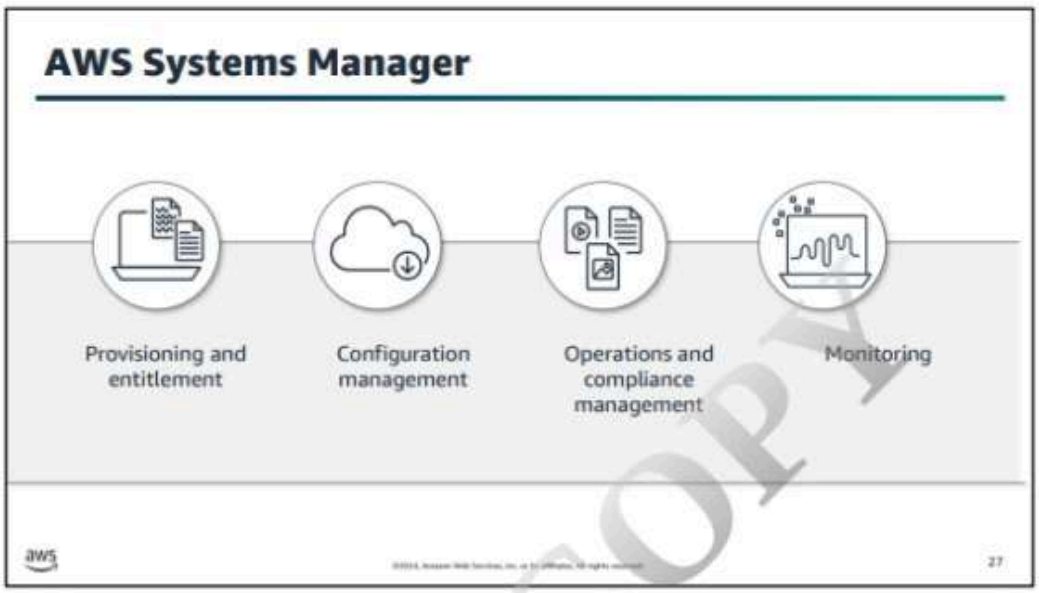


AWS CDK is a software development framework that defines your cloud application resources by using a declarative model and familiar programming languages. AWS CDK includes a library of customizable constructs, which are building blocks that consist of one or more resources and include common configurations. You can use AWS CDK to generate CloudFormation templates and deploy your infrastructure along with your application runtime assets.

You can use AWS CDK with common programming languages such as Python, JavaScript, TypeScript, Java, or C#.

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.



When designing infrastructure, you must plan for its management. This planning affects your infrastructure deployments because you must grant the correct security policies to your management tools. You might also need to install management agents on your instances. AWS Systems Manager provides a central place to view and manage your AWS resources, so you can have complete visibility and control over your operations.

With Systems Manager, you can do the following:

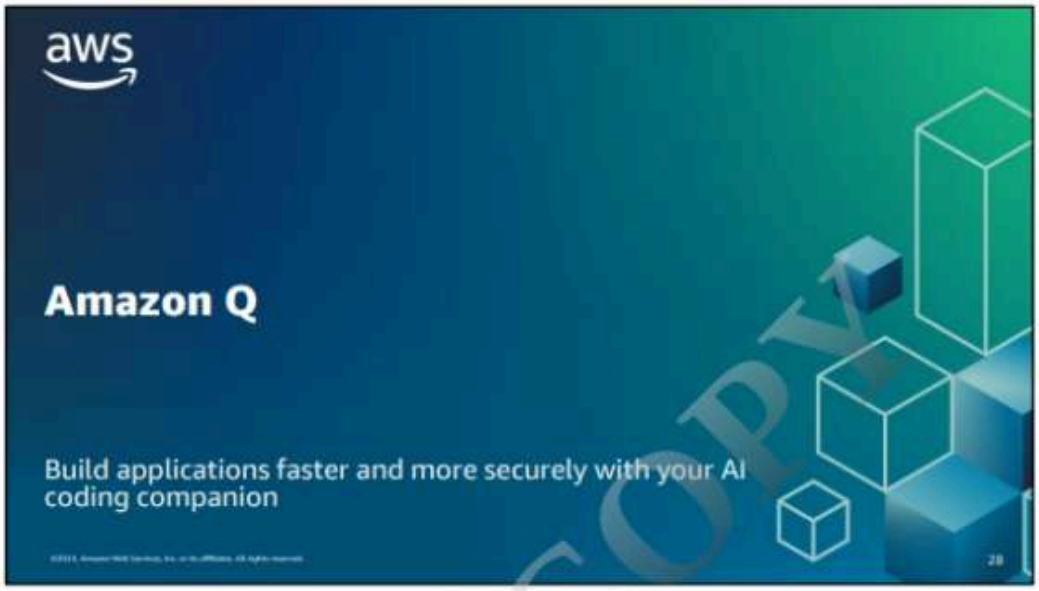
- Create logical groups of resources such as applications, different layers of an application stack, or development and production environments.
- Select a resource group and view its recent API activity, resource configuration changes, related notifications, operational alerts, software inventory, and patch compliance status.
- Take action on each resource group depending on your operational needs.
- Centralize operational data from multiple AWS services and automate tasks across your AWS resources.

You can open Systems Manager from the Amazon EC2 console. You select the instances that you want to manage, and define the management tasks that you want to perform. Systems Manager is available at no cost to manage your Amazon EC2 and on-premises resources.

For more information about Systems Manager, see "What is AWS Systems Manager?" in the *AWS Systems Manager User Guide* at https://docs.aws.amazon.com/systems-manager/latest/userguide/what-is-systems-manager.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.

Amazon Q: Generative AI-powered assistant

- Secure and private by design
- Can be customized with your own data
- Provides assistants that are designed for specific use cases

Amazon Q Developer	Amazon Q Business
• Designed for developers and IT professionals	• Designed for business analysts and every employee
• Generates code and helps you understand, build, extend, and operate AWS applications	• Answers questions, generates and summarizes content, and completes non-coding tasks

29

Amazon Q is a generative AI-powered assistant that can understand and respect existing governance identities, roles, and permissions. It is designed to be secure and private by default. Administrative controls help you to apply guardrails to customize and control responses. When it comes to collecting your data to improve the underlying model, you have options. For users who access Amazon Q with a subscription, you are automatically opted out. For the free tier, you also have the option to opt out of data collection.

Amazon Q Developer is a coding assistant that those without extensive experience can use. It is trained on years of high-quality AWS examples and documentation. It can also be trained on your company's code and systems. Amazon Q Developer can chat about code, provide in-line code completions, generate new code, and scan your code for security vulnerabilities. It can also make code upgrades and improvements, such as language updates, debugging, and optimizations. With Amazon Q Developer, you can describe the application you want to create in natural language, and Amazon Q Developer will generate and suggest code for the application. You can build applications without needing to write all of the code yourself. The no-code approach of Amazon Q Developer can be particularly useful for creating simple web applications, automating business processes, or building prototypes and proofs of concept.

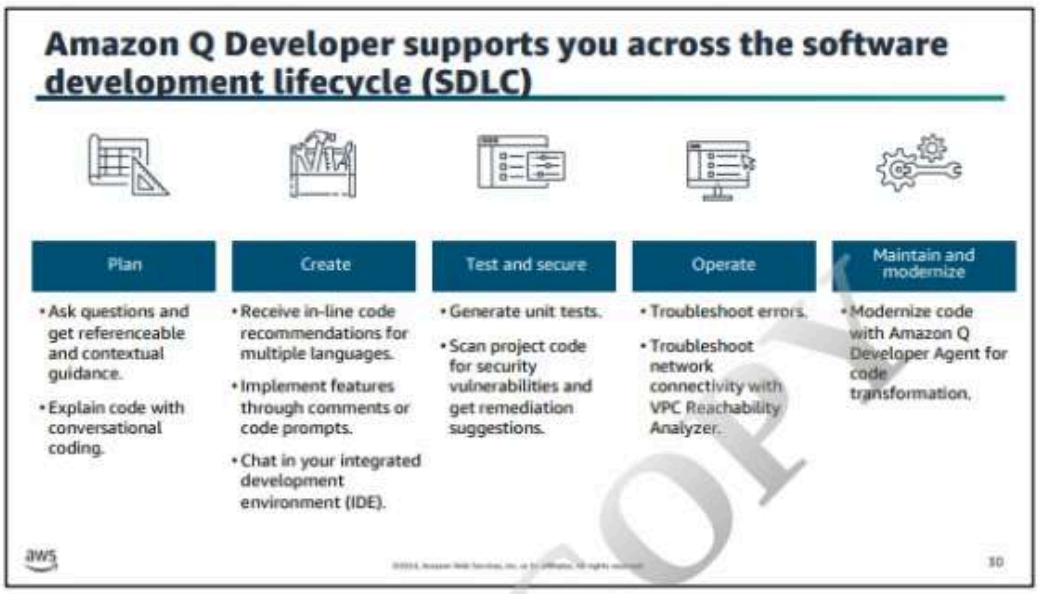
Amazon Q Business is a generative AI-powered assistant that can answer questions, provide summaries, generate content, and complete tasks. You can tailor Amazon Q Business to do any of those tasks based on data and information in your enterprise systems. You can also allow administrators to apply guardrails to customize and control responses. With admin controls you can make sure that if a user doesn't have permission to access certain data without Amazon Q, they can't access it using Amazon Q either.

For more information, see the following resources:

- Amazon Q – Generative AI Assistant at https://aws.amazon.com/q/
- Amazon Q Developer Getting Started in AWS Skill Builder at https://explore.skillbuilder.aws/learn/course/19491/play/122791

© 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 Q supports developers across the software development lifecycle (SDLC) phases in several ways.

- **Plan** – You can use Amazon Q in the AWS Management Console to ask questions, receive best practices and recommendations, optimize Amazon EC2 instances, and learn more about the AWS Well-Architected Framework. In the integrated development environment (IDE), Amazon Q can help you understand your code base and get up to speed on projects.
- **Create** – Amazon Q helps improve development productivity through in-line code recommendations directly in your IDE or command line interface (CLI), using natural language to generate new features. You can ask Amazon Q questions without leaving the IDE. The in-line code recommendations are available for multiple coding languages. For more information, see "Supported languages for Amazon Q Developer in the IDE" at https://docs.aws.amazon.com/amazonq/latest/qdeveloper-ug/q-language-ide-support.html.
- **Test and secure** – Amazon Q helps you confirm that your code is working and secure through unit test assistance. It helps you to find and fix security vulnerabilities earlier in the development cycle.
- **Operate** – Amazon Q can help troubleshoot errors, for example, in AWS Lambda, Amazon EC2, and Amazon Elastic Container Service (Amazon ECS). You can analyze your VPC reachability and get better debug and optimization tips.
- **Maintain and modernize** – Amazon Q Developer Agent for code transformation helps you maintain and modernize code by upgrading projects to more up-to-date language versions.

For more information, see "Accelerate your Software Development Lifecycle with Amazon Q" in the *AWS DevOps Blog* at https://aws.amazon.com/blogs/devops/accelerate-your-software-development-lifecycle-with-amazon-q/.

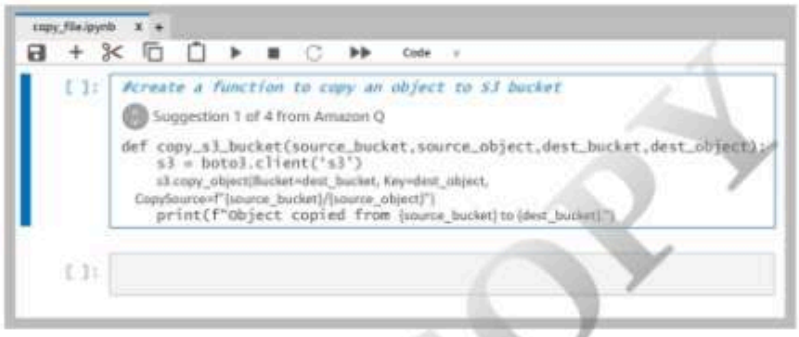
DO NOT COPY javierartiga.espublico@gmail.com

© 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 Q Developer: Code recommendations

- Recommends code based on plain English comments and prior code
- You accept or reject recommendations



31

With Amazon Q Developer, you can do the following:

- Get sample code to get started on your tasks, such as data preparation.
- Cycle through the various recommendations and choose which code snippets to accept or reject.

Amazon Q can provide you with code recommendations in real time. As you write code, Amazon Q automatically generates suggestions based on your existing code and comments. Your personalized recommendations can vary in size and scope, ranging from a single line comment to fully formed functions. You can also customize the suggestions that Amazon Q generates to proprietary algorithmic techniques, enterprise code style, and to your software development team's internal libraries.

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.

Plan and implement new functionality across files in your workspace

AMAZON Q: CHAT

Chat | Q - Dev

To get started, describe the task you are trying to accomplish.

Describe the task.

Create a class to save a log of the tokens generated to DynamoDB

Ok, let me create a plan. This may take a few minutes.

1. Create a new class for logging token generation to DynamoDB
   1. Create a new file: `TokenLogger.cs`
   2. Import the required namespaces:
      - `Amazon.DynamoDBv2`
      - `Amazon.DynamoDBv2.DataModel`
   3. Define a new class `TokenLogger` that implements
   4. Define a constructor that initializes the DynamoDB client and table name
   5. Define a method `LogToken` with the following input parameters:

Amazon Q Developer tells you the plan to accomplish the task.

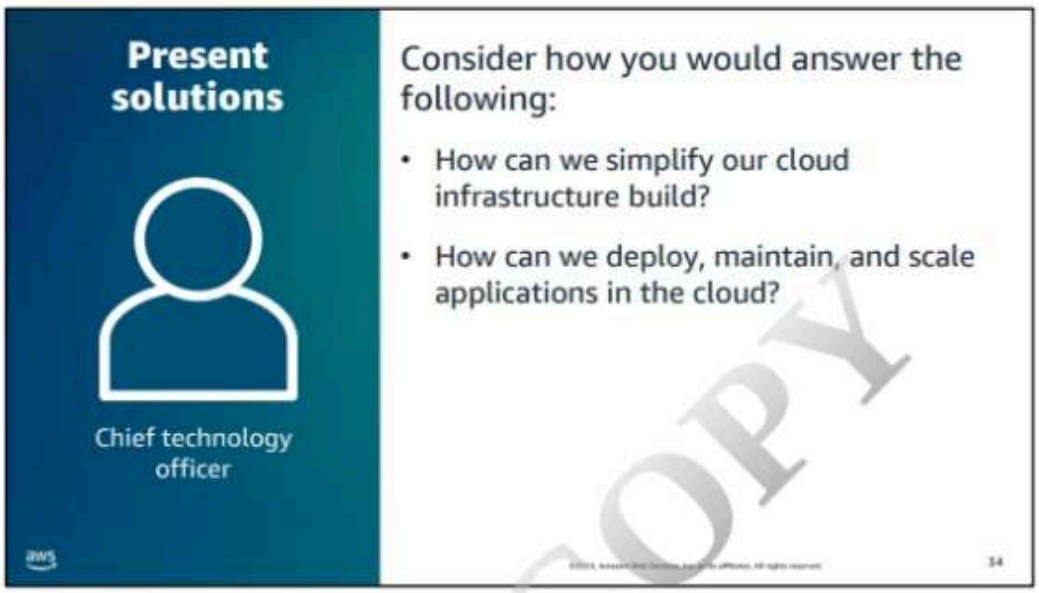
DO NOT COPY
javierartiga.espublico@gmail.com

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.



Imagine that you are now ready to talk to the chief technology officer 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:

- You can use AWS CloudFormation to simplify your infrastructure with an infrastructure as code (IaC) approach.
- You can deploy and maintain your infrastructure by using AWS CDK, solutions from the AWS Solutions Library, and AWS Elastic Beanstalk. You can use AWS Systems Manager to automate infrastructure management.

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:

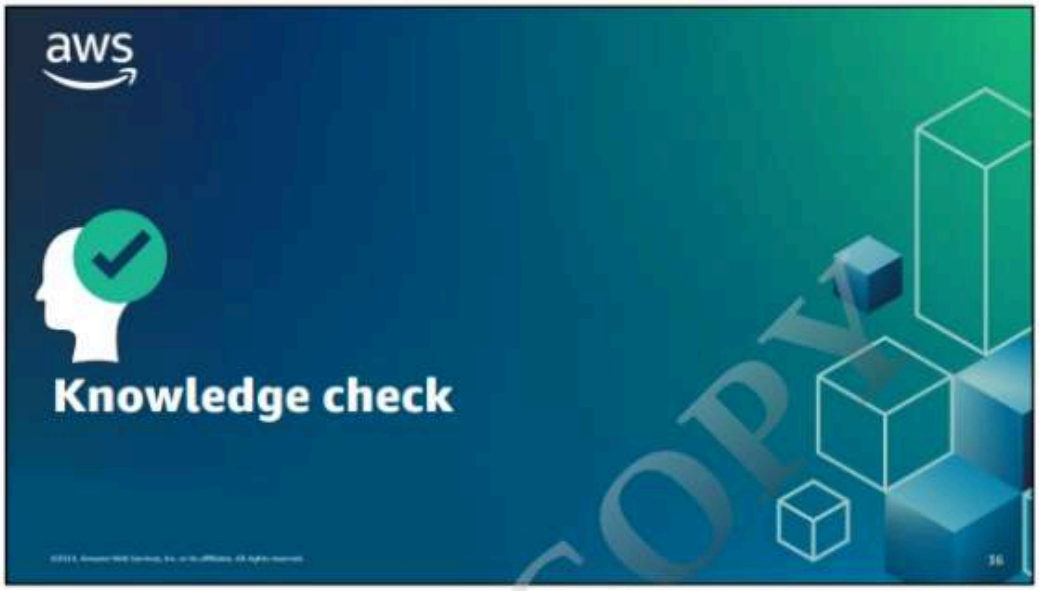
- ✓ Amazon CloudFormation
- ✓ Amazon Q
- ✓ Infrastructure management

Next, you review the following:

- 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

© 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 an AWS CloudFormation stack?

Choice	Response
A	All of the provisioned resources that are defined in a CloudFormation template
B	All of the resources that are identified as drifted in a CloudFormation template
C	A condition in which resources are added on top of each other
D	The properties of a single resource

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 an AWS CloudFormation stack?

Choice	Response
A correct	All of the provisioned resources that are defined in a CloudFormation template
B	All of the resources that are identified as drifted in a CloudFormation template
C	A condition in which resources are added on top of each other
D	The properties of a single resource

The correct answer is A, all of the provisioned resources that are defined in a CloudFormation template.

A stack is a collection of AWS resources that a CloudFormation template generates, and it is managed as a single unit.

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 statements are benefits of using AWS Cloud Development Kit (AWS CDK) with AWS CloudFormation? (Select TWO.)

Choice	Response
A	Developers can use common programming languages.
B	Bulk discounts are automatically applied to resource usage.
C	Developers can call preconfigured resources with proven defaults.
D	Components are limited to a single user.
E	Using AWS CDK does not require an AWS account or credentials.

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 statements are benefits of using AWS Cloud Development Kit (AWS CDK) with AWS CloudFormation? (Select TWO.)

Choice	Response
A correct	Developers can use common programming languages.
B	Bulk discounts are automatically applied to resource usage.
C correct	Developers can call preconfigured resources with proven defaults.
D	Components are limited to a single user.
E	Using AWS CDK does not require an AWS account or credentials.

40

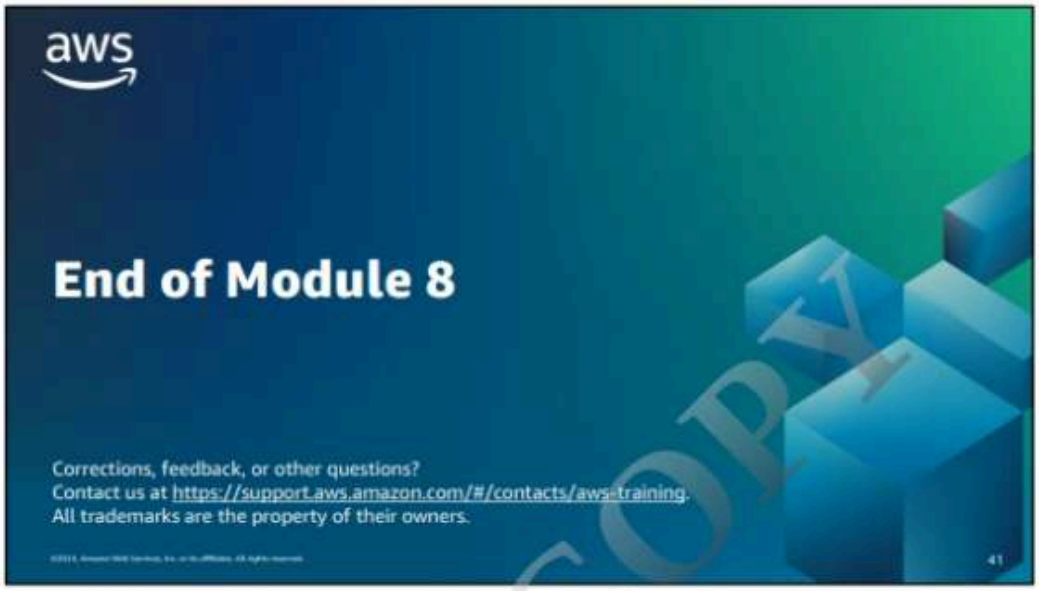
The correct answers are A and C, developers can use common programming languages and developers can call preconfigured resources with proven defaults.

AWS CDK is a software development framework that you can use to define your cloud application resources. You can use it with familiar programming languages, such as TypeScript, Python, Java, and .NET. AWS CDK offers different cloud components that include configuration detail, boilerplate, and glue logic for using one or multiple AWS services.

For more information about the features of AWS CDK, see AWS Cloud Development Kit features at https://aws.amazon.com/cdk/features/.

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.

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 percentage of your workloads run on containers?

A. Less than 10 percent

B. 10–50 percent

C. More than 50 percent

D. I'm not sure

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
- Microservices
- Containers
- Container services
- Present solutions
- Knowledge check

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 your compute operations manager meets with you to discuss different application architectures and how you support them in the cloud. Here are some questions that they are asking.

At the end of this module, you meet with the compute 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.

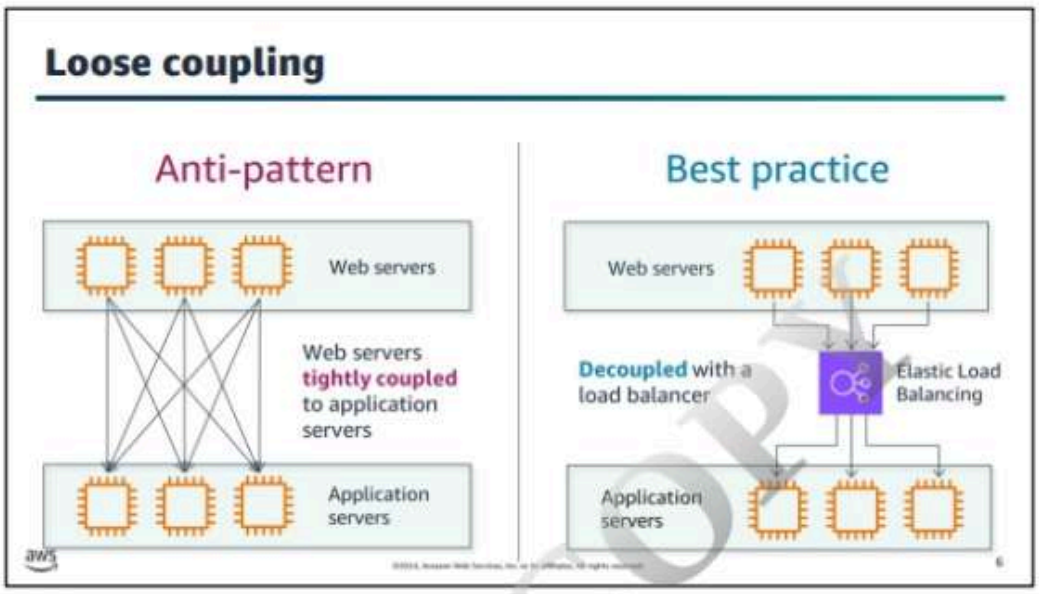
The compute operations manager asks, "How can we make components of our applications more independent so that changes in one service will not affect any other?"

The company has some legacy applications that they want to rearchitect in the cloud. These applications have many interdependent components, which makes them difficult to update and to isolate errors. The company is asking you to research different ways to architect these applications.

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.



Traditional *monolithic* infrastructures revolve around chains of tightly integrated servers, each with a specific purpose. When one of those components or layers goes down, the disruption to the system can be fatal. This configuration also impedes scaling. If you add or remove servers at one layer, you must also connect every server on each connecting layer.

With *loose coupling*, you use managed solutions as intermediaries between layers of your system. The intermediary automatically handles failures and scaling of a component or a layer. Two primary solutions for decoupling your components are load balancers and message queues.

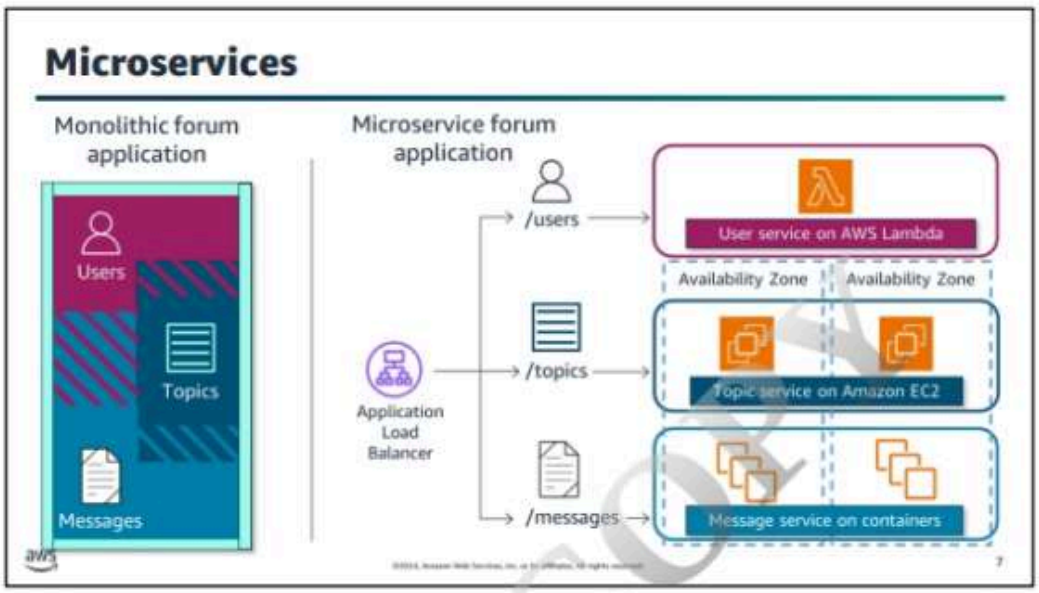
The diagram on the left illustrates a collection of web and application servers that are tightly coupled. In the tightly coupled architecture, the connection between a web server and an application will give an error if one application server goes down.

The drawing on the right shows a load balancer—in this case, using Elastic Load Balancing (ELB). This load balancer routes requests between the web servers and the application servers. If a server fails in the loosely coupled architecture, the load balancer will automatically start directing all traffic to the two healthy servers.

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.



Microservices are an architectural and organizational approach to software development. Using a microservices approach, you design software as a collection of small services. Each service is deployed independently and communicates over well-defined APIs. This process speeds up your deployment cycles, fosters innovation, and improves both maintainability and scalability of your applications.

Autonomous
The component services in a microservices architecture are isolated from one another and communicate through an API. Therefore, you can develop, update, deploy, operate, and scale a service without affecting the other services. Small autonomous teams can own these services, which can provide an agile approach.

Specialized
You design each service for a set of capabilities that focuses on solving a specific problem. Teams can write each service in the programming languages that are best suited to that service. They can also host their services on different compute resources.

In this example, a monolithic forum application is refactored to use a microservices architecture: a user service, a topic service, and a message service. The /users service team runs the user service on AWS Lambda. The /topics service team runs the topics service on Amazon Elastic Compute Cloud (Amazon EC2). The /messages service team runs the messages service on containers. The microservices application is distributed across two Availability Zones and manages traffic with an Application Load Balancer.

For more information about microservices architecture, see "Simple microservices architecture on AWS" in the *Implementing Microservices on AWS, AWS Whitepaper* at https://docs.aws.amazon.com/whitepapers/latest/microservices-on-aws/simple-microservices-architecture-on-aws.html.

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

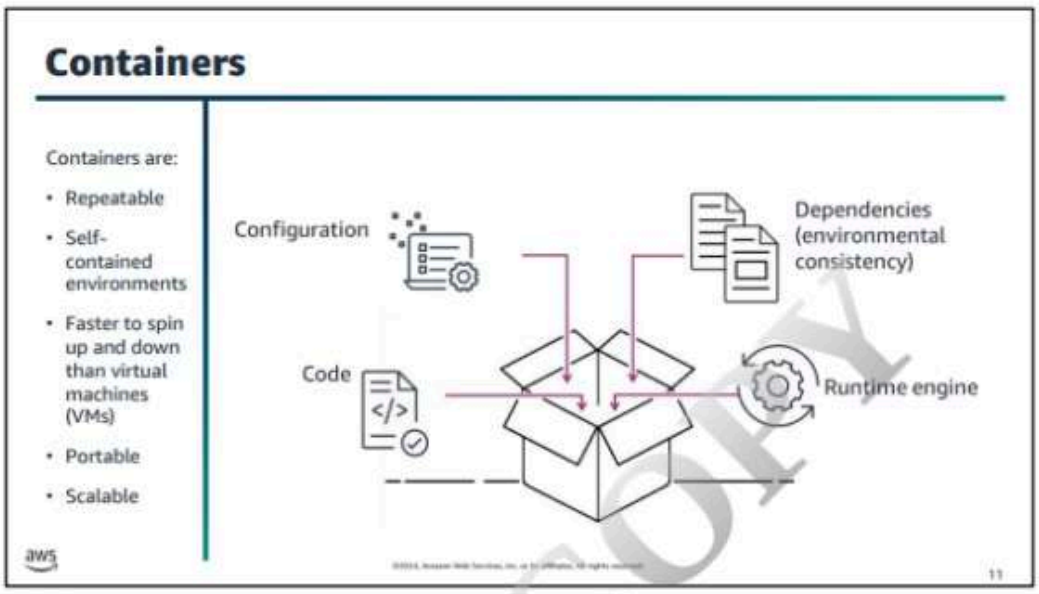
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

The compute operations manager asks, "What are the benefits of using containers for our compute needs?" The software engineering team is considering refactoring some of the company's applications to use containers. The compute operations manager is asking you to research the benefits of containers and needs to understand how containerized applications are hosted.

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 benefits of a microservice-oriented architecture should trickle down to an infrastructure level. You can achieve that with containers.

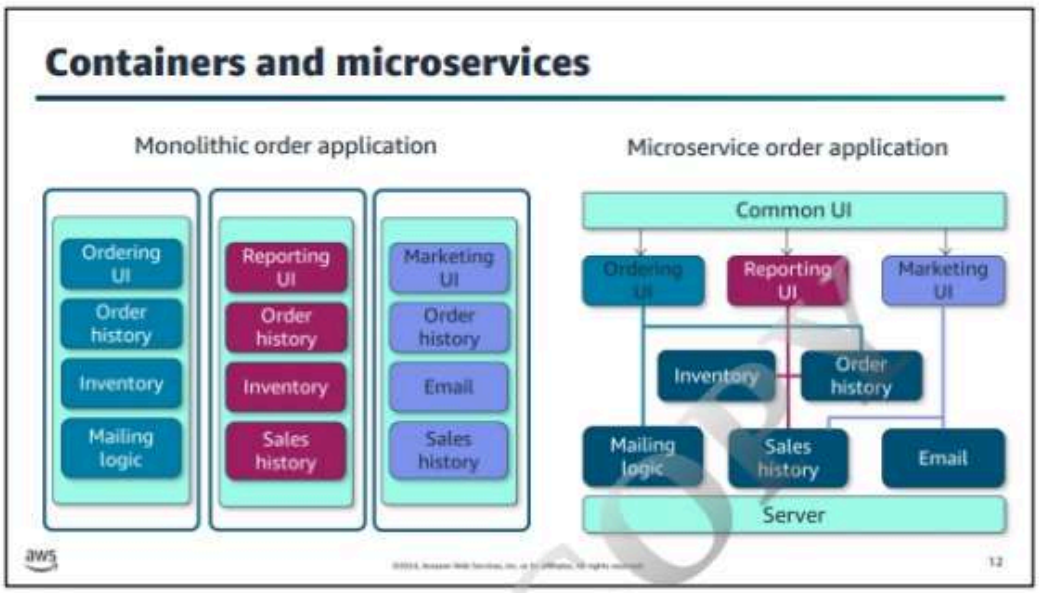
Although running virtual machines (VMs) in the cloud gives you a dynamic, elastic environment, you can simplify your developers' processes. Containers provide a standard way to package your application's code, configurations, and dependencies into a single object.

Containers share an operating system installed on the server. They run as resource-isolated processes, which offers quick, reliable, and consistent deployments, regardless of the 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.



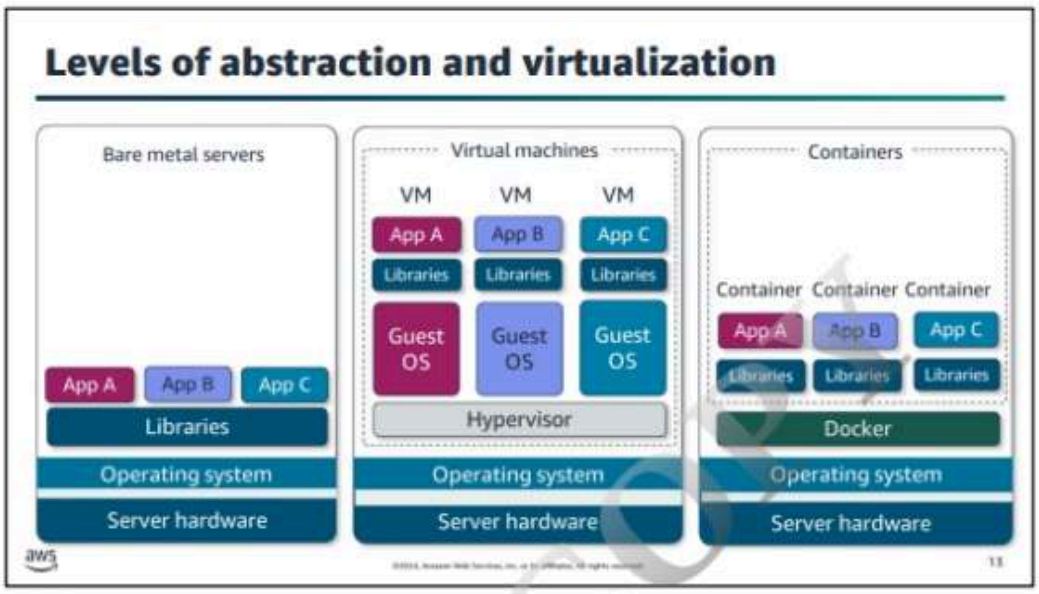
Containers are an ideal choice for microservice architectures because they are scalable, portable, and continuously deployable. Earlier in this module, you learned how microservice architectures decompose traditional, monolithic architectures into independent components that run as services and communicate through lightweight APIs. With these microservice environments, you can iterate quickly, with increased resilience, efficiency, and overall agility.

You can build each microservice on a container. Because each microservice is a separate component, it can tolerate failure better. If a container fails, you can shut it down and start a new one quickly for that particular service. If a certain service has a lot of traffic, you can scale out the containers for that microservice. This arrangement eliminates the need to deploy additional servers to support the entire application. Microservices and containers are also great for continuous deployment. You can update individual services without impacting any of the other components of your application.

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 bare metal server runs a standalone operating system (OS) with one or many applications by using libraries. Costs remain constant, whether the server is running at 0 percent usage or 100 percent usage. To scale, you must buy and configure additional servers. It is difficult to build applications that work on multiple servers because the OS on those servers would need to be the same. You also must synchronize the application library versions.

With virtual machines, you isolate applications and their libraries with their own full OS. The downside of VMs is that the virtualization layer is *heavy*. Each VM has its own OS, which requires more host CPU and RAM and reduces efficiency and performance. Having an individual OS for each VM also means more patching, more updates, and more space on the physical host.

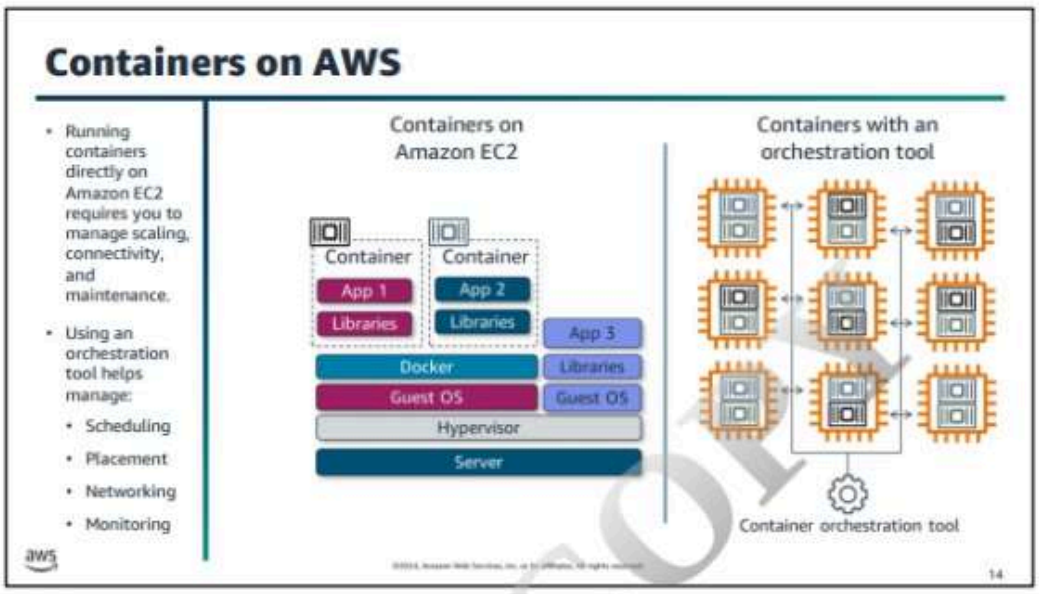
With containerization, containers share a machine's OS system kernel, and the underlying OS file system is exposed. Sharing a machine's OS system kernel permits shared libraries and can permit individual libraries as needed. As a result, containers are highly portable. You can also start and stop containers faster than VMs. Containers are lightweight, efficient, and fast.

Unlike a VM, containers can run on any Linux system, with appropriate kernel feature support and the Docker daemon, which makes them portable. Your laptop, your VM, your Amazon EC2 instance, and your bare metal server are all potential hosts.

The lack of a hypervisor requirement also results in almost no noticeable performance overhead. The processes are communicating directly to the kernel and are largely unaware of their container silo. Most containers boot in only a couple of seconds.

© 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 running containers on AWS, you have multiple options.

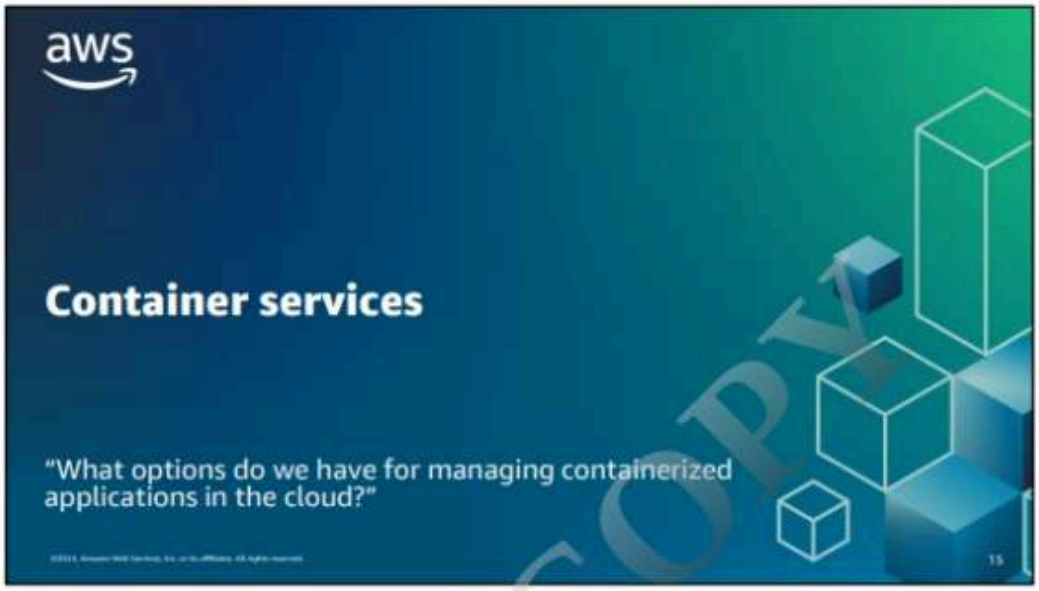
Running containers on top of an EC2 instance is common practice and uses elements of VM deployments and containerization. This diagram shows the underlying server infrastructure: a physical server, the hypervisor, and two virtual guest operating systems. One of these operating systems runs Docker, and the other runs a separate application. The virtual guest OS with Docker installed can build and run containers. Though possible, this type of deployment can only scale to the size of the EC2 instance that is used. You also must actively manage the networking, access, and maintenance of your containers.

Using an orchestration tool is a scalable solution for running containers on AWS. An orchestration tool uses a pool of compute resources, which can include hundreds of EC2 instances to host containers. The orchestration tool launches and shuts down containers as demand on your application changes. It manages connectivity to and from your containers. It also helps manage container deployments and updates.

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.

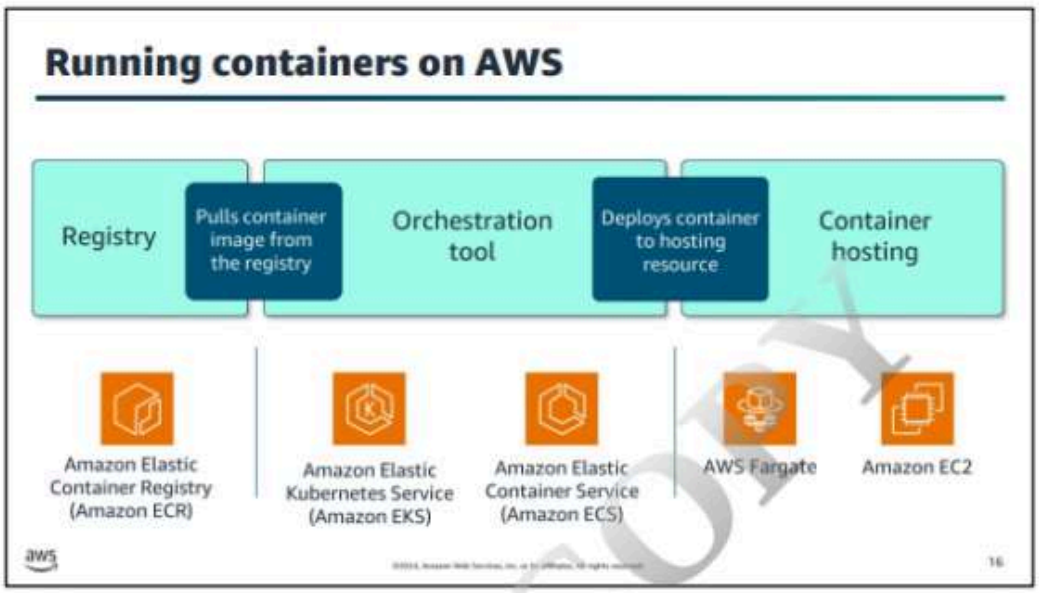


The compute operations manager asks, "What options do we have for managing containerized applications in the cloud?" The software engineering team is considering refactoring some of the company's applications to use containers. The compute operations manager is asking you to research which AWS services support containerized applications in 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.



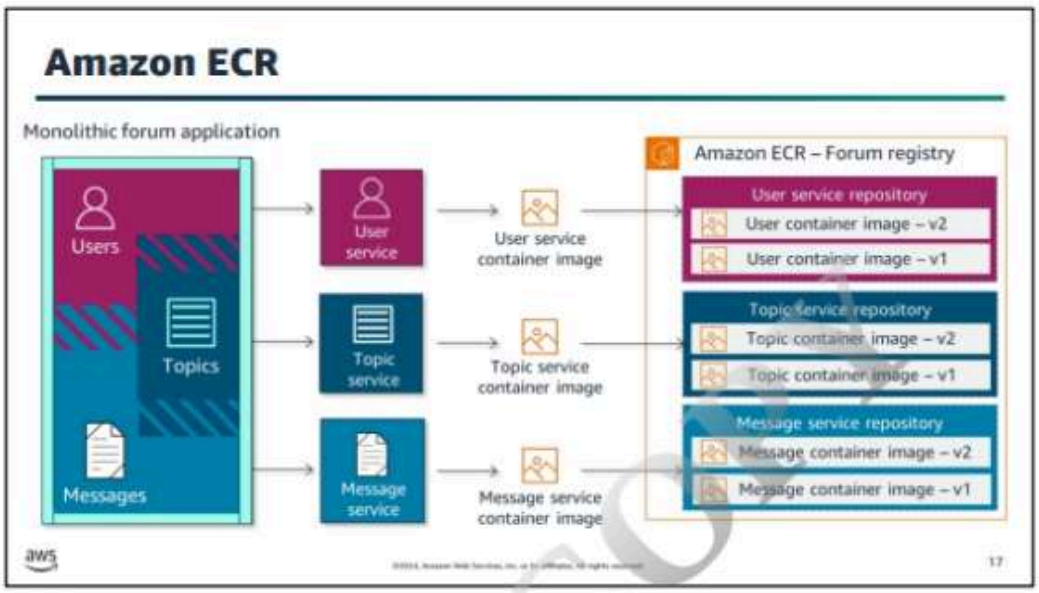
Deploying your managed container solutions on AWS involves selecting and configuring the following components:

- **Registry** – When you develop containerized applications, you build a container image that holds everything needed to run your container. The container image includes application code, runtime, system tools, system libraries, and settings. You push your images to a repository for version control and pull those images from the repository to deploy containers. A *registry* is a collection of repositories.
    - AWS offers Amazon Elastic Container Registry (Amazon ECR) as a container image registry that supports integration with other AWS services.
- **Orchestration tool** – You use a container orchestration tool to manage your containerized applications at scale. An orchestration tool manages the scaling, networking, and maintenance of your containers. It helps you manage containers' startup and shutdowns, monitor container health, deploy updates, and manage failover and recovery.
    - Amazon Elastic Kubernetes Service (Amazon EKS) is a managed service that you can use to run the Kubernetes container orchestration software on AWS. You might choose this option if you require additional control over your configurations.
    - Amazon Elastic Container Service (Amazon ECS) is a managed container orchestration service that offers a more managed model for deploying your containers. Amazon ECS features additional integrations with other AWS services.
- **Container hosting** – You must decide which compute resource your orchestration tool will use to host your containers. This resource is often referred to as the container's *launch type*.
    - You can choose Amazon EC2 to launch containers on a variety of instance types. As demand changes, you can scale out and scale in the number of EC2 instances that are used to host containers. This method is cost effective and provides more control over the instance type.
    - AWS Fargate is a serverless hosting service that automatically allocates CPU and memory resources to support your containers. With Fargate, you do not have to provision or manage the underlying compute.

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.



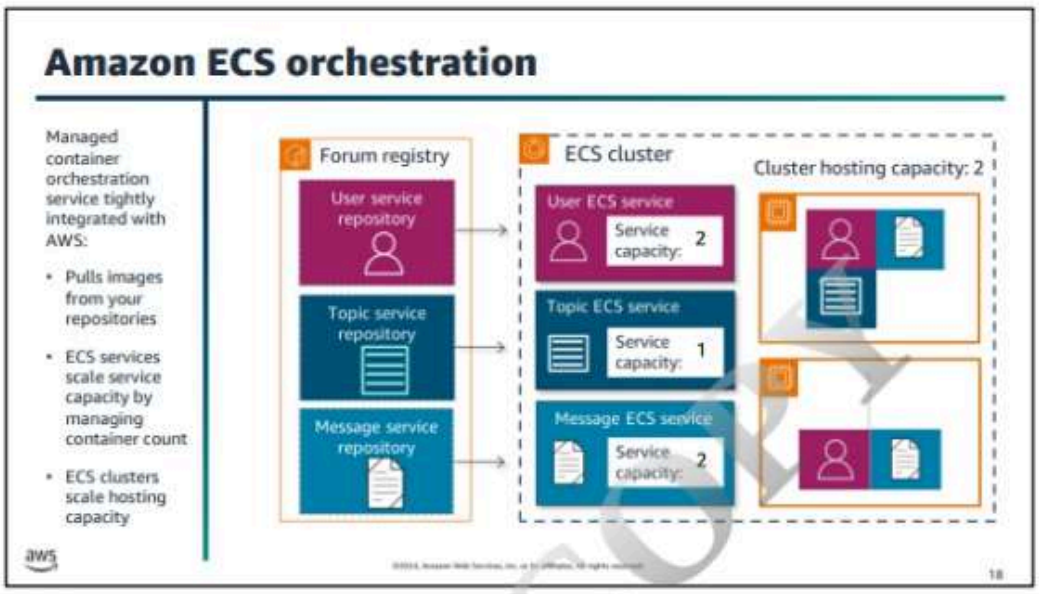
Amazon ECR is a managed Docker container registry. You push your container images to Amazon ECR, and you can then pull those images to launch containers. With Amazon ECR, you can compress, encrypt, and control access to your container images. You also manage versioning and and image tags. An Amazon ECR private registry is provided to each AWS account. You can create one or more repositories in your registry and store images in them.

This example uses containers to refactor a monolithic forum application into microservices. You break apart the code into individual encapsulated services: the user service, the topic service, and the message service. You build container images for each of these services, which you can launch, update, and shut down independently. In this example, the container image for each service is pushed to its own repository, which stores multiple versions of each image. An orchestration service can pull these container images and deploy new containers as needed.

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 ECS is a highly scalable, high-performance container management service that supports Docker containers. Amazon ECS manages the scaling, maintenance, and connectivity for your containerized applications.

With Amazon ECS, you create ECS services, which launch ECS tasks. Amazon ECS tasks can use one or more container images. Amazon ECS services scale your running task count to meet demand on your application.

You create an Amazon ECS cluster with dedicated infrastructure for your application. You can run your tasks and services on a serverless infrastructure that AWS Fargate manages. If you prefer more control over your infrastructure, you can manage your tasks and services on a cluster of EC2 instances. Your cluster can scale EC2 hosting capacity by adding or removing EC2 instances from your cluster.

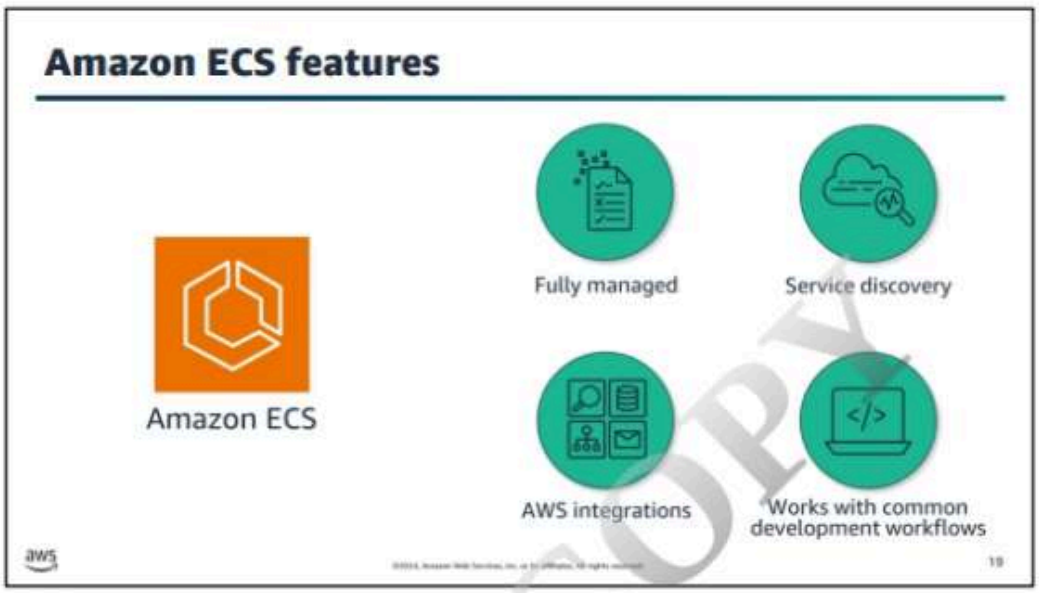
In this example, the containerized forum application is running in Amazon ECS. Three ECS services have been created and added to an ECS cluster: user, topic, and message. The Amazon ECS agent that runs on the compute service pulls container images for each of these services from the forum registry in Amazon ECR. The cluster is using EC2 hosting with a capacity set to two EC2 instances. The user and message services have a task count set to two, and the topic service has a task count of one.

With Amazon ECS, you don't have to operate your own cluster management and configuration management systems, or worry about scaling your management infrastructure.

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 ECS offers the following features:

- **Fully managed** – Because Amazon ECS is a fully managed service, you don't need to manage the control plane, nodes, or add-ons. As a result, teams can focus on building the applications, not the environment.
- **Service discovery** – Amazon ECS features support for service discovery, which you can use to register your ECS services to DNS names. For example, you could register a service called "backend" with a private DNS namespace such as backend.example and a service called "frontend" with frontend.example. You could then configure these services to be able to discover each other within the same virtual private cloud (VPC). With service discovery, your microservice components are automatically discovered and added to namespaces as they're created and shut down.
- **AWS integrations** – Amazon ECS has close integrations with many AWS services, for example, the following:
    - Amazon ECR: Amazon ECS integrates with Amazon ECR, which facilitates your containerized applications' task of accessing and running your container images.
    - AWS Identity and Access Management (IAM): You can assign granular permissions for each of your containers. This allows for a high level of isolation when building your applications.
    - Amazon CloudWatch Logs and Container Insights: You can view the logs from your containerized applications and instances in one convenient location.
- **Flexible hosting options** – With ECS, you can use both Amazon EC2 and serverless hosting with AWS Fargate. You can schedule the placement of your containers across your cluster based on your resource needs, isolation policies, and availability requirements.
- **Development workflows** – Amazon ECS supports continuous integration and continuous deployment (CI/CD). This is a common process for microservice architectures that are based on Docker containers. You can create a CI/CD pipeline that takes the following actions:
    - Monitors changes to a source code repository
    - Builds a new Docker image from that source
    - Pushes the image to an image repository, such as Amazon ECR or Docker Hub
    - Updates your Amazon ECS services to use the new image in your application

For more information about service discovery, see "Use service discovery to connect Amazon ECS services with DNS names" in the *Amazon Elastic Container Service Developer Guide* at

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

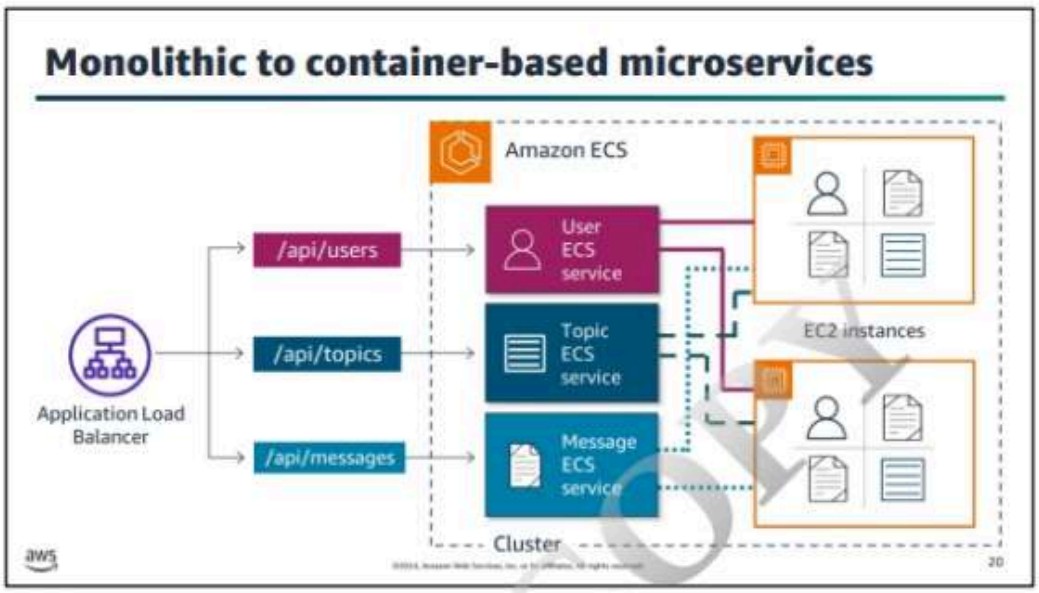
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/service-discovery.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.



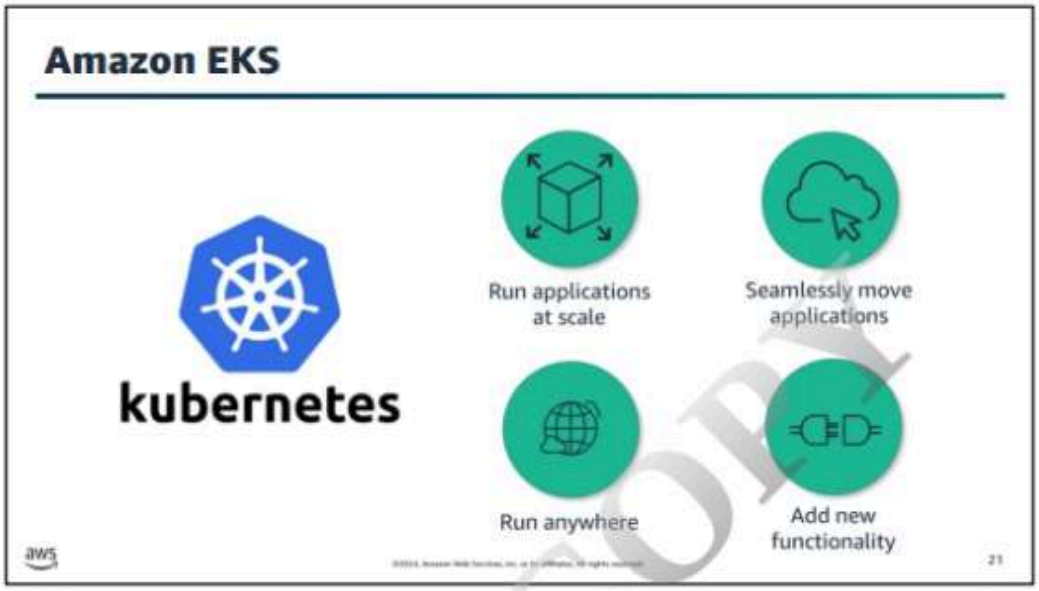
Earlier in this module, you learned the difference between traditional monolithic infrastructures and microservices. Now you can run the microservices on a managed Amazon ECS cluster.

This diagram shows an Application Load Balancer that is sending web traffic based on the path of APIs in the request for each service. You register the user service, topic service, and message service with different target groups. When Amazon ECS starts a task for your service, it registers the container and port combination with the service's target group. The Application Load Balancer routes traffic to and from that container.

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.



Kubernetes is an open-source software that you can use to deploy and manage containerized applications at scale. Kubernetes manages clusters of Amazon EC2 compute instances and runs containers on those instances with processes for deployment, maintenance, and scaling. With Kubernetes, you can run any type of containerized applications by using the same tool set on premises and in the cloud.

Amazon Elastic Kubernetes Service (Amazon EKS) is a certified conformant, managed Kubernetes service. Amazon EKS helps you provide highly available and secure clusters and automates key tasks such as patching, node provisioning, and updates.

With Amazon EKS, you can do the following:

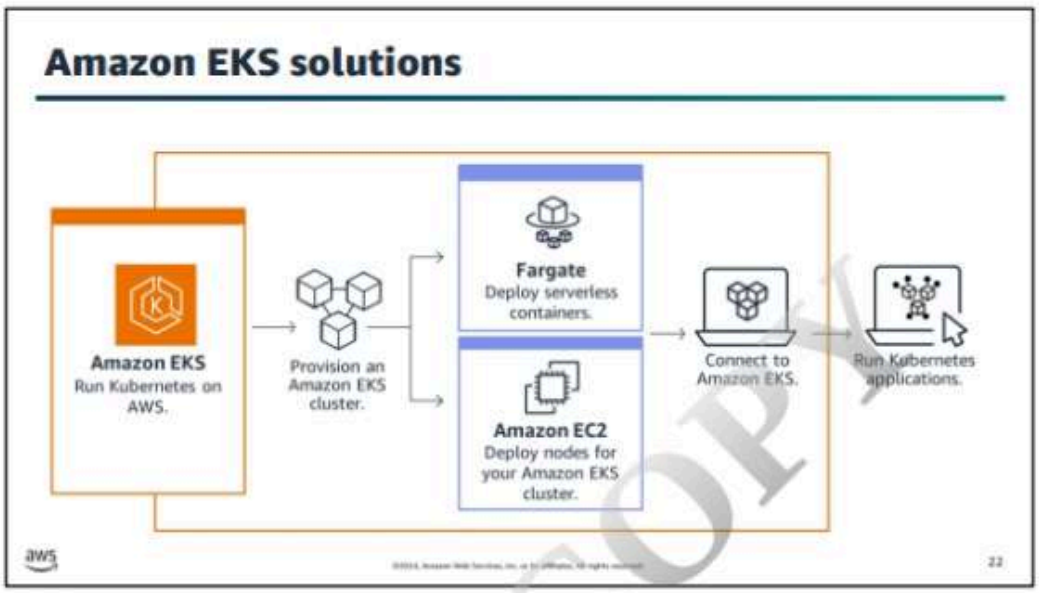
- **Run applications at scale** – Define complex containerized applications and run them at scale across a cluster of servers.
- **Seamlessly move applications** – Move containerized applications from local development to production deployments on the cloud.
- **Run anywhere** – Run highly available and scalable Kubernetes clusters.
- **Add new functionality** – Amazon EKS add-ons allow you to provide supporting operational capabilities to Kubernetes applications. This add-on software is typically built and maintained by the Kubernetes community, cloud providers like AWS or third-party vendors.

For more information, see Kubernetes on AWS at https://aws.amazon.com/kubernetes/.

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.



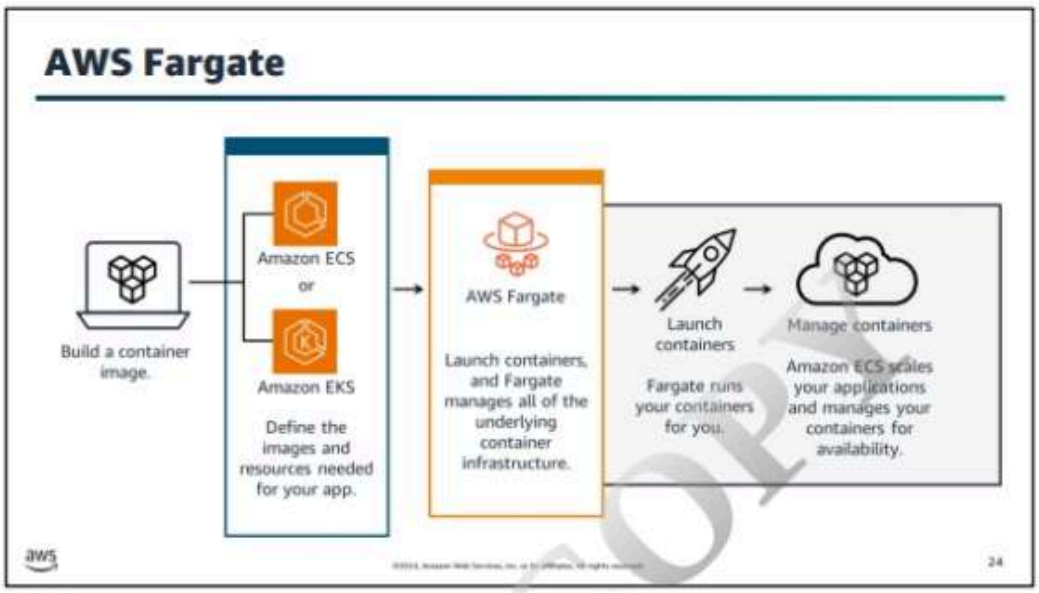
Amazon EKS is a managed service that you can use to run Kubernetes on AWS without having to install and operate your own Kubernetes clusters. With Amazon EKS, AWS manages highly available services and upgrades for you. Amazon EKS runs three Kubernetes managers across three Availability Zones. It detects and replaces unhealthy managers and provides automated version upgrades and patching for the managers. Amazon EKS is also integrated with many AWS services to provide scalability and security for your applications.

Amazon EKS runs the latest version of the open-source Kubernetes software, so you can use all existing plugins and tooling from the Kubernetes community. Applications that run on Amazon EKS are fully compatible with applications that run on any standard Kubernetes environment. This compatibility exists, regardless of whether they are running in on-premises data centers or on public clouds.

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.



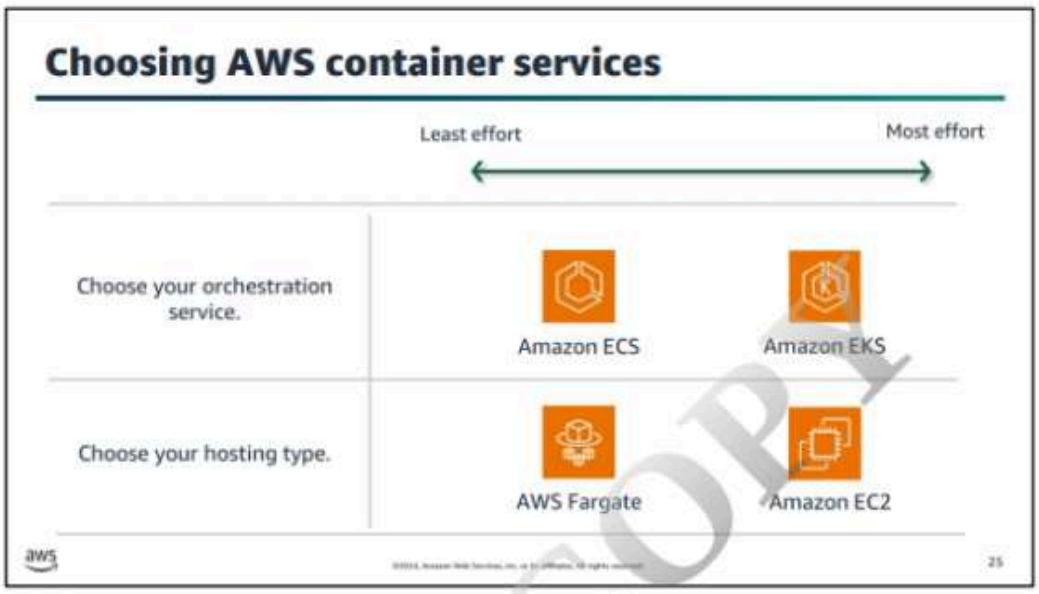
AWS Fargate is a technology for Amazon ECS and Amazon EKS that you can use to run containers without having to manage servers or clusters. With Fargate, you no longer need to provision, configure, and scale clusters of VMs to run containers. Therefore, it removes the need for you to choose server types, decide when to scale your clusters, or optimize cluster packing.

Fargate eliminates the need for you to interact with or think about servers or clusters. With Fargate, you can focus on designing and building your applications instead of managing the infrastructure that runs them.

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.



Deploying your managed container solutions on AWS involves selecting an orchestration tool and a launch type.

Managing your containerized applications

- Amazon ECS provides a more managed solution with less manual configuration and integration with other AWS services.
- Amazon EKS offers the flexibility to start, run, and scale Kubernetes applications in the AWS Cloud or on premises without installing and operating your own Kubernetes clusters. This option is good for organizations that work with open source tools. Amazon EKS requires more configuration, but offers more control over your environment.

Container hosting

- Hosting on Amazon EC2 requires more configuration, but also provides more control over the resources that you use to host your containers. Hosting on Amazon EC2 is also more cost effective.
- Hosting on AWS Fargate uses serverless technology to deliver autonomous container operations, which reduces the time spent on configuration, patching, and security.

For more information about Amazon EKS, see Amazon EKS features at https://aws.amazon.com/eks/features/.

For more information about Amazon ECS, see Amazon Elastic Container Service features at https://aws.amazon.com/ecs/features/.

For more information about Amazon ECS cluster auto scaling, see "Automatically manage Amazon ECS capacity with cluster auto scaling" in the *Amazon Elastic Container Service Developer Guide* at https://docs.aws.amazon.com/AmazonECS/latest/developerguide/cluster-auto-scaling.html.

For more information about AWS Fargate, see AWS Fargate at https://aws.amazon.com/fargate/.

DO NOT COPY javierartiga.espublico@gmail.com

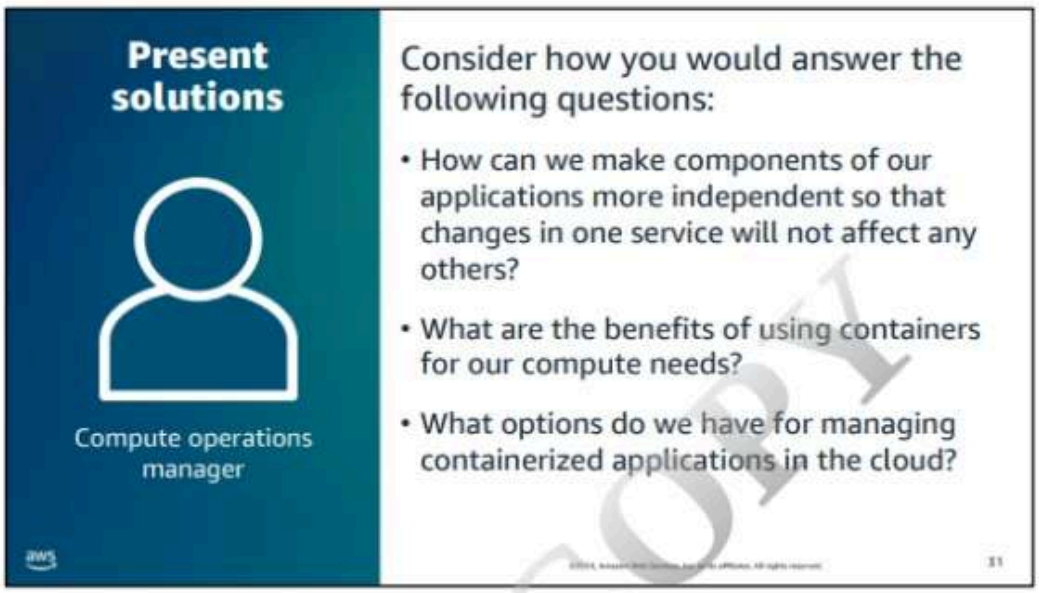
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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 compute 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:

- By using microservice architectures, you can separate your application into component services that you can develop, deploy, operate, and scale without affecting the other services. A failure in one service is isolated to that service.
- Containers provide a standard way to package your application's code, configurations, and dependencies into a single object. This arrangement creates a consistent environment for your applications regardless of the underlying hardware. Containers share an operating system installed on the server, and they run as resource-isolated processes, which offers quick, reliable, and consistent deployments, regardless of the environment.
- You can use Amazon ECR as a repository for your container images. You can use Amazon EKS for container orchestration if you use Kubernetes. You can use Amazon ECS for container orchestration for integrations with other AWS services. You can host your containers on Amazon EC2, or you can choose AWS Fargate to use serverless container hosting.

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:

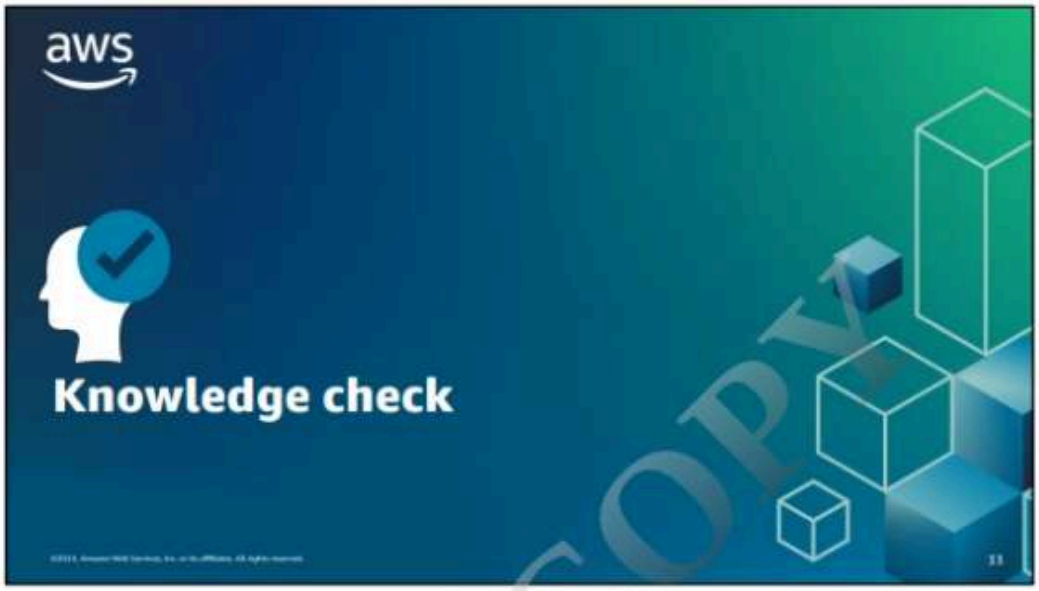
- ✓ Microservices
- ✓ Containers
- ✓ Container services

Next, you review the following:

- 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

© 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 descriptions are characteristics of microservices? (Select TWO.)

Choice	Response
A	Loosely coupled
B	Redundant
C	Autonomous and independent
D	Tightly integrated and dependent
E	Interdependent components

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 descriptions are characteristics of microservices? (Select TWO.)

Choice	Response
A correct	Loosely coupled
B	Redundant
C correct	Autonomous and independent
D	Tightly integrated and dependent
E	Interdependent components

The correct answers are A, loosely coupled, and C, autonomous and independent.

Microservices are loosely coupled. Failures and scaling are automatically handled by the intermediary, such as a load balancer or message queues.

Microservices are autonomous and independent. You can develop, deploy, operate, and scale each component service in a microservices architecture without affecting the other services. Services do not need to share any of their code or implementation with other services. Any communication between individual components happens through well-defined APIs.

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 2

Which descriptions are characteristics of containers? (Select TWO.)

Choice	Response
A	Portable and scalable
B	Require a hypervisor
C	Automatic
D	Repeatable
E	Each requires its own operating 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.

Knowledge check question 2 and answer

Which descriptions are characteristics of containers? (Select TWO.)

Choice	Response
A correct	Portable and scalable
B	Require a hypervisor
C	Automatic
D correct	Repeatable
E	Each requires its own operating system

17

The correct answers are A, portable and scalable, and D, repeatable.

Containers can run on any Linux or Windows system with appropriate kernel-feature support and the container runtime daemon present, which makes them portable. Your laptop, your VM, your Amazon EC2 instance, and your bare metal server are all potential hosts.

Containers are also self-contained environments. Regardless of where you deploy them, the underlying requirements are present and provide uniform behavior.

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

How are containers in Amazon Elastic Container Service (Amazon ECS) logically organized?

Choice	Response
A	In a cluster
B	In pods
C	In Amazon Elastic Block Store (Amazon EBS) volumes
D	In Amazon S3

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 and answer

How are containers in Amazon Elastic Container Service (Amazon ECS) logically organized?

Choice	Response
A correct	In a cluster
B	In pods
C	In Amazon Elastic Block Store (Amazon EBS) volumes
D	In Amazon S3

The correct answer is A, in a cluster.

An Amazon ECS cluster is a logical grouping of tasks or services. Your tasks and services are run on infrastructure that is registered to a cluster.

For more information about ECS clusters, see "Amazon ECS Clusters" in the *Amazon Elastic Container Service Developer Guide* at https://docs.aws.amazon.com/AmazonECS/latest/developerguide/clusters.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.

Knowledge check question 4

Why would developers choose to deploy their containers to AWS Fargate over Amazon EC2?

Choice	Response
A	To take control of their infrastructure
B	To avoid manual infrastructure updates
C	To optimize price for a large load
D	To manage their own patches and updates

40

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 4 and answer

Why would developers choose to deploy their containers to AWS Fargate over Amazon EC2?

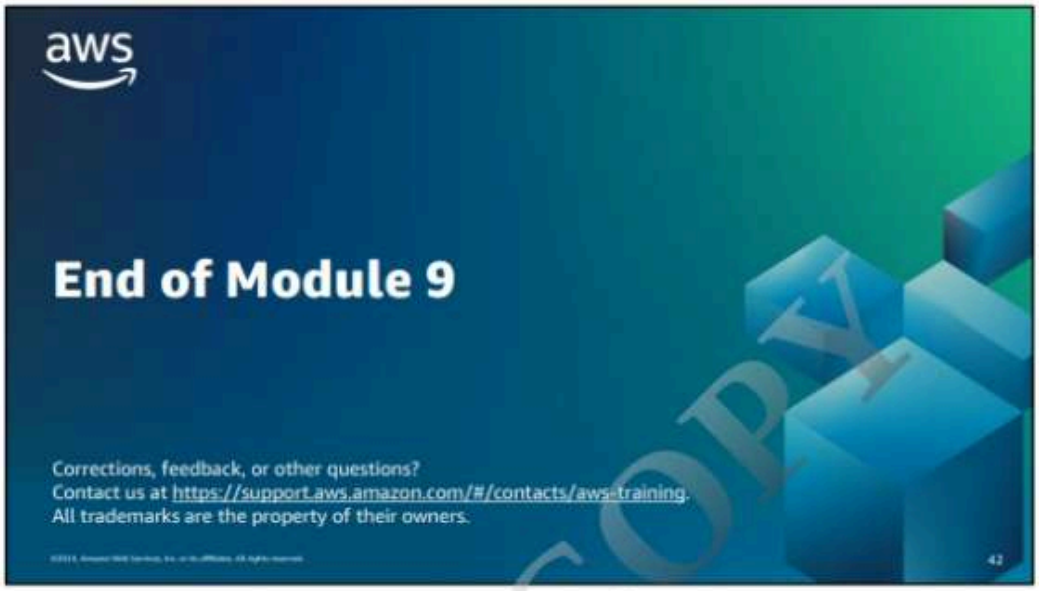
Choice	Response
A	To take control of their infrastructure
B correct	To avoid manual infrastructure updates
C	To optimize price for a large load
D	To manage their own patches and updates

The correct answer is B, to avoid manual infrastructure updates.

With Fargate, you avoid manual infrastructure updates. You don't need to manage servers or clusters of Amazon EC2 instances. With Fargate, you no longer provision, configure, or scale clusters of virtual machines to run containers.

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.

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

How many virtual private clouds (VPCs) does your organization use?

A. <20

B. 20–100

C. >100

D. I'm not sure

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.

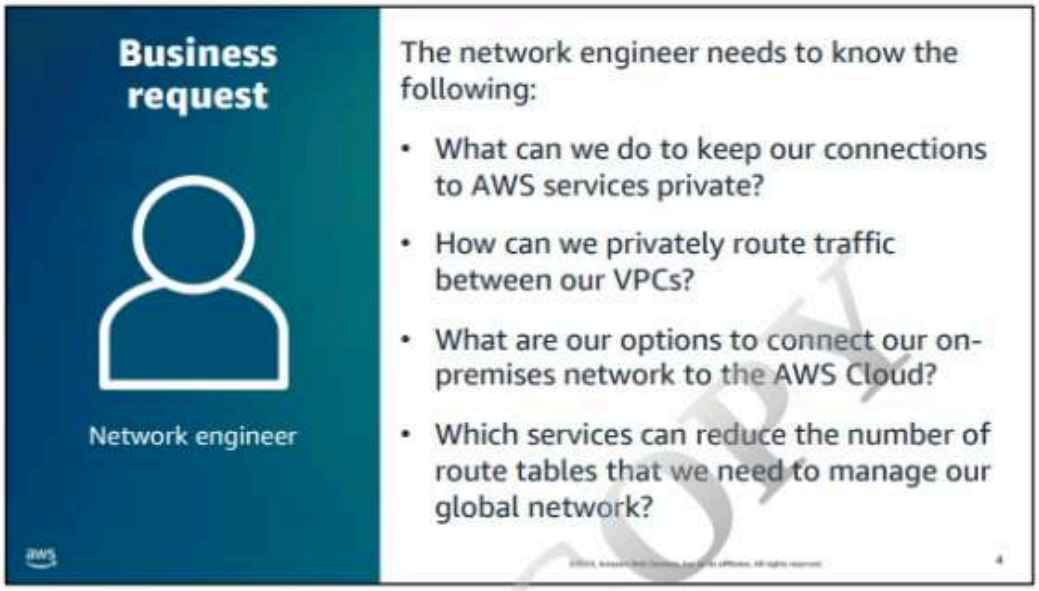
Module overview

- Business request
- VPC endpoints
- VPC peering
- Hybrid networking
- AWS Transit Gateway
- Present solutions
- Knowledge check

1

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 network engineer meets with you to discuss how to connect multiple networks together. They also want to set up a hybrid environment. Here are some questions that they are asking you about changes to Amazon Virtual Private Cloud (Amazon VPC) networking.

At the end of this module, you meet with the network engineer and present some solutions.

DO NOT COPY
javierartiga.espublico@gmail.com

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

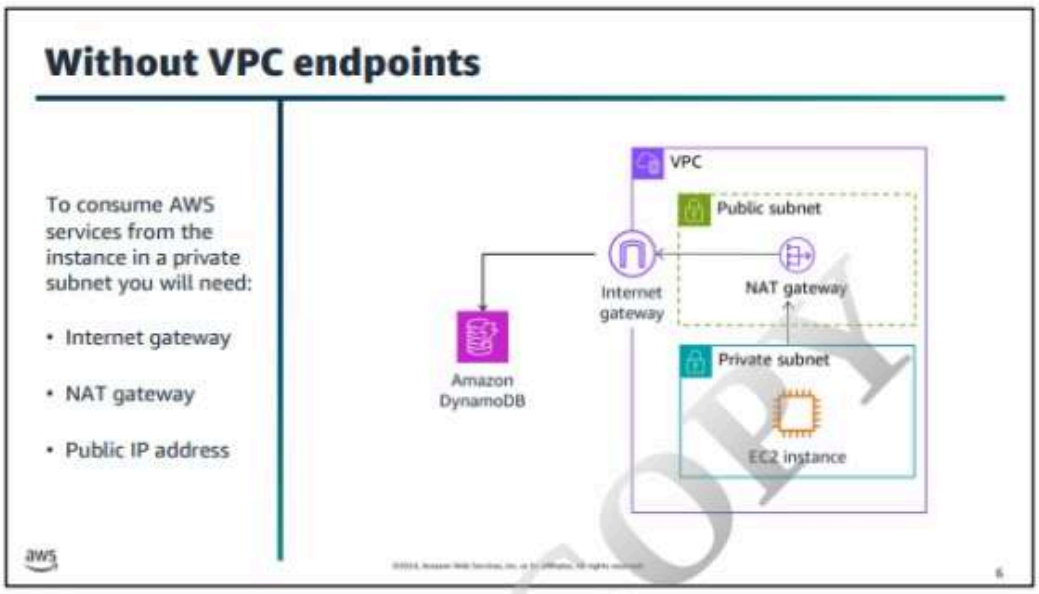
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

The network engineer asks, "What can we do to keep our connections to AWS services private?" The networking team must build paths to protect traffic to and from Amazon Web Services (AWS) services. Examples of these services include Amazon Simple Storage Service (Amazon S3) and AWS Systems Manager.

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.



Without VPC endpoints, a VPC requires an internet gateway and a NAT gateway, or a public IP address, to access serverless services outside the VPC.

A VPC endpoint provides a reliable path between your VPC and supported AWS services. You do not need an internet gateway, a NAT device, a virtual private network (VPN) connection, or an AWS Direct Connect connection. Amazon Elastic Compute Cloud (Amazon EC2) instances in your VPC do not require public IP addresses to communicate with resources in the service.

Endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components. Endpoints permit communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.

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.

Interface VPC endpoints

- Supports access to 200+ AWS services
- Is an elastic network interface with a private IP address
- Horizontally scaled
- Redundant
- Highly available
- Powered by AWS PrivateLink

An *interface VPC endpoint* (interface endpoint) is an elastic network interface with a private IP address from the IP address range of your subnet. The network interface serves as an entry point for traffic that is destined to a supported service. AWS PrivateLink powers interface endpoints, and it avoids exposing traffic to the public internet.

This course does not cover gateway VPC endpoints. For more information about gateway VPC endpoints, see "Gateway endpoints" at https://docs.aws.amazon.com/vpc/latest/privatelink/gateway-endpoints.html.

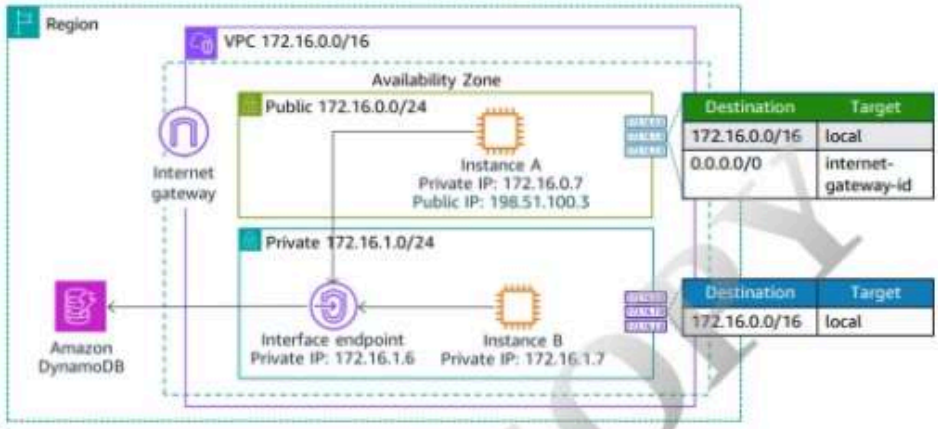
This course does not cover Gateway Load Balancer endpoints. For more information about Gateway Load Balancer endpoints, see "Module 10: Networking 2" in the *Architecting on AWS – Online Course Supplement* at https://explore.skillbuilder.aws/learn/course/external/view/elearning/8319/architecting-on-aws-online-course-supplement.

For more information about all services that integrate with interface endpoints, see "What is AWS PrivateLink?" at https://docs.aws.amazon.com/vpc/latest/privatelink/what-is-privatelink.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.

Example interface VPC endpoints



With an interface VPC endpoint (interface endpoint), you can privately connect your VPC to services as if they were in your VPC. When the interface endpoint is created, traffic is directed to the new endpoint without changes to any route tables in your VPC.

In this example, a Region is shown with DynamoDB outside the example VPC. The example VPC has a public and private subnet with an Amazon EC2 instance in each. DynamoDB traffic that is directed to dynamodb.*region*.amazonaws.com is sent to an elastic network interface in the private subnet.

For more information about interface VPC endpoints, see "Access an AWS service using an interface VPC endpoint" at https://docs.aws.amazon.com/vpc/latest/privatelink/vpce-interface.html.

You can see a full list of AWS services that support interface endpoints. For more information, see "AWS services that integrate with AWS PrivateLink" at https://docs.aws.amazon.com/vpc/latest/privatelink/aws-services-privatelink-support.html.

DO NOT COPY javierartiga.espublico@gmail.com

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

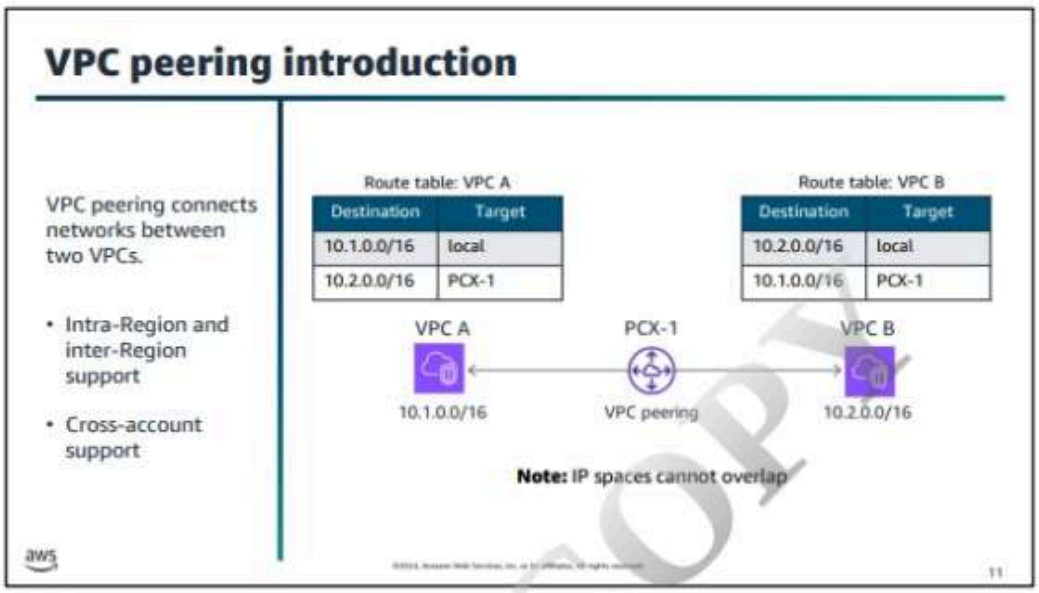
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

The network engineer asks, "How can we privately route traffic between our VPCs?" The networking team team is considering options for how to network across VPCs both in a single account and across multiple accounts and AWS Regions.

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 business or architecture becomes large enough, you will need to separate logical elements for security or architectural needs, or just for simplicity. A VPC peering connection is a one-to-one relationship between two VPCs. You can have only one peering resource between any two VPCs. You can create multiple VPC peering connections for each VPC that you own.

VPC peering limitations and rules include the following:

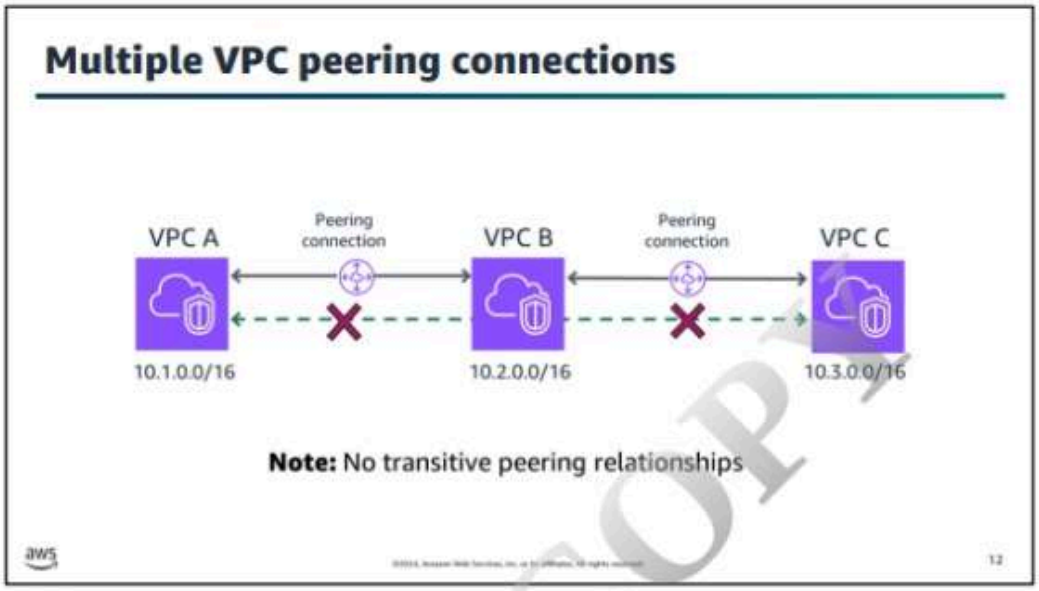
- There is a limit on the number of active and pending VPC peering connections that you can have per VPC.
- You can have only one VPC peering connection between the same two VPCs.
- The maximum transmission unit (MTU) across a VPC peering connection is 1,500 bytes.

In the diagram, VPC A and VPC B are peered. The route table for each VPC has a route with the Classless Inter-Domain Routing (CIDR) range of the opposite VPC targeting the peering connection ID. In the diagram, the peering ID is PCX-1. Local traffic stays within each 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.



In this diagram, VPCs A and B are peered, and B and C are peered. This peering setup does not mean that A can communicate with C. By default, VPC peering does not permit VPC A to connect to VPC C unless they are explicitly established as peers. You control which VPCs can communicate with each other.

You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported. You will not have any peering relationship with VPCs that your VPC is not directly peered with. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account within a single Region.

For more information about VPC peering limits, see "Amazon VPC quotas" in the *Amazon Virtual Private Cloud User Guide* at https://docs.aws.amazon.com/vpc/latest/userguide/amazon-vpc-limits.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.

Benefits of VPC peering

- Bypasses the internet gateway or virtual private gateway
- Provides highly available connections—no single point of failure
- Avoids bandwidth bottlenecks
- Uses private IP addresses to direct traffic between VPCs

11

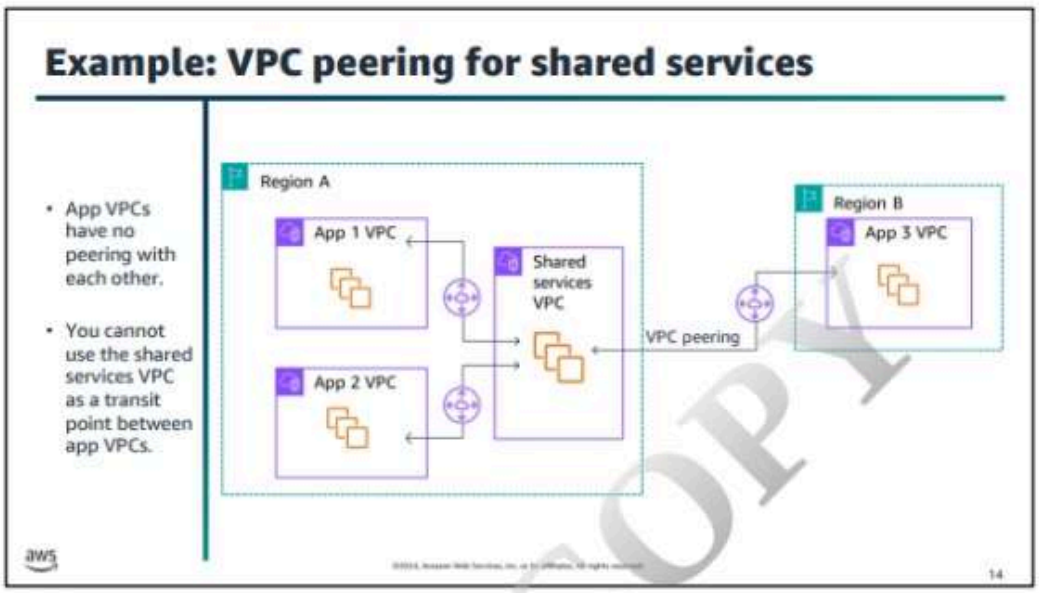
Some benefits of using VPC peering to connect multiple VPCs together includes the following:

- Bypass the internet gateway or virtual private gateway. Use VPC peering to quickly connect two or more of your networks without needing other virtual appliances in your environment.
- Use highly available connections. VPC peering connections are redundant by default. AWS manages your connection.
- Avoid bandwidth bottlenecks. All inter-Region traffic is encrypted with no single point of failure or bandwidth bottlenecks. Traffic always stays on the global AWS backbone, and never traverses the public internet. This arrangement reduces threats, such as common exploits and distributed denial of service (DDoS) attacks.
- Use private IP addresses to direct traffic. The VPC peering traffic remains in the private IP space.

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.



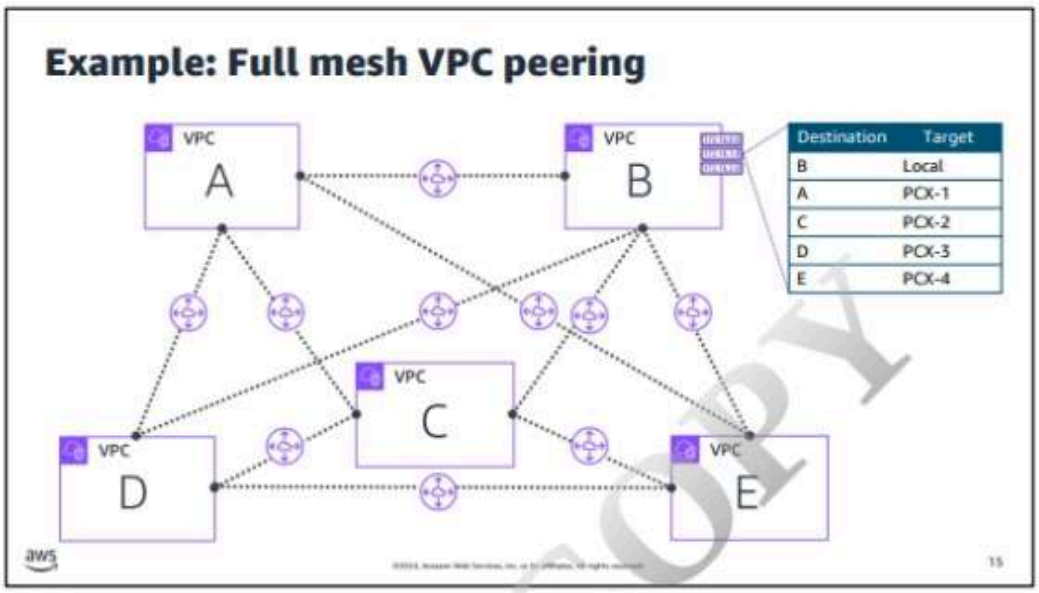
In this example, your security team provides you with a *shared services VPC* that each department can peer with. This VPC allows your resources to connect to a shared directory service, security scanning tools, monitoring or logging tools, and other services.

A VPC peering connection with a VPC in a different Region is present. Inter-Region VPC peering allows VPC resources that run in different AWS Regions to communicate with each other by using private IP addresses. You won't be required to use gateways, VPN connections, or separate physical hardware to send traffic between your Regions.

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 create a full mesh network design by using VPC peering to connect each VPC to every other VPC in the organization. In this architecture, each VPC must have a one-to-one connection with each VPC that it is approved to communicate with. This requirement is because each VPC peering connection is nontransitive in nature. It does not permit network traffic to pass from one peering connection to another.

For example, VPC A is peered with VPC C, and VPC C is peered with VPC E. You cannot route packets from VPC A to VPC E through VPC C. To route packets directly between VPC A and VPC E, you must create a separate VPC peering connection between them.

The number of connections that are required has a direct impact on the number of potential points of failure and the requirement for monitoring. The fewer connections that you need, the fewer that you need to monitor and the fewer potential points of failure. You should consider another option as your networking needs scale up. You learn about an alternative solution later in this module.

DO NOT COPY javierartiga.espublico@gmail.com

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

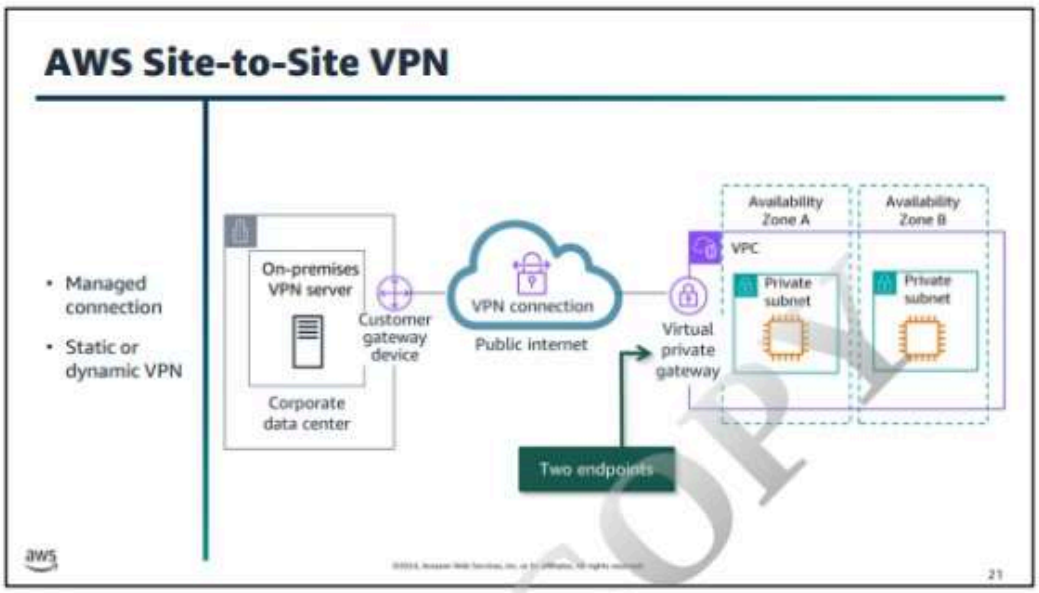
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

The network engineer asks, "What are our options to connect our on-premises network to the AWS Cloud?" The networking team has a requirement to create a hybrid environment that combines their on-premises data centers and their VPCs.

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 AWS Site-to-Site VPN connection offers two VPN tunnels. These tunnels go between a virtual private gateway (or a transit gateway) on the AWS side, and a customer gateway on the on-premises side.

- A *virtual private gateway* is the VPN concentrator on the AWS side of the AWS Site-to-Site VPN connection.
- The VPN tunnels per one VPN connection terminate in different Availability Zones.
- A *customer gateway* is a resource that you create in AWS. It represents the customer gateway device in your on-premises network. Your network administrator configures the customer gateway device or application in your remote network. AWS provides you with the required configuration information.
- You choose either static routing or dynamic routing based on the features of your customer gateway device. The dynamic routing option uses Border Gateway Protocol (BGP) to automatically discover routes.

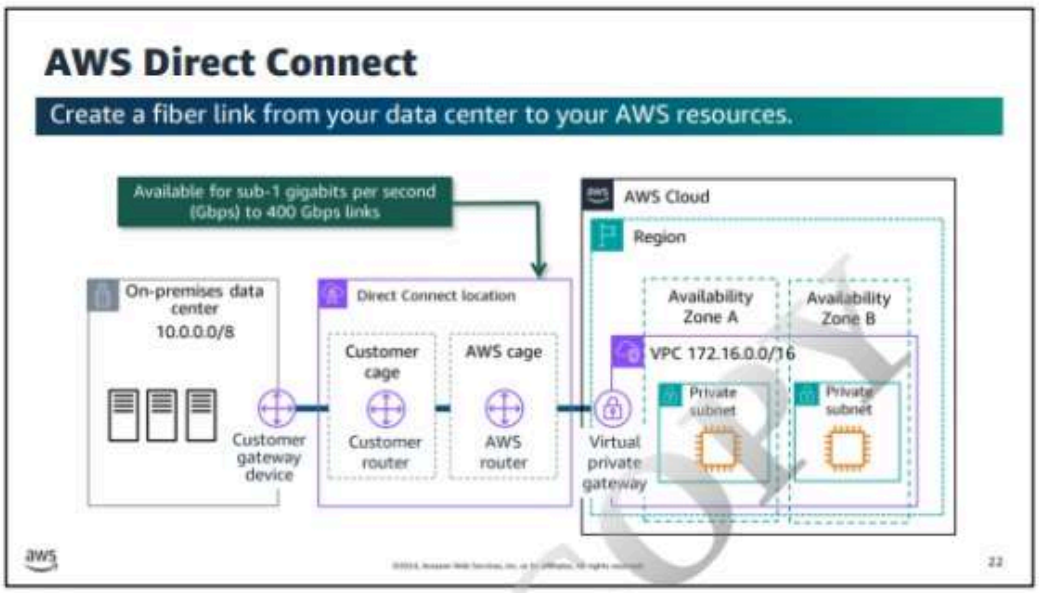
Your customer gateway device must bring up the tunnels for your AWS Site-to-Site VPN connection by generating traffic and initiating the Internet Key Exchange (IKE) negotiation process. When you create a customer gateway, you provide information about your device to AWS.

For more information, see "What is AWS Site-to-Site VPN?" in the *AWS Site-to-Site VPN User Guide* at https://docs.aws.amazon.com/vpn/latest/s2svpn/VPC_VPN.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 Direct Connect links your internal network to a Direct Connect location over a standard Ethernet fiber-optic cable. One end of the cable is connected to your router, the other to a Direct Connect router. This connection is called the *cross-connect*. With this connection, you can create *virtual interfaces* directly to public AWS services (for example, to Amazon S3) or to Amazon VPC, bypassing internet service providers (ISPs) in your network path.

A Letter of Authorization and Connecting Facility Assignment (LOA-CFA) is required to begin the process of creating the cross-connect in the data center.

In the example, an on-premises data center holds your customer gateway device. In the data center, traffic is passed to your customer cage that is holding a router. It brings your traffic to an AWS router in an AWS cage. In the AWS Cloud, a virtual private gateway receives traffic over the AWS backbone, connecting the on-premises data center to the VPC. You can then create routes in your VPC to allow traffic to flow between your on-premises data center and the private subnets in your VPC.

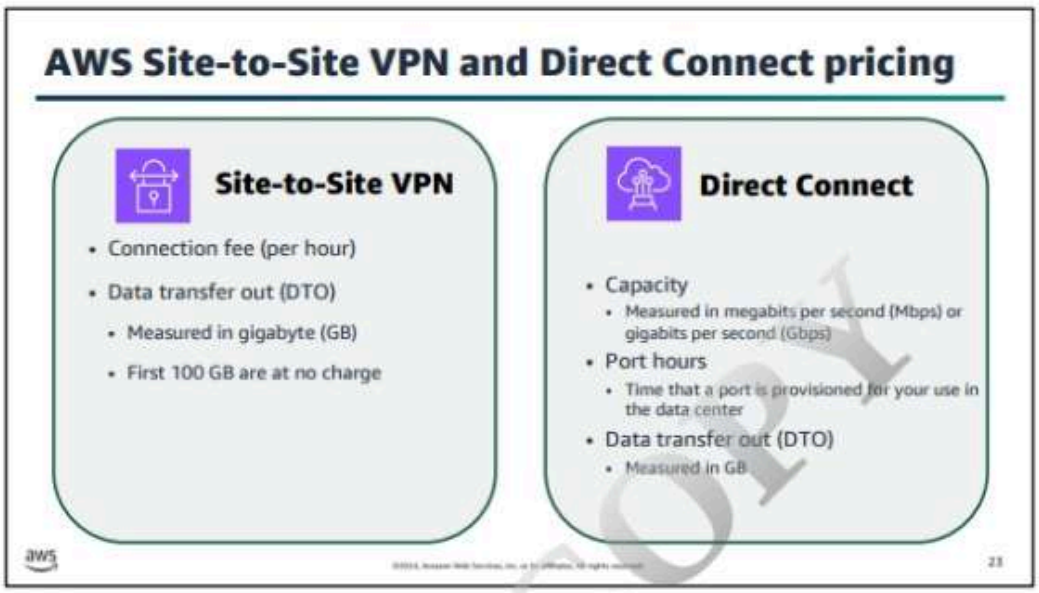
A Direct Connect location provides access to AWS in the Region with which it is associated. You can use a single connection in a public Region or AWS GovCloud (US) to access public AWS services in all other public Regions.

To use Direct Connect in a Direct Connect location, your network must meet one of the following conditions:
- Your network is collocated with an existing Direct Connect location.
- You are working with a Direct Connect Partner.
- You are working with an independent service provider to connect to Direct Connect.

For more information about AWS Direct Connect connections, see AWS Direct Connect at https://aws.amazon.com/directconnect/.

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.



It is important for you to consider pricing as a factor when deciding whether to use AWS Site-to-Site VPN or AWS Direct Connect.

With AWS Site-to-Site VPN, you are charged a per-hour connection fee for your use. Similarly to Direct Connect, you are also charged for data transfer out (DTO), which is explained later in the student notes. With AWS Site-to-Site VPN, you receive your first 100 GB of data transfer out at no charge.

Direct Connect uses the following pricing factors:

- *Capacity* is the maximum rate that data can be transferred through a network connection. The capacity of AWS Direct Connect connections are measured in megabits per second (Mbps) or gigabits per second (Gbps).
- *Port hours* measure the time that a port is provisioned for your use with AWS. Or it can be with an AWS Direct Connect Delivery Partner's networking equipment inside an AWS Direct Connect location. Even when no data is passing through the port, you are charged for port hours. Port hour pricing is determined by the connection type: dedicated or hosted.
- *Data transfer out (DTO)* refers to the cumulative network traffic that is sent through AWS Direct Connect to destinations outside AWS. DTO is charged per GB, and unlike capacity measurements, DTO refers to the amount of data transferred, not the speed. When calculating DTO, exact pricing depends on the AWS Region and AWS Direct Connect location that you are using.

For more information about AWS Site-to-Site VPN pricing, see AWS VPN pricing at https://aws.amazon.com/vpn/pricing/.

For more information about Direct Connect pricing, see AWS Direct Connect pricing at https://aws.amazon.com/directconnect/pricing/.

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.

Choosing AWS VPN or Direct Connect

AWS Site-to-Site VPN	Direct Connect
Limited to 1.25 Gbps connection maximum	Sub-1, 1, 10, 100 or 400 Gbps connection options
Faster to configure than Direct Connect	Requires special agreements and physical cabling to the data center
Don't need to pay for inactive connections	Pay for port hours whether the connection is active or not
Encrypted in transit by default, but travels over public internet	Not encrypted by default, but a private, dedicated connection

24

You should choose the product that best meets your hybrid connectivity needs. You can choose to use either Site-to-Site VPN, Direct Connect, or both, depending on your use case.

Choose AWS VPN solutions for the following:

- When you need a way to quickly establish a network connection between your on-premises networks and your VPC
- When you need to stay within a small budget
- When you require encryption in transit

Consider Direct Connect for the following:

- When you need faster connectivity options than what AWS Site-to-Site VPN can provide
- When you are already in a collocation that supports Direct Connect
- When you need predictable network performance

DO NOT COPY javierartiga.espublico@gmail.com

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

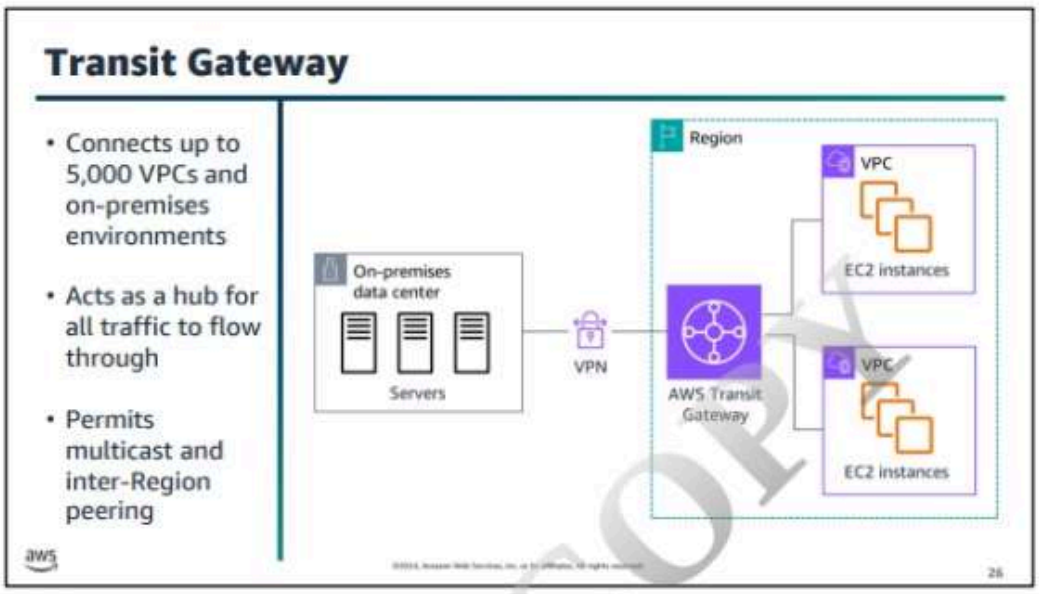
Printed by: javierartiga.espublico@gmail.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

The network engineer asks, "Which services can reduce the number of route tables that we need to manage our global network?" The networking team must scale out the hybrid network in a way that reduces the number of route tables and connections that they must manage.

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 Transit Gateway acts as a hub that controls how traffic is routed among all the connected networks, which act like spokes. This hub-and-spoke model significantly simplifies management and reduces operational costs because each network only has to connect to Transit Gateway, not to every other network. Any new VPC is connected to Transit Gateway and is then automatically available to every other connected network.

Routing through a transit gateway operates at Layer 3, where the packets are sent to a specific next-hop attachment based on their destination IP addresses. Your transit gateway routes Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) packets between attachments using transit gateway route tables. Configure route tables to propagate routes from the tables for the attached VPCs and VPN connections. You can add static routes to the transit gateway route tables.

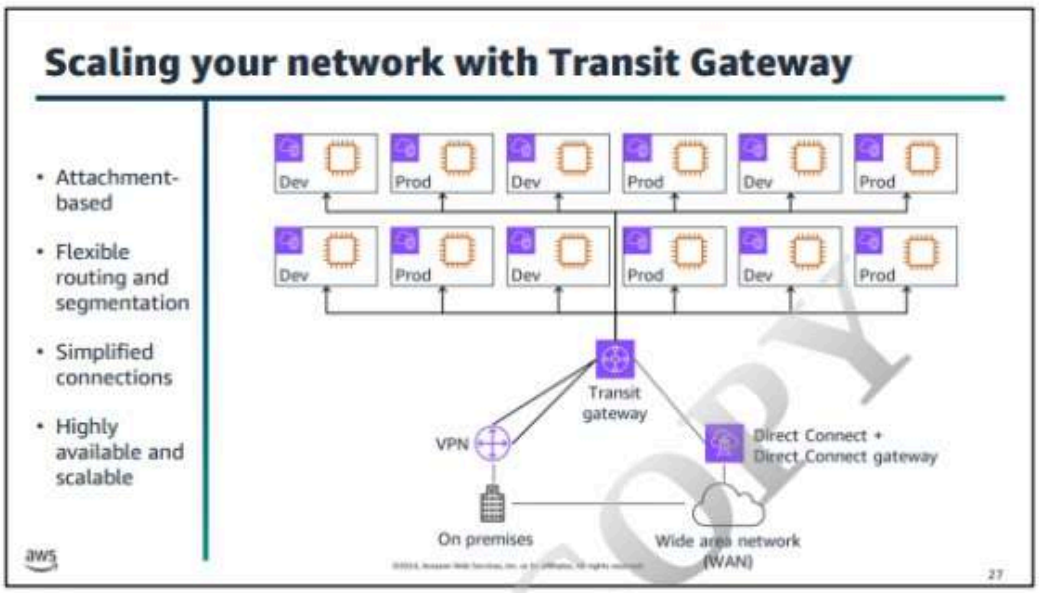
A *transit gateway* is a network transit hub that you can use to interconnect your VPCs and on-premises networks, and it scales elastically, based on traffic.

For more information about AWS Transit Gateway, see AWS Transit Gateway at https://aws.amazon.com/transit-gateway/.

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 transit gateway acts as a cloud router to simplify your network architecture. As your network grows, the complexity of managing incremental connections doesn't slow you down. When building global applications, you can connect transit gateways by using inter-Region peering.

With Transit Gateway Network Manager, you can monitor your VPCs and edge connections from a central console. Integrated with popular software-defined wide area network (SD-WAN) devices, Transit Gateway Network Manager helps you identify issues in your global network.

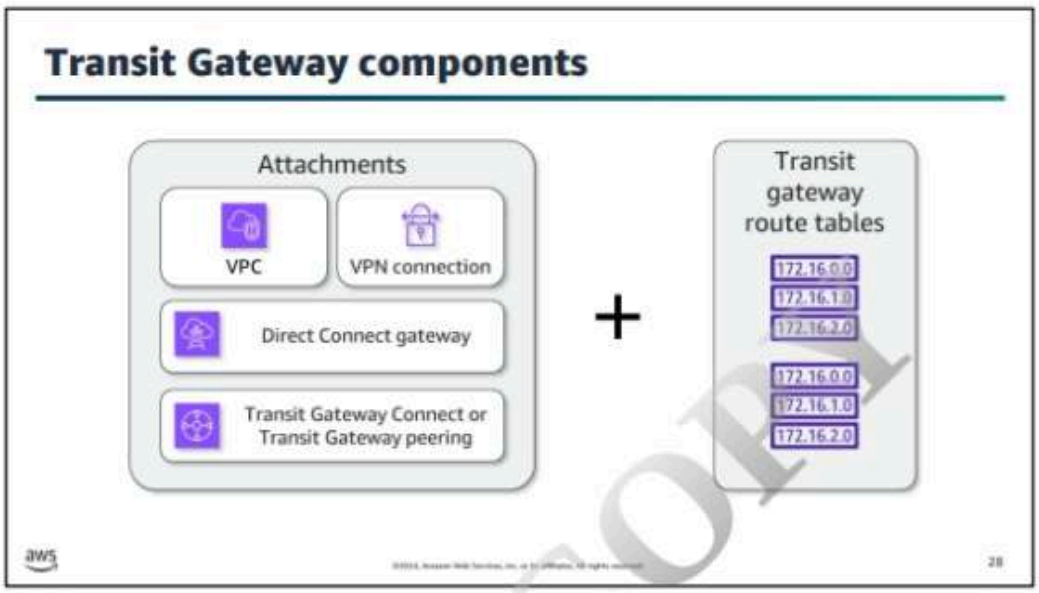
Traffic between a VPC and transit gateway remains on the AWS global private network and is not exposed to the public internet. Transit Gateway inter-Region peering encrypts all traffic. With no single point of failure or bandwidth bottleneck, it protects you against DDoS attacks and other common exploits.

[**Image description:** On premises is connected through their WAN to Direct Connect and Direct Connect gateway VPC, which are then connected to a transit gateway. As a backup, on premises is also connected to the transit gateway using a VPN. Then transit gateway is connected to 12 VPCs made up of development and production environments. This establishes the connectivity between on premises and those 12 VPCs. **End description.**]

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.



Transit Gateway is made up of two important components: attachments and route tables.

A transit gateway *attachment* is a source and a destination of packets. You can attach one or more of the following resources if they are in the same Region as the transit gateway:

- VPC
- VPN connection
- Direct Connect gateway
- Transit Gateway Connect
- Transit Gateway peering connection

Transit Gateway Connect establishes a connection between a transit gateway and third-party virtual appliances running in a VPC.

You can use VPN connections and Direct Connect gateways to connect your on-premises data centers to transit gateways. With a transit gateway, you can connect with VPCs in the AWS Cloud, which creates a hybrid network.

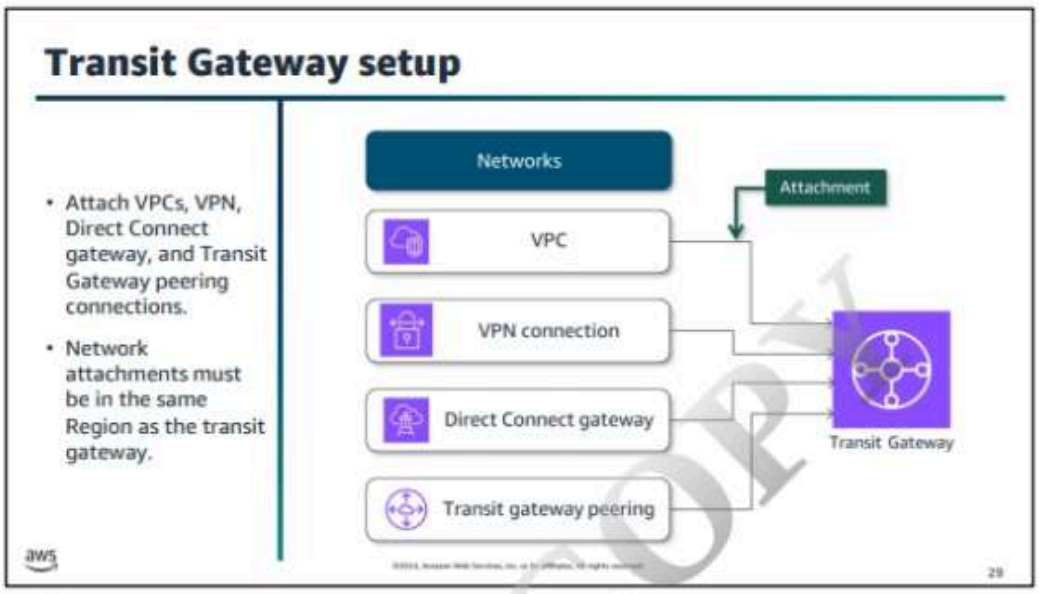
A transit gateway has a default route table and can optionally have additional route tables. A route table includes dynamic and static routes that decide the next hop based on the destination IP address of the packet. The target of these routes could be any transit gateway attachment. By default, transit gateway attachments are associated with the default transit gateway route table.

Each attachment is *associated* with exactly one route table. Each route table can be associated with zero to many attachments.

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 transit gateway works across AWS accounts. You can use AWS Resource Access Manager to share your transit gateway with other accounts. After you share a transit gateway with another AWS account, the account owner can attach their VPCs to your transit gateway. A user from either account can delete the attachment at any time.

A transit gateway attachment is a source and a destination of packets. You can attach the following resources to your transit gateway:

- One or more VPCs
- One or more VPN connections
- One or more Direct Connect gateways
- One or more transit gateway peering connections

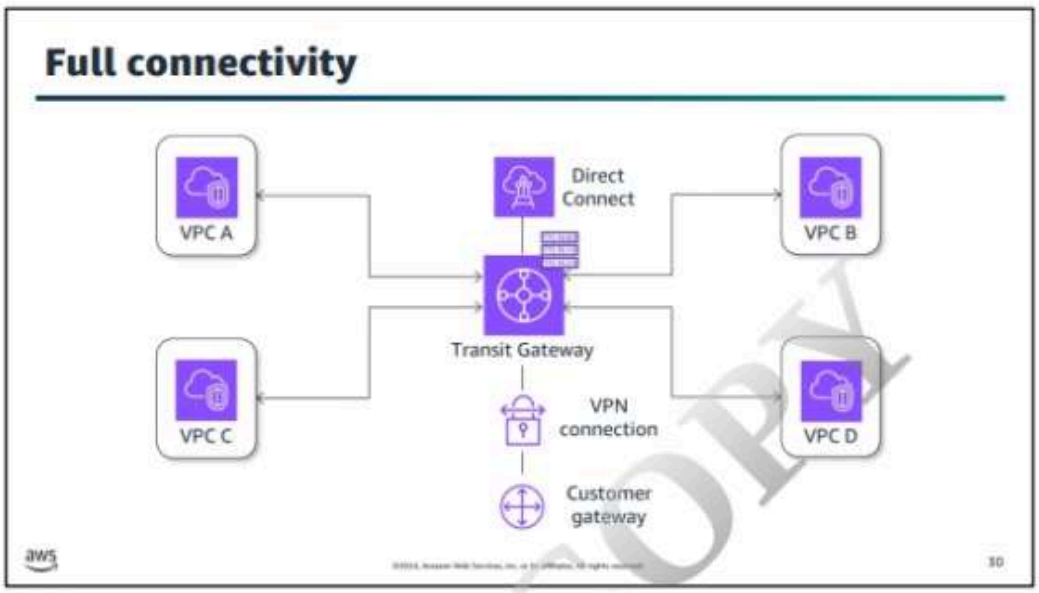
You can peer both intra-Region and inter-Region transit gateways, and route traffic between them. Transit gateways support dynamic and static routing between attached VPCs and VPN connections. You can turn on or turn off route propagation for each attachment.

For more information about transit gateway setup, see "Example transit gateway scenarios" in the *Amazon VPC AWS Transit Gateway* guide at https://docs.aws.amazon.com/vpc/latest/tgw/TGW_Scenarios.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.



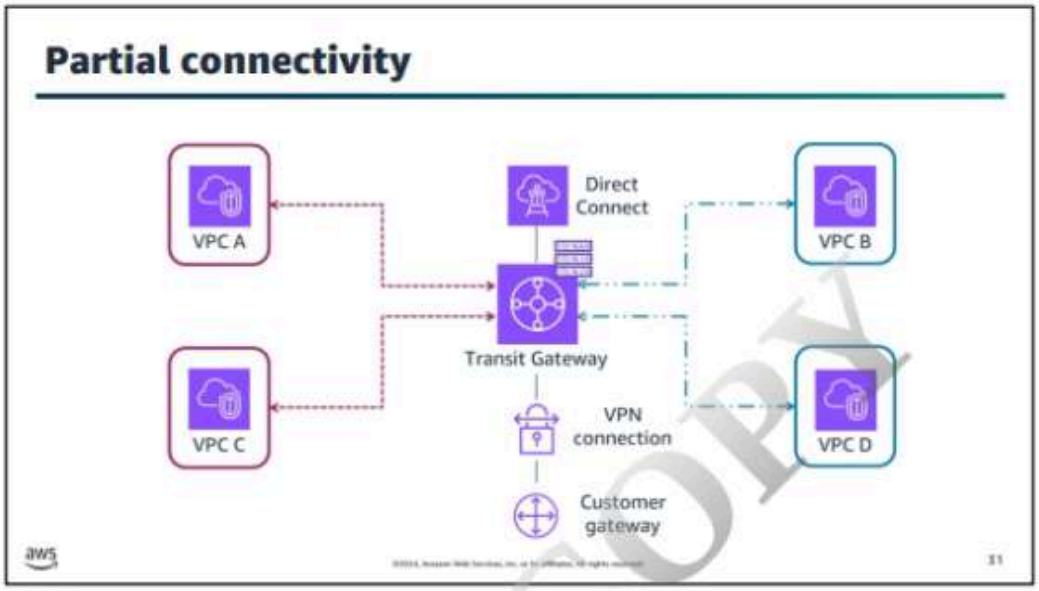
Transit Gateway is the central hub that helps you control communication between attached resources.

This diagram shows four VPCs (VPCs A, B, C, and D) with attachments to the transit gateway. A Direct Connect gateway and a VPN connection are also attached to the same transit gateway. A customer gateway device is on the other side of the VPN connection.

In this diagram, all of the VPCs can communicate with each other.

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.



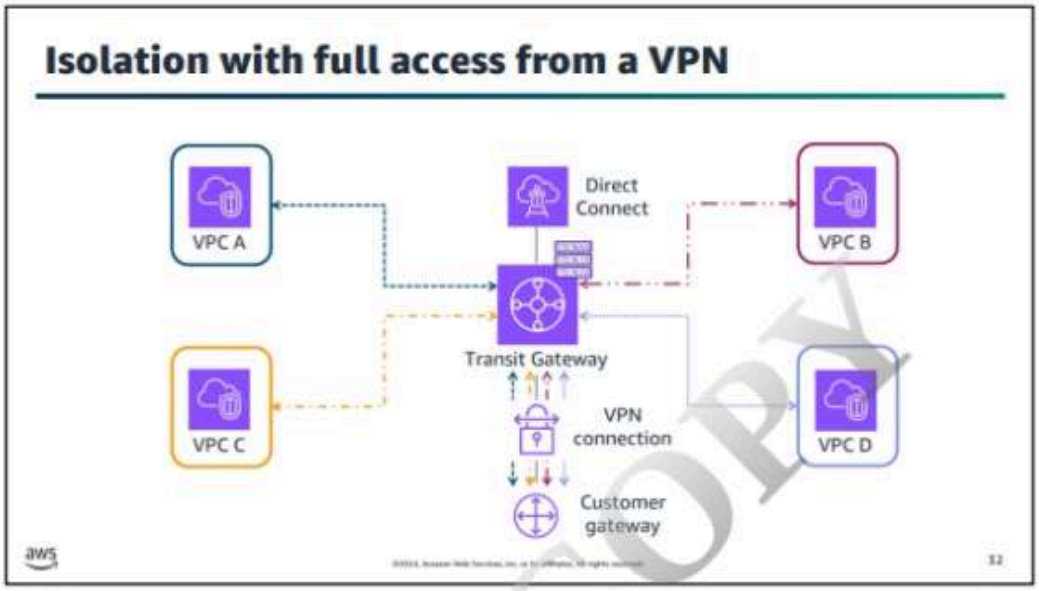
In this diagram, VPC A and VPC C can communicate with each other, but not with VPC B or VPC D.

VPC B and VPC D can communicate with each other, but not with VPC A or VPC C.

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 this diagram, none of the VPCs can communicate with each other, but they can all be accessed through the VPN connection.

DO NOT COPY
javierartiga.espublico@gmail.com

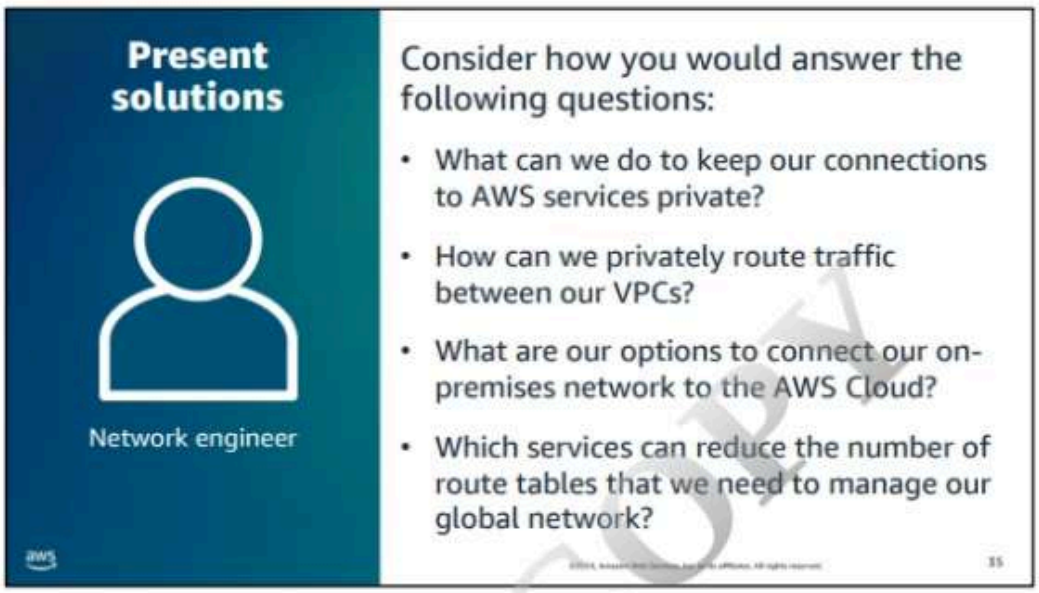
© 2025, Amazon Web Services, Inc. or its affiliates. All rights reserved.

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 network engineer 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:

- You can keep connections to AWS services private by using interface VPC endpoints. You can also create gateway endpoints to allow connectivity in your VPCs to Amazon S3 or DynamoDB without using an internet gateway or a NAT gateway.
- Multiple solutions exist that will privately route traffic between VPCs. By using VPC peering, you can quickly network two VPCs together, but it is difficult to scale. Consider using a transit gateway if you will be routing traffic between many VPCs.
- Two options to connect on-premises networks to VPCs include AWS Site-to-Site VPN and AWS Direct Connect. Choose your solution based on your use case to optimize cost and performance.
- Transit Gateway route tables can be managed and shared between multiple VPCs in your global network. You can reduce the number of route tables that you create and manage by associating existing route tables to new VPCs when appropriate.

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:

- ✓ VPC endpoints
- ✓ VPC peering
- ✓ Hybrid networking
- ✓ Transit Gateway

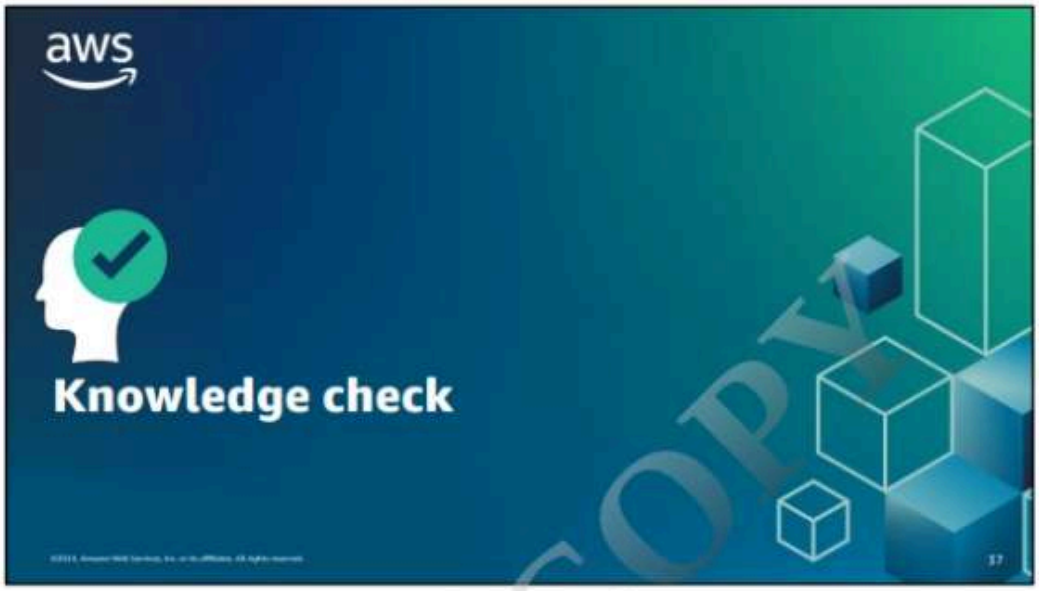
Next, you review the following:

Knowledge check

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.

Knowledge check question 1

What is a connection to a transit gateway called?

Choice	Response
A	VPN
B	Attachment
C	Route
D	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.

Knowledge check question 1 and answer

What is a connection to a transit gateway called?

Choice	Response
A	VPN
B correct	Attachment
C	Route
D	VPC

The answer is B, attachment.

A transit gateway attachment is both a source and a destination of packets. You can attach the following resources to your transit gateway:

- One or more VPCs
- One or more VPN connections
- One or more AWS Direct Connect gateways
- One or more Transit Gateway Connect attachments
- One or more transit gateway peering connections

If you attach a transit gateway peering connection, the transit gateway must be in a different Region.

For more information about transit gateways and attachments, see "How transit gateways work" in the *Amazon VPC AWS Transit Gateway* guide at https://docs.aws.amazon.com/vpc/latest/tgw/how-transit-gateways-work.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.

Knowledge check question 2

What are the components of an AWS Site-to-Site VPN connection? (Select TWO.)

Choice	Response
A	Customer gateway device
B	Interface endpoint
C	Virtual private gateway
D	VPC peering connection
E	Gateway endpoint

40

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

What are the components of an AWS Site-to-Site VPN connection? (Select TWO.)

Choice	Response
A correct	Customer gateway device
B	Interface endpoint
C correct	Virtual private gateway
D	VPC peering connection
E	Gateway endpoint

The answers are A and C, customer gateway device and virtual private gateway.

A customer gateway device is a physical device or software application on your side of the Site-to-Site VPN connection.

A virtual private gateway is the VPN concentrator on the Amazon side of the AWS Site-to-Site VPN connection. Use a virtual private gateway or a transit gateway as the gateway for the Amazon side of the Site-to-Site VPN connection.

For more information about AWS Site-to-Site VPN, see "What is AWS Site-to-Site VPN?" in the *AWS Site-to-Site VPN User Guide* at https://docs.aws.amazon.com/vpn/latest/s2svpn/VPC_VPN.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.

Knowledge check question 3

Which statements are true of VPC peering connections? (Select TWO.)

Choice	Response
A	Connections are one-to-many.
B	Connections are one-to-one.
C	Connections require a transit gateway.
D	Connections can span accounts.
E	Connections are transitive.

42

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 and answer

Which statements are true of VPC peering connections? (Select TWO.)

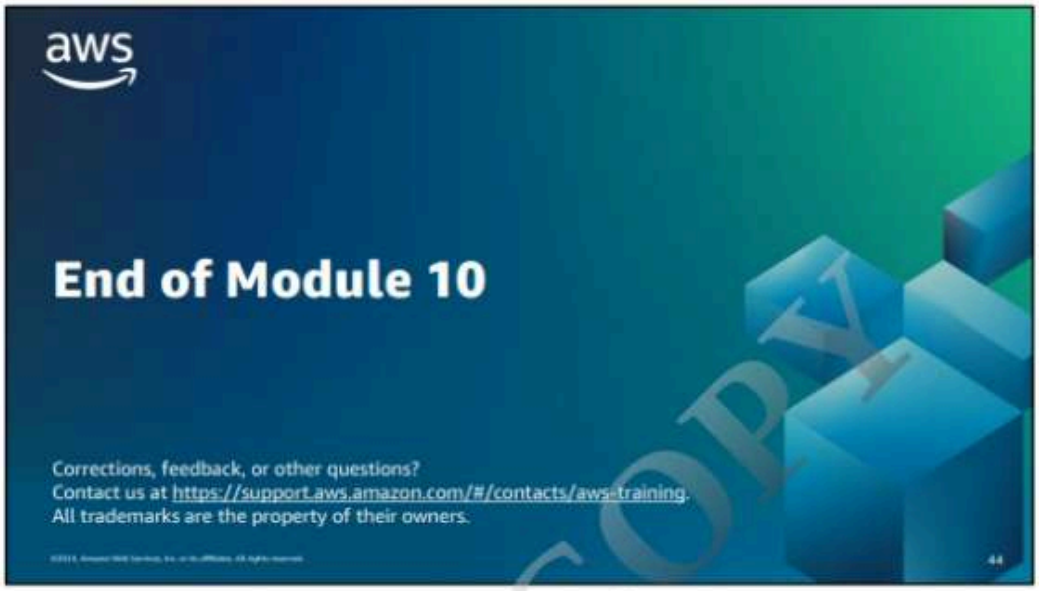
Choice	Response
A	Connections are one-to-many.
B correct	Connections are one-to-one.
C	Connections require a transit gateway.
D correct	Connections can span accounts.
E	Connections are transitive.

The answers are B and D, connections are one-to-one and connections can span accounts.

A VPC peering connection is a one-to-one relationship between two VPCs. Only one peering resource can exist between any two VPCs. You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported.

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.

DO NOT COPY
javierartiga.espublico@gmail.com

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