Chapter 11
Additional Key Services

THE AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE EXAM TOPICS OBJECTIVES COVERED IN THIS CHAPTER MAY INCLUDE, BUT ARE NOT LIMITED TO, THE FOLLOWING:

Domain 1.0: Designing highly available, cost-efficient, fault-tolerant, and scalable systems

✓ **1.1 Identify and recognize cloud architecture considerations, such as fundamental components and effective designs.**

Content may include the following:

- How to design cloud services
- Planning and design
- Monitoring and logging

Domain 2.0: Implementation/Deployment

✓ **2.1 Identify the appropriate techniques and methods using Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), AWS Elastic Beanstalk, AWS CloudFormation, AWS OpsWorks, Amazon Virtual Private Cloud (Amazon VPC), and AWS Identity and Access Management (IAM) to code and implement a cloud solution.**

Content may include the following:

- Configure services to support compliance requirements in the cloud
- Launch instances across the AWS global infrastructure

Domain 3.0: Data Security

✓ **3.1 Recognize and implement secure practices for optimum cloud deployment and maintenance.**

Content may include the following:

- AWS platform compliance
- AWS security attributes (customer workloads down to physical layer)
- AWS administration and security services
- AWS CloudTrail
- Ingress vs. egress filtering and which AWS cloud services and features fit
- Encryption solutions (e.g., key services)
- AWS Trusted Advisor

✓ **3.2 Recognize critical disaster recovery techniques and their**

implementation.

Content may include the following:

- AWS Import/Export
- AWS Storage Gateway

Introduction

Because Solutions Architects are often involved in solutions across a wide variety of business verticals and use cases, it is important to understand the basics of all AWS cloud service offerings. This chapter focuses on additional key AWS services that you should know at a high level to be successful on the exam. These services are grouped into four categories: Storage and Content Delivery, Security, Analytics, and DevOps.

Before architecting any system, foundational practices that influence security should be in place; for example, providing directories that contain organizational information or how encryption protects data by way of rendering it unintelligible to unauthorized access. As a Solutions Architect, understanding the AWS cloud services available to support an organization's directories and encryption are important because they support objectives such as identity management or complying with regulatory obligations.

Architecting analytical solutions is critical because the amount of data that companies need to understand continues to grow to record sizes. AWS provides analytic services that can scale to very large data stores efficiently and cost-effectively. Understanding these services allows Solutions Architects to build virtually any big data application and support any workload regardless of volume, velocity, and variety of data.

DevOps becomes an important concept as the pace of innovation accelerates and customer needs rapidly evolve, forcing businesses to become increasingly agile. Time to market is key, and to facilitate overall business goals, IT departments need to be agile. Understanding the DevOps options that are available on AWS will help Solutions Architects meet the demands of agile businesses that need IT operations to deploy applications in a consistent, repeatable, and reliable manner.

Understanding these additional services will not only help in your exam preparation, but it will also help you establish a foundation for growing as a Solutions Architect on the AWS platform.

Storage and Content Delivery

This section covers two additional storage and content delivery services that are important for a Solutions Architect to understand: Amazon CloudFront and AWS Storage Gateway.

Amazon CloudFront

Amazon CloudFront is a global Content Delivery Network (CDN) service. It integrates with other AWS products to give developers and businesses an easy way to distribute content to end users with low latency, high data transfer speeds, and no minimum usage commitments.

Overview

A *Content Delivery Network (CDN)* is a globally distributed network of caching servers that speed up the downloading of web pages and other content. CDNs use Domain Name System (DNS) *geo-location* to determine the geographic location of each request for a web page or other content, then they serve that content from edge caching servers closest to that location instead of the original web server. A CDN allows you to increase the scalability of a website or mobile application easily in response to peak traffic spikes. In most cases, using a CDN is completely transparent—end users simply experience better website performance, while the load on your original website is reduced.

Amazon CloudFront is AWS CDN. It can be used to deliver your web content using Amazon's global network of *edge locations*. When a user requests content that you're serving with Amazon CloudFront, the user is routed to the edge location that provides the lowest latency (time delay), so content is delivered with the best possible performance. If the content is already in the edge location with the lowest latency, Amazon CloudFront delivers it immediately. If the content is not currently in that edge location, Amazon CloudFront retrieves it from the *origin server*, such as an Amazon Simple Storage Service (Amazon S3) bucket or a web server, which stores the original, definitive versions of your files.

Amazon CloudFront is optimized to work with other AWS cloud services as the origin server, including Amazon S3 buckets, Amazon S3 static websites, Amazon Elastic Compute Cloud (Amazon EC2), and Elastic Load Balancing. Amazon CloudFront also works seamlessly with any non-AWS origin server, such as an existing on-premises web server. Amazon CloudFront also integrates with Amazon Route 53.

Amazon CloudFront supports all content that can be served over HTTP or HTTPS. This includes any popular static files that are a part of your web application, such as HTML files, images, JavaScript, and CSS files, and also audio, video, media files, or software downloads. Amazon CloudFront also supports serving dynamic web pages, so it can actually be used to deliver your entire website. Finally, Amazon CloudFront supports media *streaming*, using both HTTP and RTMP.

Amazon CloudFront Basics

There are three core concepts that you need to understand in order to start using CloudFront: distributions, origins, and cache control. With these concepts, you can easily use CloudFront to speed up delivery of static content from your websites.

Distributions To use Amazon CloudFront, you start by creating a *distribution*, which is identified by a DNS domain name such as d111111abcdef8.cloudfront.net. To serve files from Amazon CloudFront, you simply use the distribution domain name in place of your website's domain name; the rest of the file paths stay unchanged. You can use the Amazon CloudFront distribution domain name as-is, or you can create a user-friendly DNS name in your own domain by creating a CNAME record in Amazon Route 53 or another DNS service. The CNAME is automatically redirected to your Amazon CloudFront distribution domain name.

Origins When you create a distribution, you must specify the DNS domain name of the *origin*—the Amazon S3 bucket or HTTP server—from which you want Amazon CloudFront to get the definitive version of your objects (web files). For example:

- **Amazon S3 bucket:** `myawsbucket.s3.amazonaws.com`
- **Amazon EC2 instance:** `ec2-203-0-113-25.compute-1.amazonaws.com`
- **Elastic Load Balancing load balancer:** `my-load-balancer-1234567890.us-west-2.elb.amazonaws.com`
- **Website URL:** `mywebserver.mycompanydomain.com`

Cache Control Once requested and served from an edge location, objects stay in the cache until they expire or are evicted to make room for more frequently requested content. By default, objects expire from the cache after 24 hours. Once an object expires, the next request results in Amazon CloudFront forwarding the request to the origin to verify that the object is unchanged or to fetch a new version if it has changed.

Optionally, you can control how long objects stay in an Amazon CloudFront cache before expiring. To do this, you can choose to use Cache-Control headers set by your origin server or you can set the minimum, maximum, and default *Time to Live (TTL)* for objects in your Amazon CloudFront distribution.

You can also remove copies of an object from all Amazon CloudFront edge locations at any time by calling the *invalidation* Application Program Interface (API). This feature removes the object from every Amazon CloudFront edge location regardless of the expiration period you set for that object on your origin server. The invalidation feature is designed to be used in unexpected circumstances, such as to correct an error or to make an unanticipated update to a website, not as part of your everyday workflow.

Instead of invalidating objects manually or programmatically, it is a best practice to use a version identifier as part of the object (file) path name. For example:

- **Old file:** `assets/v1/css/narrow.css`
- **New file:** `assets/v2/css/narrow.css`

When using versioning, users always see the latest content through Amazon CloudFront when you update your site without using invalidation. Old versions will expire from the cache automatically.

Amazon CloudFront Advanced Features

CloudFront can do much more than simply serve static web files. To start using CloudFront's advanced features, you will need to understand how to use cache behaviors, and how to

restrict access to sensitive content.

Dynamic Content, Multiple Origins, and Cache Behaviors Serving static assets, such as described previously, is a common way to use a CDN. An Amazon CloudFront distribution, however, can easily be set up to serve dynamic content in addition to static content and to use more than one origin server. You control which requests are served by which origin and how requests are cached using a feature called *cache behaviors*.

A cache behavior lets you configure a variety of Amazon CloudFront functionalities for a given URL path pattern for files on your website. For example see Figure 11.1. One cache behavior applies to all PHP files in a web server (dynamic content), using the path pattern `*.php`, while another behavior applies to all JPEG images in another origin server (static content), using the path pattern `*.jpg`.

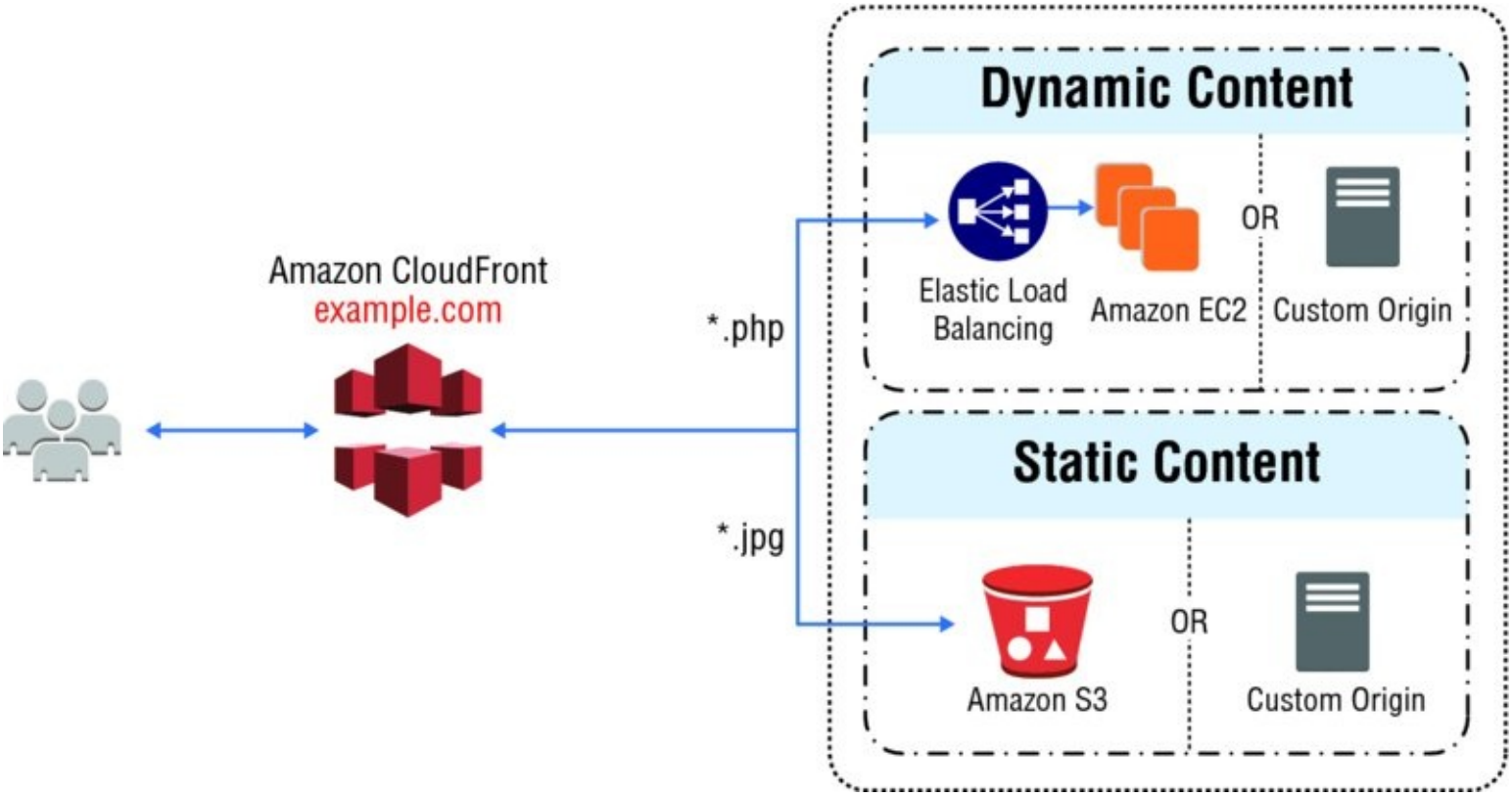

FIGURE 11.1 Delivering static and dynamic content

The functionality you can configure for each cache behavior includes the following:

- The path pattern
- Which origin to forward your requests to
- Whether to forward query strings to your origin
- Whether accessing the specified files requires signed URLs
- Whether to require HTTPS access
- The amount of time that those files stay in the Amazon CloudFront cache (regardless of the value of any Cache-Control headers that your origin adds to the files)

Cache behaviors are applied in order; if a request does not match the first path pattern, it drops down to the next path pattern. Normally the last path pattern specified is `*` to match all files.

Whole Website Using cache behaviors and multiple origins, you can easily use Amazon CloudFront to serve your whole website and to support different behaviors for different client devices.

Private Content In many cases, you may want to restrict access to content in Amazon CloudFront to only selected requestors, such as paid subscribers or to applications or users in your company network. Amazon CloudFront provides several mechanisms to allow you to serve private content. These include:

Signed URLs Use URLs that are valid only between certain times and optionally from certain IP addresses.

Signed Cookies Require authentication via public and private key pairs.

Origin Access Identities (OAI) Restrict access to an Amazon S3 bucket only to a special Amazon CloudFront user associated with your distribution. This is the easiest way to ensure that content in a bucket is only accessed by Amazon CloudFront.

Use Cases

There are several use cases where Amazon CloudFront is an excellent choice, including, but not limited to:

Serving the Static Assets of Popular Websites Static assets such as images, CSS, and JavaScript traditionally make up the bulk of requests to typical websites. Using Amazon CloudFront will speed up the user experience and reduce load on the website itself.

Serving a Whole Website or Web Application Amazon CloudFront can serve a whole website containing both dynamic and static content by using multiple origins, cache behaviors, and short TTLs for dynamic content.

Serving Content to Users Who Are Widely Distributed Geographically Amazon CloudFront will improve site performance, especially for distant users, and reduce the load on your origin server.

Distributing Software or Other Large Files Amazon CloudFront will help speed up the download of these files to end users.

Serving Streaming Media Amazon CloudFront helps serve streaming media, such as audio and video.

There are also use cases where CloudFront is not appropriate, including:

All or Most Requests Come From a Single Location If all or most of your requests come from a single geographic location, such as a large corporate campus, you will not take advantage of multiple edge locations.

All or Most Requests Come Through a Corporate VPN Similarly, if your users connect via a corporate Virtual Private Network (VPN), even if they are distributed, user requests appear to CloudFront to originate from one or a few locations. These use cases will generally not see benefit from using Amazon CloudFront.

AWS Storage Gateway

AWS Storage Gateway is a service connecting an on-premises software appliance with cloud-

based storage to provide seamless and secure integration between an organization's on-premises IT environment and AWS storage infrastructure. The service enables you to store data securely on the AWS cloud in a scalable and cost-effective manner. AWS Storage Gateway supports industry-standard storage protocols that work with your existing applications. It provides low-latency performance by caching frequently accessed data on-premises while encrypting and storing all of your data in Amazon S3 or Amazon Glacier.

Overview

AWS Storage Gateway's software appliance is available for download as a Virtual Machine (VM) image that you install on a host in your data center and then register with your AWS account through the AWS Management Console. The storage associated with the appliance is exposed as an iSCSI device that can be mounted by your on-premises applications.

There are three configurations for AWS Storage Gateway: Gateway-Cached volumes, Gateway-Stored volumes, and Gateway-Virtual Tape Libraries (VTL).

Gateway-Cached Volumes *Gateway-Cached volumes* allow you to expand your local storage capacity into Amazon S3. All data stored on a Gateway-Cached volume is moved to Amazon S3, while recently read data is retained in local storage to provide low-latency access. While each volume is limited to a maximum size of 32TB, a single gateway can support up to 32 volumes for a maximum storage of 1 PB.

Point-in-time snapshots can be taken to back up your AWS Storage Gateway. These snapshots are performed incrementally, and only the data that has changed since the last snapshot is stored.

All Gateway-Cached volume data and snapshot data is transferred to Amazon S3 over encrypted Secure Sockets Layer (SSL) connections. It is encrypted at rest in Amazon S3 using Server-Side Encryption (SSE). However, you cannot directly access this data with the Amazon S3 API or other tools such as the Amazon S3 console; instead you must access it through the AWS Storage Gateway service.

Gateway-Stored Volumes *Gateway-Stored volumes* allow you to store your data on your on-premises storage and asynchronously back up that data to Amazon S3. This provides low-latency access to all data, while also providing off-site backups taking advantage of the durability of Amazon S3. The data is backed up in the form of Amazon Elastic Block Store (Amazon EBS) snapshots. While each volume is limited to a maximum size of 16TB, a single gateway can support up to 32 volumes for a maximum storage of 512TB.

Similar to Gateway-Cached volumes, you can take snapshots of your Gateway-Stored volumes. The gateway stores these snapshots in Amazon S3 as *Amazon EBS snapshots*. When you take a new snapshot, only the data that has changed since your last snapshot is stored. You can initiate snapshots on a scheduled or one-time basis. Because these snapshots are stored as Amazon EBS snapshots, you can create a new Amazon EBS volume from a Gateway-Stored volume.

All Gateway-Stored volume data and snapshot data is transferred to Amazon S3 over encrypted SSL connections. It is encrypted at rest in Amazon S3 using SSE. However, you cannot access this data with the Amazon S3 API or other tools such as the Amazon S3 console.

If your on-premises appliance or even entire data center becomes unavailable, the data in AWS Storage Gateway can still be retrieved. If it's only the appliance that is unavailable, a new appliance can be launched in the data center and attached to the existing AWS Storage Gateway. A new appliance can also be launched in another data center or even on an Amazon EC2 instance on the cloud.

Gateway Virtual Tape Libraries (VTL) Gateway-VTL offers a durable, cost-effective solution to archive your data on the AWS cloud. The *VTL* interface lets you leverage your existing tape-based backup application infrastructure to store data on virtual tape cartridges that you create on your Gateway-VTL.

A virtual tape is analogous to a physical tape cartridge, except the data is stored on the AWS cloud. Tapes are created blank through the console or programmatically and then filled with backed up data. A gateway can contain up to 1,500 tapes (1 PB) of total tape data. Virtual tapes appear in your gateway's VTL, a virtualized version of a physical tape library. Virtual tapes are discovered by your backup application using its standard media inventory procedure.

When your tape software ejects a tape, it is archived on a Virtual Tape Shelf (VTS) and stored in Amazon Glacier. You're allowed 1 VTS per AWS region, but multiple gateways in the same region can share a VTS.

Use Cases

There are several use cases where AWS Storage Gateway is an excellent choice, including, but not limited to:

- Gateway-Cached volumes enable you to expand local storage hardware to Amazon S3, allowing you to store much more data without drastically increasing your storage hardware or changing your storage processes.
- Gateway-Stored volumes provide seamless, asynchronous, and secure backup of your on-premises storage without new processes or hardware.
- Gateway-VTLs enable you to keep your current tape backup software and processes while storing your data more cost-effectively and simply on the cloud.

Security

Cloud security at AWS is the highest priority. AWS customers benefit from data centers and network architectures built to meet the requirements of the most security-sensitive organizations.

An advantage of the AWS cloud is that it allows customers to scale and innovate while maintaining a secure environment. Cloud security is much like security in your on-premises data centers, only without the costs of maintaining facilities and hardware. In the cloud, you don't have to manage physical servers or storage devices. Instead, you use software-based security tools to monitor and protect the flow of information into and of out of your cloud resources.

This section will focus on four AWS services that are directly related to the specific security purposes: AWS Directory Service for identity management, AWS Key Management Service (KMS), AWS CloudHSM for key management, and AWS CloudTrail for auditing.

AWS Directory Service

AWS Directory Service is a managed service offering that provides directories that contain information about your organization, including users, groups, computers, and other resources.

Overview

You can choose from three directory types:

- AWS Directory Service for Microsoft Active Directory (Enterprise Edition), also referred to as Microsoft AD
- Simple AD
- AD Connector

As a managed offering, AWS Directory Service is designed to reduce identity management tasks, thereby allowing you to focus more of your time and resources on your business. There is no need to build out your own complex, highly-available directory topology because each directory is deployed across multiple Availability Zones, and monitoring automatically detects and replaces domain controllers that fail. In addition, data replication and automated daily snapshots are configured for you. There is no software to install, and AWS handles all of the patching and software updates.

AWS Directory Service for Microsoft Active Directory (Enterprise Edition) AWS Directory Service for Microsoft Active Directory (Enterprise Edition) is a managed *Microsoft Active Directory* hosted on the AWS cloud. It provides much of the functionality offered by Microsoft Active Directory plus integration with AWS applications. With the additional Active Directory functionality, you can, for example, easily set up trust relationships with your existing Active Directory domains to extend those directories to AWS cloud services.

Simple AD Simple AD is a Microsoft Active Directory-compatible directory from AWS Directory Service that is powered by Samba 4. Simple AD supports commonly used Active

Directory features such as user accounts, group memberships, domain-joining Amazon EC2 instances running Linux and Microsoft Windows, Kerberos-based Single Sign-On (SSO), and group policies. This makes it even easier to manage Amazon EC2 instances running Linux and Windows and deploy Windows applications on the AWS cloud.

Many of the applications and tools you use today that require Microsoft Active Directory support can be used with Simple AD. User accounts in Simple AD can also access AWS applications, such as Amazon WorkSpaces, Amazon WorkDocs, or Amazon WorkMail. They can also use AWS IAM roles to access the AWS Management Console and manage AWS resources. Finally, Simple AD provides daily automated snapshots to enable point-in-time recovery.

Note that you cannot set up trust relationships between Simple AD and other Active Directory domains. Other features not supported at the time of this writing by Simple AD include DNS dynamic update, schema extensions, Multi-Factor Authentication (MFA), communication over Lightweight Directory Access Protocol (LDAP), PowerShell AD cmdlets, and the transfer of *Flexible Single-Master Operations (FSMO)* roles.

AD Connector AD Connector is a proxy service for connecting your on-premises Microsoft Active Directory to the AWS cloud without requiring complex directory synchronization or the cost and complexity of hosting a federation infrastructure.

AD Connector forwards sign-in requests to your Active Directory domain controllers for authentication and provides the ability for applications to query the directory for data. After setup, your users can use their existing corporate credentials to log on to AWS applications, such as Amazon WorkSpaces, Amazon WorkDocs, or Amazon WorkMail. With the proper IAM permissions, they can also access the AWS Management Console and manage AWS resources such as Amazon EC2 instances or Amazon S3 buckets. You can also use AD Connector to enable MFA by integrating it with your existing *Remote Authentication Dial-Up Service (RADIUS)*-based MFA infrastructure to provide an additional layer of security when users access AWS applications.

With AD Connector, you continue to manage your Active Directory as usual. For example, adding new users, adding new groups, or updating passwords are all accomplished using standard directory administration tools with your on-premises directory. Thus, in addition to providing a streamlined experience for your users, AD Connector enables consistent enforcement of your existing security policies, such as password expiration, password history, and account lockouts, whether users are accessing resources on-premises or on the AWS cloud.

Use Cases

AWS Directory Service provides multiple ways to use Microsoft Active Directory with other AWS cloud services. You can choose the directory service with the features you need at a cost that fits your budget.

AWS Directory Service for Microsoft Active Directory (Enterprise Edition) This Directory Service is your best choice if you have more than 5,000 users and need a trust relationship set up between an AWS-hosted directory and your on-premises directories.

Simple AD In most cases, Simple AD is the least expensive option and your best choice if

you have 5,000 or fewer users and don't need the more advanced Microsoft Active Directory features.

AD Connector AD Connector is your best choice when you want to use your existing on-premises directory with AWS cloud services.

AWS Key Management Service (KMS) and AWS CloudHSM

Key management is the management of *cryptographic keys* within a *cryptosystem*. This includes dealing with the generation, exchange, storage, use, and replacement of keys.

Overview

AWS offers two services that provide you with the ability to manage your own *symmetric* or *asymmetric* cryptographic keys:

- **AWS KMS:** A service enabling you to generate, store, enable/disable, and delete symmetric keys
- **AWS CloudHSM:** A service providing you with secure cryptographic key storage by making Hardware Security Modules (HSMs) available on the AWS cloud

AWS Key Management Service (AWS KMS) *AWS KMS* is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. AWS KMS lets you create keys that can never be exported from the service and that can be used to encrypt and decrypt data based on policies you define.

By using AWS KMS, you gain more control over access to data you encrypt. You can use the key management and cryptographic features directly in your applications or through AWS cloud services that are integrated with AWS KMS. Whether you are writing applications for AWS or using AWS cloud services, AWS KMS enables you to maintain control over who can use your keys and gain access to your encrypted data.

Customer Managed Keys AWS KMS uses a type of key called a *Customer Master Key (CMK)* to encrypt and decrypt data. CMKs are the fundamental resources that AWS KMS manages. They can be used inside of AWS KMS to encrypt or decrypt up to 4 KB of data directly. They can also be used to encrypt generated *data keys* that are then used to encrypt or decrypt larger amounts of data outside of the service. CMKs can never leave AWS KMS unencrypted, but data keys can leave the service unencrypted.

Data Keys You use data keys to encrypt large data objects within your own application outside AWS KMS. When you call `GenerateDataKey`, AWS KMS returns a plaintext version of the key and ciphertext that contains the key encrypted under the specified CMK. AWS KMS tracks which CMK was used to encrypt the data key. You use the plaintext data key in your application to encrypt data, and you typically store the encrypted key alongside your encrypted data. Security best practices suggest that you should remove the plaintext key from memory as soon as is practical after use. To decrypt data in your application, pass the encrypted data key to the Decrypt function. AWS KMS uses the associated CMK to decrypt and retrieve your plaintext data key. Use the plaintext key to decrypt your data, and then remove the key from memory.

Envelope Encryption AWS KMS uses *envelope encryption* to protect data. AWS KMS creates a data key, encrypts it under a CMK, and returns plaintext and encrypted versions of the data key to you. You use the plaintext key to encrypt data and store the encrypted key alongside the encrypted data. The key should be removed from memory as soon as is practical after use. You can retrieve a plaintext data key only if you have the encrypted data key and you have permission to use the corresponding master key.

Encryption Context All AWS KMS cryptographic operations accept an optional key/value map of additional contextual information called an *encryption context*. The specified context must be the same for both the encrypt and decrypt operations or decryption will not succeed. The encryption context is logged, can be used for additional auditing, and is available as context in the AWS policy language for fine-grained policy-based authorization.

AWS CloudHSM AWS CloudHSM helps you meet corporate, contractual, and regulatory compliance requirements for data security by using dedicated HSM appliances within the AWS cloud. An *HSM* is a hardware appliance that provides secure key storage and cryptographic operations within a tamper-resistant hardware module. HSMs are designed to securely store cryptographic key material and use the key material without exposing it outside the cryptographic boundary of the appliance.

The recommended configuration for using AWS CloudHSM is to use two HSMs configured in a high-availability configuration, as illustrated in Figure 11.2.

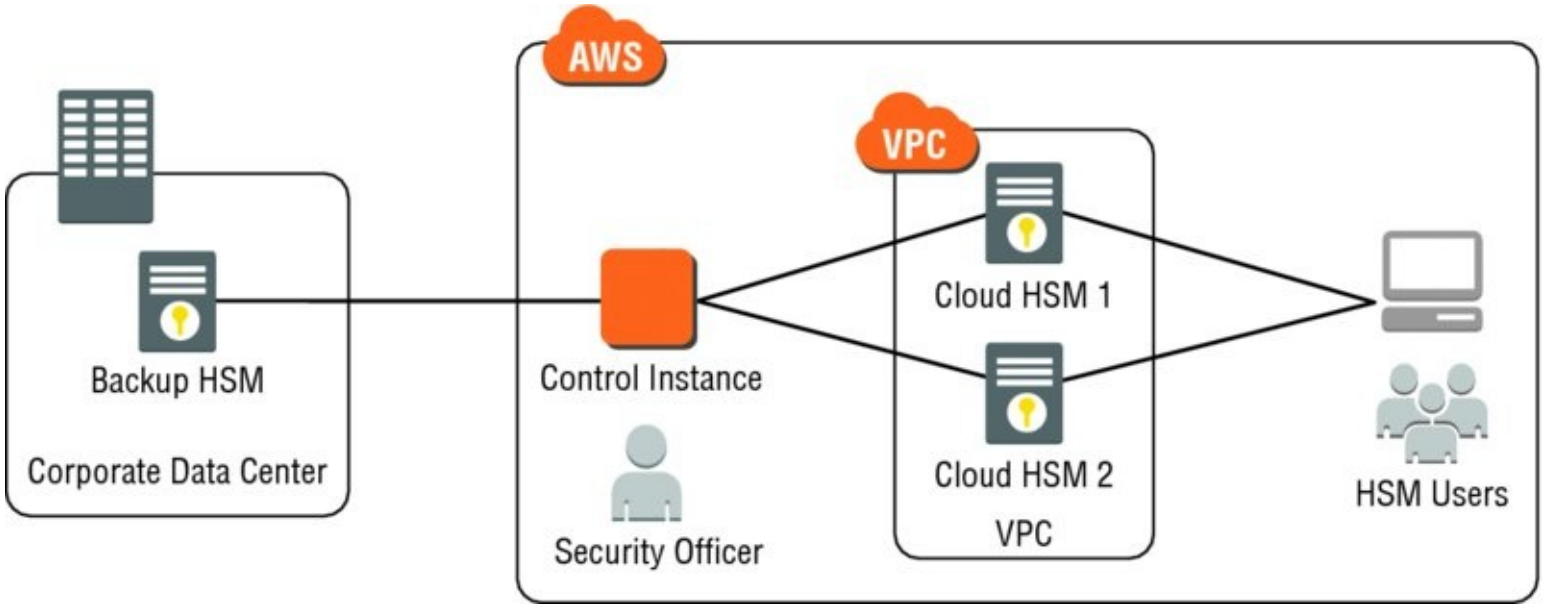

FIGURE 11.2 High availability CloudHSM architecture

AWS CloudHSM allows you to protect your encryption keys within HSMs that are designed and validated to government standards for secure key management. You can securely generate, store, and manage the cryptographic keys used for data encryption in a way that ensures that only you have access to the keys. AWS CloudHSM helps you comply with strict key management requirements within the AWS cloud without sacrificing application performance.

Use Cases

The AWS key management services address several security needs that would require extensive effort to deploy and manage otherwise, including, but not limited to:

Scalable Symmetric Key Distribution Symmetric encryption algorithms require that the same key be used for both encrypting and decrypting the data. This is problematic because transferring the key from the sender to the receiver must be done either through a known secure channel or some "out of band" process.

Government-Validated Cryptography Certain types of data (for example, Payment Card Industry—PCI—or health information records) must be protected with cryptography that has been validated by an outside party as conforming to the algorithm(s) asserted by the claiming party.

AWS CloudTrail

AWS CloudTrail provides visibility into user activity by recording API calls made on your account. AWS CloudTrail records important information about each API call, including the name of the API, the identity of the caller, the time of the API call, the request parameters, and the response elements returned by the AWS service. This information helps you to track changes made to your AWS resources and to troubleshoot operational issues. AWS CloudTrail makes it easier to ensure compliance with internal policies and regulatory standards.

Overview

AWS CloudTrail captures AWS API calls and related events made by or on behalf of an AWS account and delivers log files to an Amazon S3 bucket that you specify. Optionally, you can configure AWS CloudTrail to deliver events to a log group monitored by Amazon CloudWatch Logs. You can also choose to receive Amazon Simple Notification Service (Amazon SNS) notifications each time a log file is delivered to your bucket. You can create a *trail* with the AWS CloudTrail console, the AWS Command Line Interface (CLI), or the AWS CloudTrail API. A trail is a configuration that enables logging of the AWS API activity and related events in your account.

You can create two types of trails:

A Trail That Applies to All Regions When you create a trail that applies to all AWS regions, AWS CloudTrail creates the same trail in each region, records the log files in each region, and delivers the log files to the single Amazon S3 bucket (and optionally to the Amazon CloudWatch Logs log group) that you specify. This is the default option when you create a trail using the AWS CloudTrail console. If you choose to receive Amazon SNS notifications for log file deliveries, one Amazon SNS topic will suffice for all regions. If you choose to have AWS CloudTrail send events from a trail that applies to all regions to an Amazon CloudWatch Logs log group, events from all regions will be sent to the single log group.

A Trail That Applies to One Region You specify a bucket that receives events only from that region. The bucket can be in any region that you specify. If you create additional individual trails that apply to specific regions, you can have those trails deliver event logs to a single Amazon S3 bucket.

By default, your log files are encrypted using Amazon S3 SSE. You can store your log files in your bucket for as long as you want, but you can also define Amazon S3 lifecycle rules to archive or delete log files automatically.

AWS CloudTrail typically delivers log files within 15 minutes of an API call. In addition, the service publishes new log files multiple times an hour, usually about every five minutes. These log files contain API calls from all of the account's services that support AWS CloudTrail.


Enable AWS CloudTrail on all of your AWS accounts. Instead of configuring a trail for one region, you should enable trails for all regions.

Use Cases

AWS CloudTrail is beneficial for several use cases:

External Compliance Audits Your business must demonstrate compliance to a set of regulations pertinent to some or all data being transmitted, processed, and stored within your AWS accounts. Events from AWS CloudTrail can be used to show the degree to which you are compliant with the regulations.

Unauthorized Access to Your AWS Account AWS CloudTrail records all sign-on attempts to your AWS account, including AWS Management Console login attempts, AWS

Software Development Kit (SDK) API calls, and AWS CLI API calls. Routine examination of AWS CloudTrail events will provide the needed information to determine if your AWS account is being targeted for unauthorized access.

Analytics

Analytics, and the associated big data that it requires, presents a unique list of challenges to a Solutions Architect. The big data must be ingested at a very high rate, stored in very high volume, and processed with a tremendous amount of compute. Often, the need to perform analytics on the big data is sporadic, with a great deal of compute infrastructure needed regularly for very small time periods. The cloud, with its easy access to compute and nearly limitless storage capacity, is ideally suited to address these analytics challenges. This section covers several AWS cloud services that will help you address analytics and big data issues on the exam.

Amazon Kinesis

Amazon Kinesis is a platform for handling massive streaming data on AWS, offering powerful services to make it easy to load and analyze streaming data and also providing the ability for you to build custom streaming data applications for specialized needs.

Overview

Amazon Kinesis is a streaming data platform consisting of three services addressing different real-time streaming data challenges:

- **Amazon Kinesis Firehose:** A service enabling you to load massive volumes of streaming data into AWS
- **Amazon Kinesis Streams:** A service enabling you to build custom applications for more complex analysis of streaming data in real time
- **Amazon Kinesis Analytics:** A service enabling you to easily analyze streaming data real time with standard SQL

Each of these services can scale to handle virtually limitless data streams.

Amazon Kinesis Firehose *Amazon Kinesis Firehose* receives stream data and stores it in Amazon S3, Amazon Redshift, or *Amazon Elasticsearch*. You do not need to write any code; just create a delivery stream and configure the destination for your data. Clients write data to the stream using an AWS API call and the data is automatically sent to the proper destination. The various destination options are shown in Figure 11.3.

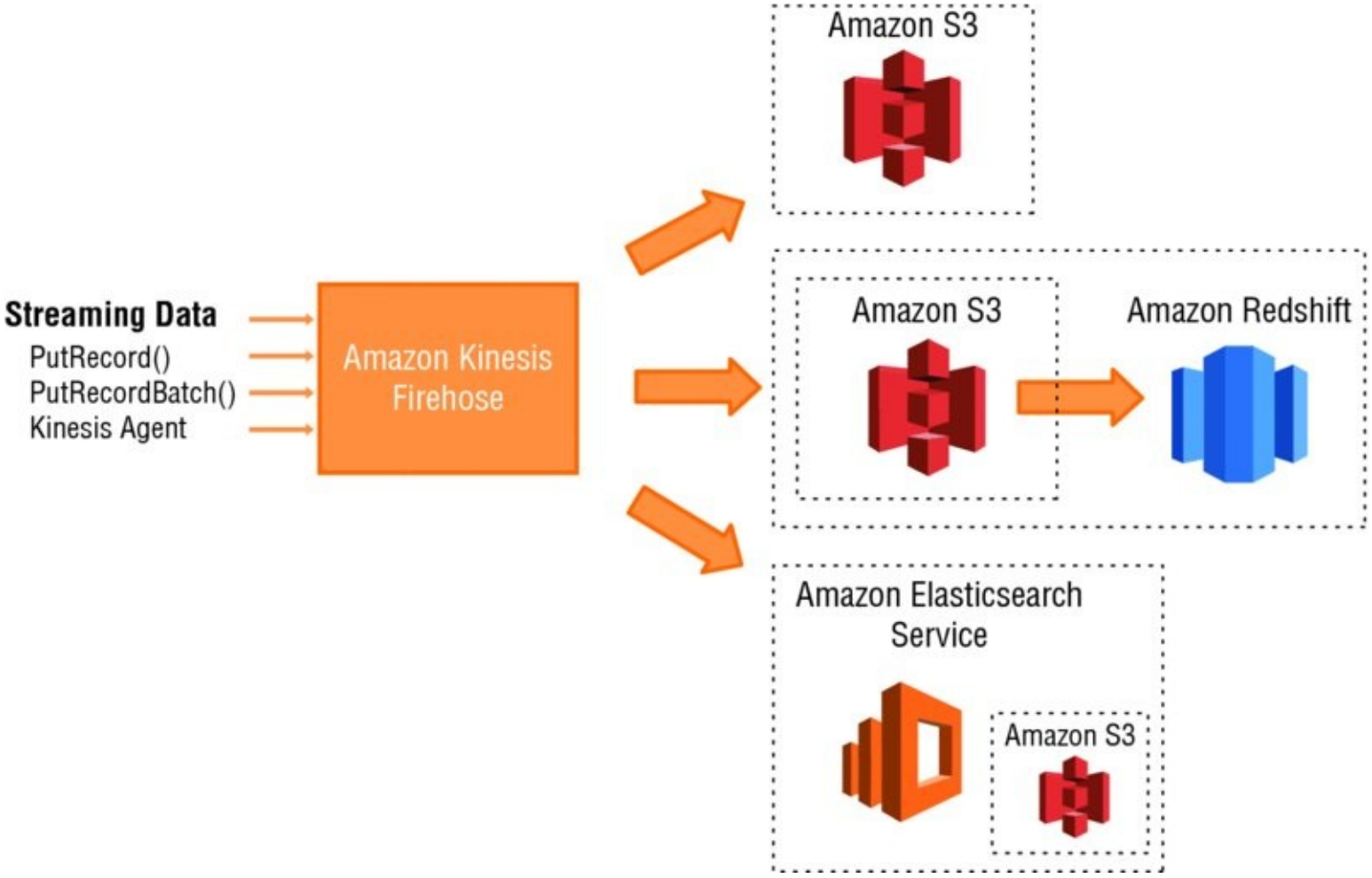

FIGURE 11.3 Amazon Kinesis Firehose

When configured to save a stream to Amazon S3, Amazon Kinesis Firehose sends the data directly to Amazon S3. For an Amazon Redshift destination, the data is first written to Amazon S3, and then an Amazon Redshift `COPY` command is executed to load the data into Amazon Redshift. Amazon Kinesis Firehose can also write data out to Amazon Elasticsearch, with the option to back the data up concurrently to Amazon S3.

Amazon Kinesis Streams *Amazon Kinesis Streams* enable you to collect and process large streams of data records in real time. Using AWS SDKs, you can create an *Amazon Kinesis Streams application* that processes the data as it moves through the stream. Because response time for data intake and processing is in near real time, the processing is typically lightweight. Amazon Kinesis Streams can scale to support nearly limitless data streams by distributing incoming data across a number of *shards*. If any shard becomes too busy, it can be further divided into more shards to distribute the load further. The processing is then executed on consumers, which read data from the shards and run the Amazon Kinesis Streams application. This architecture is shown in Figure 11.4.

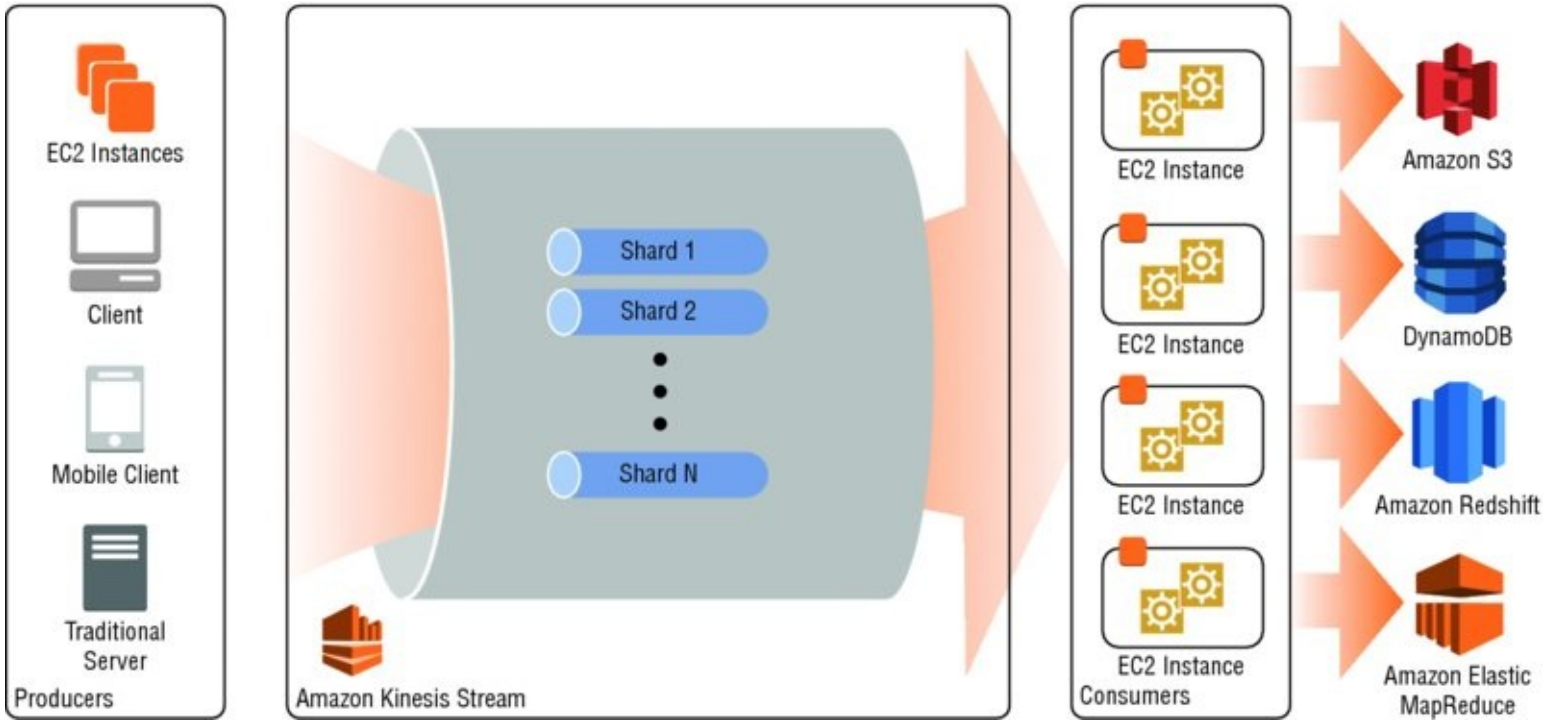

FIGURE 11.4 Amazon Kinesis Streams

Amazon Kinesis Analytics At the time of this writing, Amazon Kinesis Analytics has been announced but not yet released.

Use Cases

The Amazon Kinesis services support many strategic workloads that would otherwise require extensive effort to deploy and manage, including, but not limited to:

Data Ingestion The first challenge with a huge stream of data is accepting it reliably. Whether it is user data from highly trafficked websites, input data from thousands of monitoring devices, or any other sources of huge streams, Amazon Kinesis Firehose is an excellent choice to ensure that all of your data is successfully stored in your AWS infrastructure.

Real-Time Processing of Massive Data Streams Companies often need to act on knowledge gleaned from a big data stream right away, whether to feed a dashboard application, alter advertising strategies based on social media trends, allocate assets based on real-time situations, or a host of other scenarios. Amazon Kinesis Streams enables you to gather this knowledge from the data in your stream on a real-time basis.

It's good to remember that while Amazon Kinesis is ideally suited for ingesting and processing streams of data, it is less appropriate for batch jobs such as nightly *Extract, Transform, Load (ETL)* processes. For those types of workloads, consider AWS Data Pipeline, which is described later in this chapter.

Amazon Elastic MapReduce (Amazon EMR)

Amazon Elastic MapReduce (Amazon EMR) provides you with a fully managed, on-demand Hadoop framework. Amazon EMR reduces the complexity and up-front costs of setting up Hadoop and, combined with the scale of AWS, gives you the ability to spin up large Hadoop clusters instantly and start processing within minutes.

Overview

When you launch an Amazon EMR cluster, you specify several options, the most important being:

- The instance type of the nodes in your cluster
- The number of nodes in your cluster
- The version of Hadoop you want to run (Amazon EMR supports several recent versions of Apache Hadoop, and also several versions of MapR Hadoop.)
- Additional tools or applications like Hive, Pig, Spark, or Presto

There are two types of storage that can be used with Amazon EMR:

Hadoop Distributed File System (HDFS) *HDFS* is the standard file system that comes with Hadoop. All data is replicated across multiple instances to ensure durability. Amazon EMR can use Amazon EC2 instance storage or Amazon EBS for HDFS. When a cluster is shut down, instance storage is lost and the data does not persist. HDFS can also make use of Amazon EBS storage, trading in the cost effectiveness of instance storage for the ability to shut down a cluster without losing data.

EMR File System (EMRFS) *EMRFS* is an implementation of HDFS that allows clusters to store data on Amazon S3. EMRFS allows you to get the durability and low cost of Amazon S3 while preserving your data even if the cluster is shut down.

A key factor driving the type of storage a cluster uses is whether the cluster is persistent or transient. A *persistent cluster* continues to run 24×7 after it is launched. Persistent clusters are appropriate when continuous analysis is going to be run on the data. For persistent clusters, HDFS is a common choice. Persistent clusters take advantage of the low latency of HDFS, especially on instance storage, when constant operation means no data lost when shutting down a cluster. In other situations, big data workloads are frequently run inconsistently, and it can be cost-effective to turn the cluster off when not in use. Clusters that are started when needed and then immediately stopped when done are called *transient clusters*. EMRFS is well suited for transient clusters, as the data persists independent of the lifetime of the cluster. You can also choose to use a combination of local HDFS and EMRFS to meet your workload needs.

Because Amazon EMR is an instance of Apache Hadoop, you can use the extensive ecosystem of tools that work on top of Hadoop, such as Hive, Pig, and Spark. Many of these tools are natively supported and can be included automatically when you launch your cluster, while others can be installed through *bootstrap* actions.

Use Cases

Amazon EMR is well suited for a large number of use cases, including, but not limited to:

Log Processing Amazon EMR can be used to process logs generated by web and mobile applications. Amazon EMR helps customers turn petabytes of unstructured or semi-structured data into useful insights about their applications or users.

Clickstream Analysis Amazon EMR can be used to analyze *clickstream* data in order to segment users and understand user preferences. Advertisers can also analyze clickstreams

and advertising impression logs to deliver more effective ads.

Genomics and Life Sciences Amazon EMR can be used to process vast amounts of genomic data and other large scientific datasets quickly and efficiently. Processes that require years of compute can be completed in a day when scaled across large clusters.

AWS Data Pipeline

AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, and also on-premises data sources, at specified intervals. With AWS Data Pipeline, you can regularly access your data where it's stored, transform and process it at scale, and efficiently transfer the results to AWS services such as Amazon S3, Amazon Relational Database Service (Amazon RDS), Amazon DynamoDB, and Amazon EMR.

Overview

Everything in AWS Data Pipeline starts with the pipeline itself. A pipeline schedules and runs tasks according to the pipeline definition. The scheduling is flexible and can run every 15 minutes, every day, every week, and so forth.

The pipeline interacts with data stored in data nodes. Data nodes are locations where the pipeline reads input data or writes output data, such as Amazon S3, a MySQL database, or an Amazon Redshift cluster. Data nodes can be on AWS or on your premises.

The pipeline will execute *activities* that represent common scenarios, such as moving data from one location to another, running Hive queries, and so forth. Activities may require additional resources to run, such as an Amazon EMR cluster or an Amazon EC2 instance. In these situations, AWS Data Pipeline will automatically launch the required resources and tear them down when the activity is completed.

Distributed data flows often have dependencies; just because an activity is scheduled to run does not mean that there is data waiting to be processed. For situations like this, AWS Data Pipeline supports preconditions, which are conditional statements that must be true before an activity can run. These include scenarios such as whether an Amazon S3 key is present, whether an Amazon DynamoDB table contains any data, and so forth.

If an activity fails, retry is automatic. The activity will continue to retry up to the limit you configure. You can define actions to take in the event when the activity reaches that limit without succeeding.

Use Cases

AWS Data Pipeline can be used for virtually any batch mode ETL process. A simple example is shown in Figure 11.5.

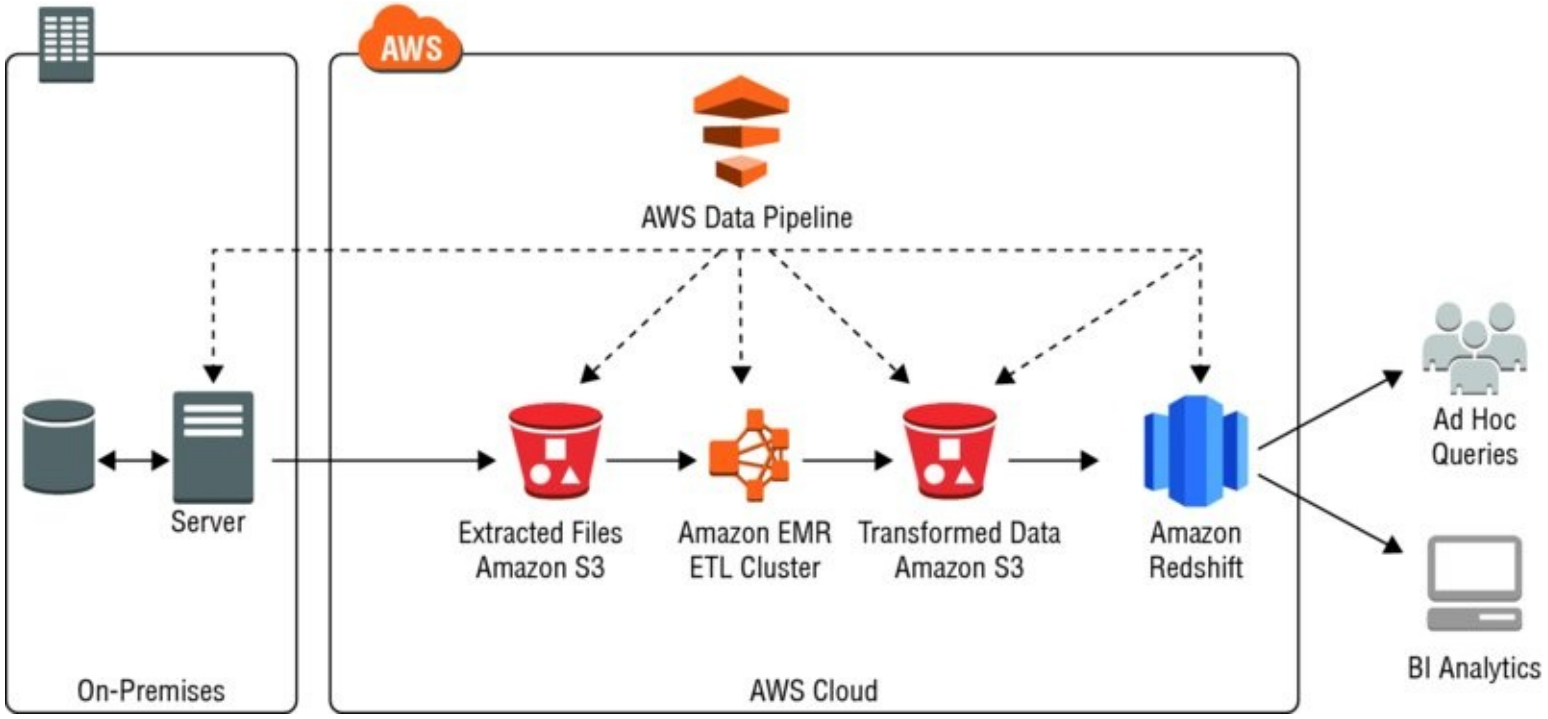

FIGURE 11.5 Example pipeline

The pipeline in Figure 11.5 is performing the following workflow:

- Every hour an activity begins to extract log data from on-premises storage to Amazon S3. A precondition checks that there is data to be transferred before actually starting the activity.
- The next activity launches a transient Amazon EMR cluster that uses the extracted dataset as input, validates and transforms it, and then outputs the data to an Amazon S3 bucket.
- The final activity moves the transformed data from Amazon S3 to Amazon Redshift via an Amazon Redshift `COPY` command.

AWS Data Pipeline is best for regular batch processes instead of for continuous data streams; use Amazon Kinesis for data streams.

AWS Import/Export

One key challenge of big data on the AWS cloud is getting huge datasets to the cloud in the first place, or retrieving them back to on-premises when necessary. Regardless of how much bandwidth you configure out of your data center, there are times when there is more data to transfer than can move over the connection in a reasonable period of time. AWS Import/Export is a service that accelerates transferring large amounts of data into and out of AWS using physical storage appliances, bypassing the Internet. The data is copied to a device at the source (your data center or an AWS region), shipped via standard shipping mechanisms, and then copied to the destination (your data center or an AWS region).

Overview

AWS Import/Export has two features that support shipping data into and out of your AWS infrastructure: AWS Import/Export Snowball (AWS Snowball) and AWS Import/Export Disk.

AWS Snowball AWS Snowball uses Amazon-provided shippable storage appliances shipped

through UPS. Each AWS Snowball is protected by AWS KMS and made physically rugged to secure and protect your data while the device is in transit. At the time of this writing, AWS Snowballs come in two sizes: 50TB and 80TB, and the availability of each varies by region.

AWS Snowball provides the following features:

- You can import and export data between your on-premises data storage locations and Amazon S3.
- Encryption is enforced, protecting your data at rest and in physical transit.
- You don't have to buy or maintain your own hardware devices.
- You can manage your jobs through the AWS Snowball console.
- The AWS Snowball is its own shipping container, and the shipping label is an *E Ink display* that automatically shows the correct address when the AWS Snowball is ready to ship. You can drop it off with UPS, no box required.

With AWS Snowball, you can import or export terabytes or even petabytes of data.

AWS Import/Export Disk AWS Import/Export Disk supports transfers data directly onto and off of storage devices you own using the Amazon high-speed internal network.

Important things to understand about AWS Import/Export Disk include:

- You can import your data into Amazon Glacier and Amazon EBS, in addition to Amazon S3.
- You can export data from Amazon S3.
- Encryption is optional and not enforced.
- You buy and maintain your own hardware devices.
- You can't manage your jobs through the AWS Snowball console.
- Unlike AWS Snowball, AWS Import/Export Disk has an upper limit of 16TB.

Use Cases

AWS Import/Export can be used for just about any situation where you have more data to move than you can get through your Internet connection in a reasonable time, including, but not limited to:

Storage Migration When companies shut down a data center, they often need to move massive amounts of storage to another location. AWS Import/Export is a suitable technology for this requirement.

Migrating Applications Migrating an application to the cloud often involves moving huge amounts of data. This can be accelerated using AWS Import/Export.

DevOps

As organizations created increasingly complex software applications, IT development teams evolved their software creation practices for more flexibility, moving from waterfall models to agile or lean development practices. This change also propagated to operations teams, which blurred the lines between traditional development and operations teams. AWS provides a flexible environment that facilitated the successes of organizations like Netflix, Airbnb, General Electric, and many others that embraced DevOps. This section reviews elements of AWS cloud services that support DevOps practices.

AWS OpsWorks

AWS OpsWorks is a configuration management service that helps you configure and operate applications using Chef. AWS OpsWorks will work with applications of any level of complexity and is independent of any particular architectural pattern. You can define an application's architecture and the specification of each component, including package installation, software configuration, and resources such as storage.

AWS OpsWorks supports both Linux or Windows servers, including existing Amazon EC2 instances or servers running in your own data center. This allows organizations to use a single configuration management service to deploy and operate applications across hybrid architectures.

Overview

Many solutions on AWS usually involve groups of resources, such as Amazon EC2 instances and Amazon RDS instances, which must be created and managed collectively. For example, these architectures typically require application servers, database servers, load balancers, and so on. This group of resources is typically called a *stack*. A simple application server stack might be arranged something like in Figure 11.6.

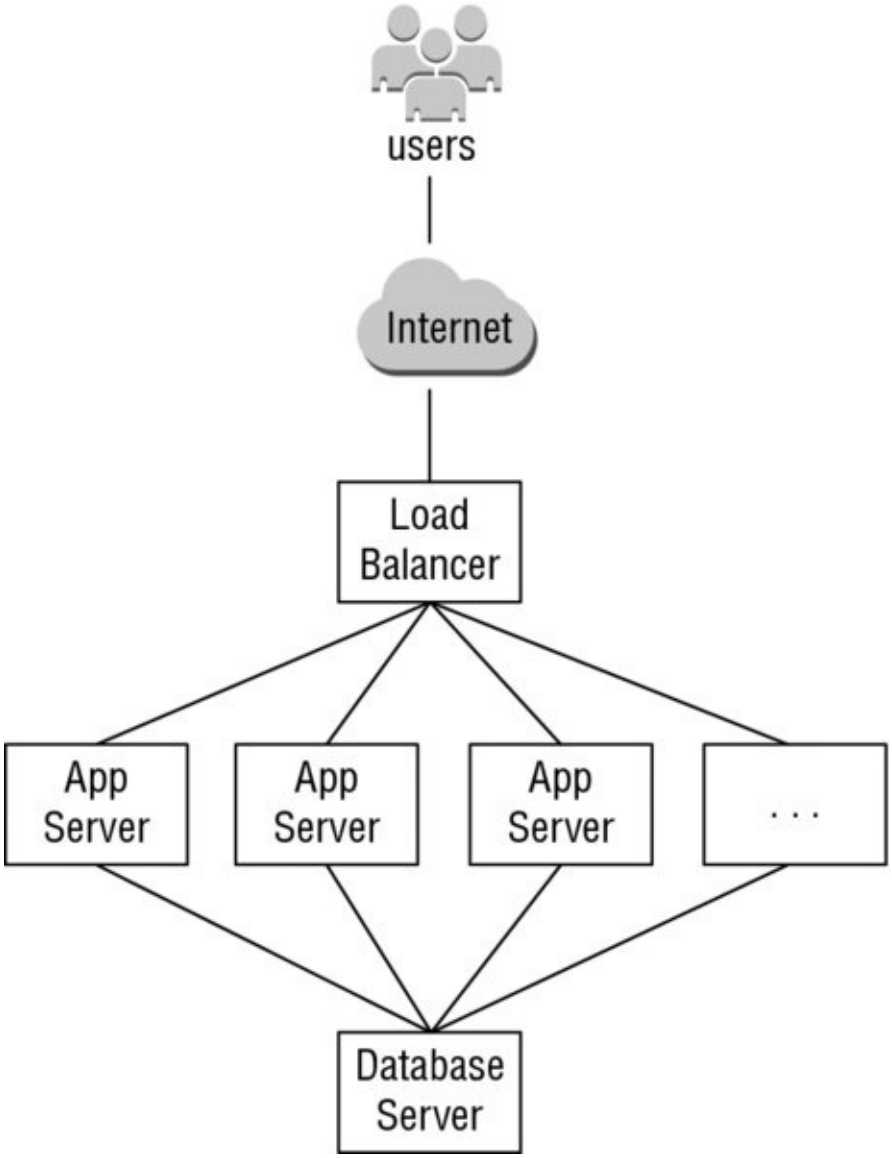

FIGURE 11.6 Simple application server stack

In addition to creating the instances and installing the necessary packages, you typically need a way to distribute applications to the application servers, monitor the stack's performance, manage security and permissions, and so on. AWS OpsWorks provides a simple and flexible way to create and manage stacks and applications. Figure 11.7 depicts how a simple application server stack might look with AWS OpsWorks. Although relatively simple, this stack shows the key AWS OpsWorks features.

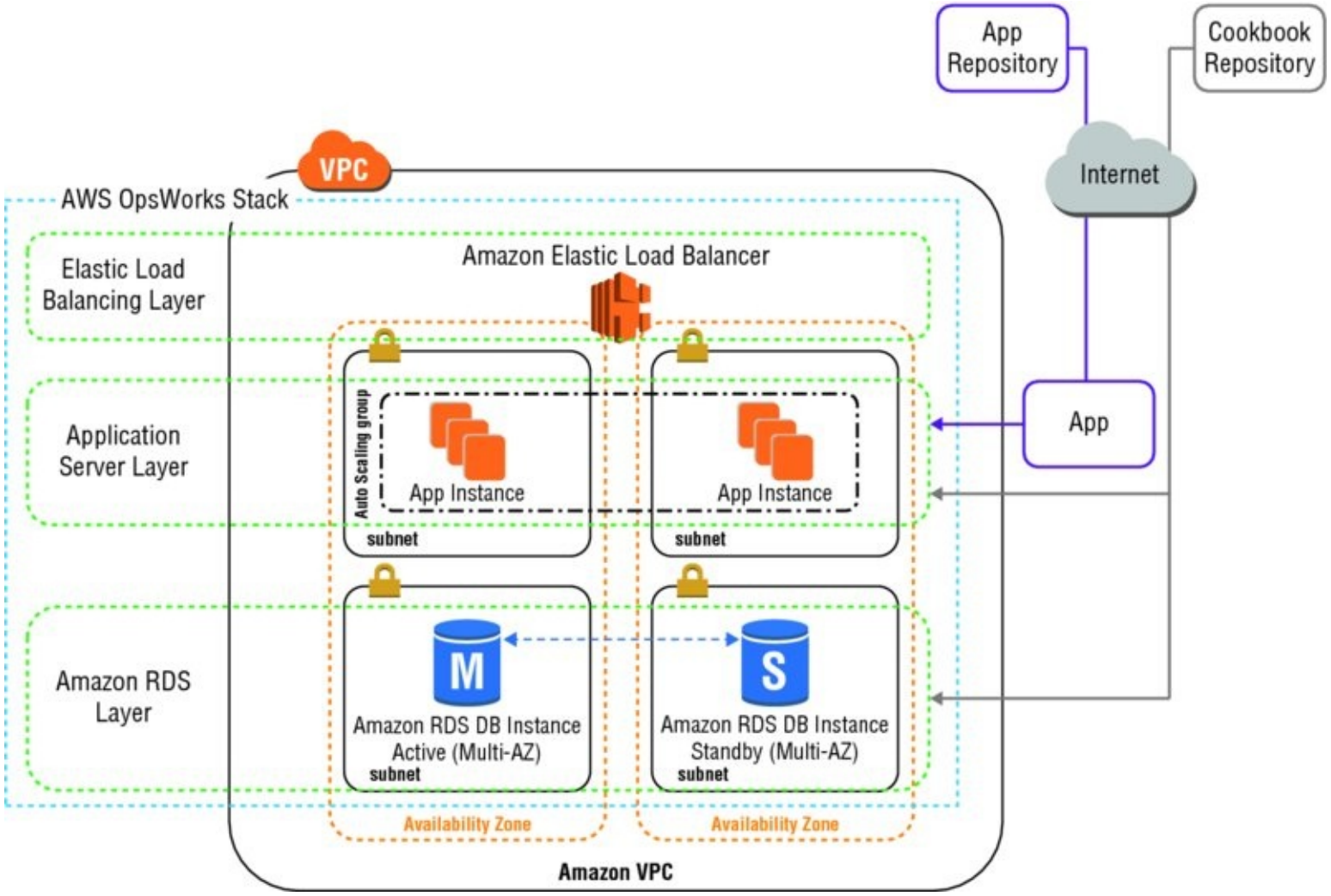

FIGURE 11.7 Simple application server stack with AWS OpsWorks

The *stack* is the core AWS OpsWorks component. It is basically a container for AWS resources—Amazon EC2 instances, Amazon RDS database instances, and so on—that have a common purpose and make sense to be logically managed together. The stack helps you manage these resources as a group and defines some default configuration settings, such as the Amazon EC2 instances' operating system and AWS region. If you want to isolate some stack components from direct user interaction, you can run the stack in an Amazon Virtual Private Cloud (Amazon VPC). Each stack lets you grant users permission to access the stack and specify what actions they can take.

You can use AWS OpsWorks or IAM to manage user permissions. Note that the two options are not mutually exclusive; it is sometimes desirable to use both.

You define the elements of a stack by adding one or more layers. A *layer* represents a set of resources that serve a particular purpose, such as load balancing, web applications, or hosting a database server. You can customize or extend layers by modifying the default configurations or adding Chef recipes to perform tasks such as installing additional packages. Layers give you complete control over which packages are installed, how they are configured, how applications are deployed, and more.

Layers depend on Chef recipes to handle tasks such as installing packages on instances,

deploying applications, and running scripts. One of the key AWS OpsWorks features is a set of lifecycle events that automatically run a specified set of recipes at the appropriate time on each instance.

An instance represents a single computing resource, such as an Amazon EC2 instance. It defines the resource's basic configuration, such as operating system and size. Other configuration settings, such as Elastic IP addresses or Amazon EBS volumes, are defined by the instance's layers. The layer's recipes complete the configuration by performing tasks, such as installing and configuring packages and deploying applications.

You store applications and related files in a repository, such as an Amazon S3 bucket or Git repo. Each application is represented by an *app*, which specifies the application type and contains the information that is needed to deploy the application from the repository to your instances, such as the repository URL and password. When you deploy an app, AWS OpsWorks triggers a Deploy event, which runs the Deploy recipes on the stack's instances.

Using the concepts of stacks, layers, and apps, you can model and visualize your application and resources in an organized fashion.

Finally, AWS OpsWorks sends all of your resource metrics to Amazon CloudWatch, making it easy to view graphs and set alarms to help you troubleshoot and take automated action based on the state of your resources. AWS OpsWorks provides many custom metrics, such as CPU idle, memory total, average load for one minute, and more. Each instance in the stack has detailed monitoring to provide insights into your workload.

Use Cases

AWS OpsWorks supports many DevOps efforts, including, but not limited to:

Host Multi-Tier Web Applications AWS OpsWorks lets you model and visualize your application with layers that define how to configure a set of resources that are managed together. Because AWS OpsWorks uses the Chef framework, you can bring your own recipes or leverage hundreds of community-built configurations.

Support Continuous Integration AWS OpsWorks supports DevOps principles, such as continuous integration. Everything in your environment can be automated.

AWS CloudFormation

AWS CloudFormation is a service that helps you model and set up your AWS resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS. AWS CloudFormation allows organizations to deploy, modify, and update resources in a controlled and predictable way, in effect applying version control to AWS infrastructure the same way one would do with software.

Overview

AWS CloudFormation gives developers and systems administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly

and predictable fashion. When you use AWS CloudFormation, you work with *templates* and *stacks*.

You create AWS CloudFormation templates to define your AWS resources and their properties. A *template* is a text file whose format complies with the JSON standard. AWS CloudFormation uses these templates as blueprints for building your AWS resources.

When you use AWS CloudFormation, you can reuse your template to set up your resources consistently and repeatedly. Just describe your resources once, and then provision the same resources over and over in multiple regions.

When you use AWS CloudFormation, you manage related resources as a single unit called a stack. You create, update, and delete a collection of resources by creating, updating, and deleting stacks. All of the resources in a stack are defined by the stack's AWS CloudFormation template. Suppose you created a template that includes an Auto Scaling group, Elastic Load Balancing load balancer, and an Amazon RDS database instance. To create those resources, you create a stack by submitting your template that defines those resources, and AWS CloudFormation handles all of the provisioning for you. After all of the resources have been created, AWS CloudFormation reports that your stack has been created. You can then start using the resources in your stack. If stack creation fails, AWS CloudFormation rolls back your changes by deleting the resources that it created.

Often you will need to launch stacks from the same template, but with minor variations, such as within a different Amazon VPC or using AMIs from a different region. These variations can be addressed using parameters. You can use parameters to customize aspects of your template at runtime, when the stack is built. For example, you can pass the Amazon RDS database size, Amazon EC2 instance types, database, and web server port numbers to AWS CloudFormation when you create a stack. By leveraging template parameters, you can use a single template for many infrastructure deployments with different configuration values. For example, your Amazon EC2 instance types, Amazon CloudWatch alarm thresholds, and Amazon RDS read-replica settings may differ among AWS regions if you receive more customer traffic in the United States than in Europe. You can use template parameters to tune the settings and thresholds in each region separately and still be sure that the application is deployed consistently across the regions.

Figure 11.8 depicts the AWS CloudFormation workflow for creating stacks.

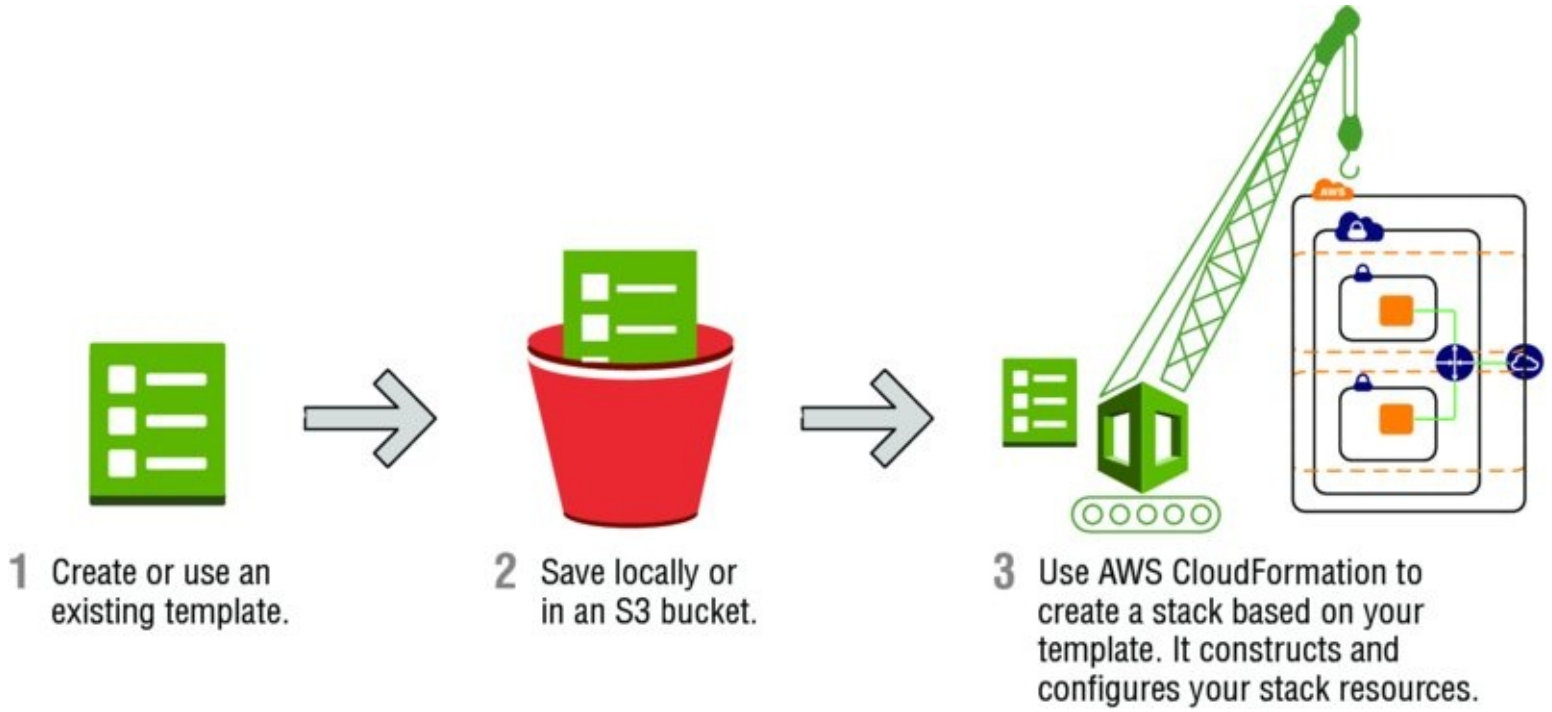

FIGURE 11.8 Creating a stack workflow

Because environments are dynamic in nature, you inevitably will need to update your stack's resources from time to time. There is no need to create a new stack and delete the old one; you can simply modify the existing stack's template. To update a stack, create a *change set* by submitting a modified version of the original stack template, different input parameter values, or both. AWS CloudFormation compares the modified template with the original template and generates a change set. The change set lists the proposed changes. After reviewing the changes, you can execute the change set to update your stack. Figure 11.9 depicts the workflow for updating a stack.

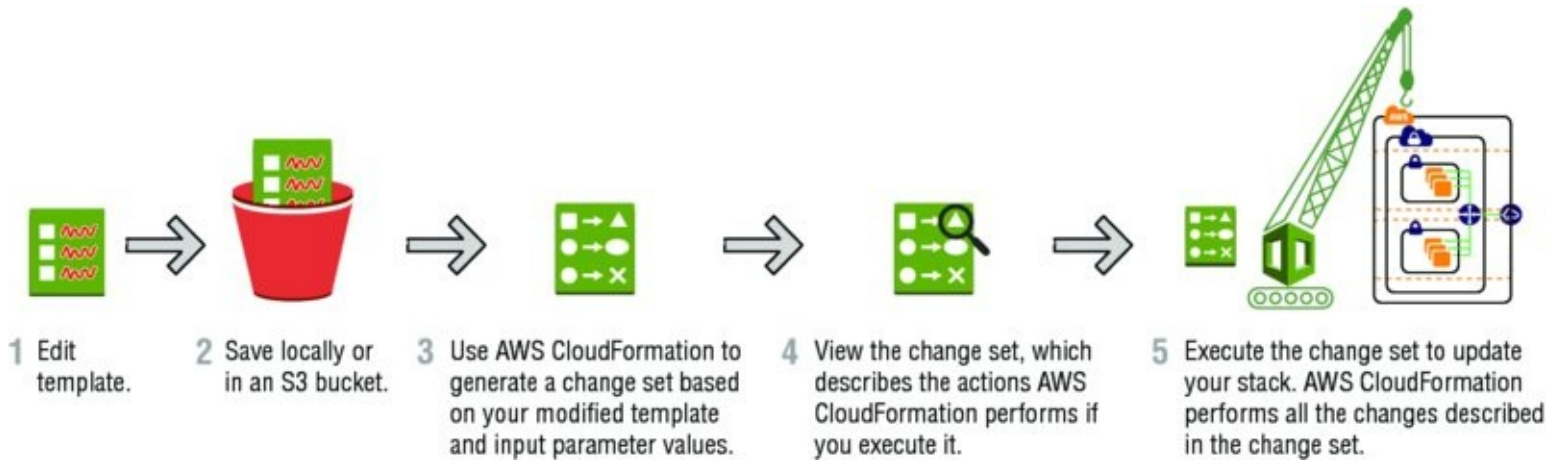

FIGURE 11.9 Updating a stack workflow

When the time comes and you need to delete a stack, AWS CloudFormation deletes the stack and all of the resources in that stack.

If you want to delete a stack but still retain some resources in that stack, you can use a deletion policy to retain those resources. If a resource has no deletion policy, AWS CloudFormation deletes the resource by default.

After all of the resources have been deleted, AWS CloudFormation signals that your stack has been successfully deleted. If AWS CloudFormation cannot delete a resource, the stack will not be deleted. Any resources that haven't been deleted will remain until you can successfully delete the stack.

Use Case

By allowing you to replicate your entire infrastructure stack easily and quickly, AWS CloudFormation enables a variety of use cases, including, but not limited to:

Quickly Launch New Test Environments AWS CloudFormation lets testing teams quickly create a clean environment to run tests without disturbing ongoing efforts in other environments.

Reliably Replicate Configuration Between Environments Because AWS CloudFormation scripts the entire environment, human error is eliminated when creating new stacks.

Launch Applications in New AWS Regions A single script can be used across multiple regions to launch stacks reliably in different markets.

AWS Elastic Beanstalk

AWS Elastic Beanstalk is the fastest and simplest way to get an application up and running on AWS. Developers can simply upload their application code, and the service automatically handles all of the details, such as resource provisioning, load balancing, Auto Scaling, and monitoring.

Overview

AWS comprises dozens of building block services, each of which exposes an area of functionality. While the variety of services offers flexibility for how organizations want to manage their AWS infrastructure, it can be challenging to figure out which services to use and how to provision them. With AWS Elastic Beanstalk, you can quickly deploy and manage applications on the AWS cloud without worrying about the infrastructure that runs those applications. AWS Elastic Beanstalk reduces management complexity without restricting choice or control.

There are key components that comprise AWS Elastic Beanstalk and work together to provide the necessary services to deploy and manage applications easily in the cloud. An *AWS Elastic Beanstalk application* is the logical collection of these AWS Elastic Beanstalk components, which includes environments, versions, and environment configurations. In AWS Elastic Beanstalk, an application is conceptually similar to a folder.

An *application version* refers to a specific, labeled iteration of deployable code for a web application. An application version points to an Amazon S3 object that contains the deployable code. Applications can have many versions and each application version is unique. In a running environment, organizations can deploy any application version they already uploaded to the application, or they can upload and immediately deploy a new application version. Organizations might upload multiple application versions to test differences between one version of their web application and another.

An *environment* is an application version that is deployed onto AWS resources. Each environment runs only a single application version at a time; however, the same version or different versions can run in as many environments at the same time as needed. When an environment is created, AWS Elastic Beanstalk provisions the resources needed to run the application version that is specified.

An *environment configuration* identifies a collection of parameters and settings that define how an environment and its associated resources behave. When an environment's configuration settings are updated, AWS Elastic Beanstalk automatically applies the changes to existing resources or deletes and deploys new resources depending on the type of change.

When an AWS Elastic Beanstalk environment is launched, the environment tier, platform, and environment type are specified. The environment tier that is chosen determines whether AWS Elastic Beanstalk provisions resources to support a web application that handles HTTP(S) requests or an application that handles background-processing tasks. An environment tier whose web application processes web requests is known as a *web server tier*. An environment tier whose application runs background jobs is known as a *worker tier*.

At the time of this writing, AWS Elastic Beanstalk provides platform support for the programming languages Java, Node.js, PHP, Python, Ruby, and Go with support for the web containers Tomcat, Passenger, Puma, and Docker.

Use Cases

A company provides a website for prospective home buyers, sellers, and renters to browse home and apartment listings for more than 110 million homes. The website processes more than three million new images daily. It receives more than 17,000 image requests per second on its website during peak traffic from both desktop and mobile clients.

The company was looking for ways to be more agile with deployments and empower its developers to focus more on writing code instead of spending time managing and configuring servers, databases, load balancers, firewalls, and networks. It began using AWS Elastic Beanstalk as the service for deploying and scaling the web applications and services. Developers were empowered to upload code to AWS Elastic Beanstalk, which then automatically handled the deployment, from capacity provisioning, load balancing, and Auto Scaling, to application health monitoring.

Because the company ingests data in a haphazard way, running feeds that dump a ton of work into the image processing system all at once, it needs to scale up its image converter fleet to meet peak demand. The company determined that an AWS Elastic Beanstalk worker fleet to run a Python Imaging Library with custom code was the simplest way to meet the requirement. This eliminated the need to have a number of static instances or, worse, trying to write their own Auto Scaling configuration.

By making the move to AWS Elastic Beanstalk, the company was able to reduce operating costs while increasing agility and scalability for its image processing and delivery system.

Key Features

AWS Elastic Beanstalk provides several management features that ease deployment and management of applications on AWS. Organizations have access to built-in Amazon CloudWatch monitoring metrics such as average CPU utilization, request count, and average

latency. They can receive email notifications through Amazon SNS when application health changes or application servers are added or removed. Server logs for the application servers can be accessed without needing to log in. Organizations can even elect to have updates applied automatically to the underlying platform running the application such as the AMI, operating system, language and framework, and application or proxy server.

Additionally, developers retain full control over the AWS resources powering their application and can perform a variety of functions by simply adjusting the configuration settings. These include settings such as:

- Selecting the most appropriate Amazon EC2 instance type that matches the CPU and memory requirements of their application
- Choosing the right database and storage options such as Amazon RDS, Amazon DynamoDB, Microsoft SQL Server, and Oracle
- Enabling login access to Amazon EC2 instances for immediate and direct troubleshooting
- Enhancing application security by enabling HTTPS protocol on the load balancer
- Adjusting application server settings (for example, JVM settings) and passing environment variables
- Adjust Auto Scaling settings to control the metrics and thresholds used to determine when to add or remove instances from an environment

With AWS Elastic Beanstalk, organizations can deploy an application quickly while retaining as much control as they want to have over the underlying infrastructure.

AWS Trusted Advisor

AWS Trusted Advisor draws upon best practices learned from the aggregated operational history of serving over a million AWS customers. AWS Trusted Advisor inspects your AWS environment and makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps. You can view the overall status of your AWS resources and savings estimations on the AWS Trusted Advisor dashboard.

AWS Trusted Advisor is accessed in the AWS Management Console. Additionally, programmatic access to AWS Trusted Advisor is available with the AWS Support API.

AWS Trusted Advisor provides best practices in four categories: cost optimization, security, fault tolerance, and performance improvement. The status of the check is shown by using color coding on the dashboard page, as depicted in Figure 11.10.

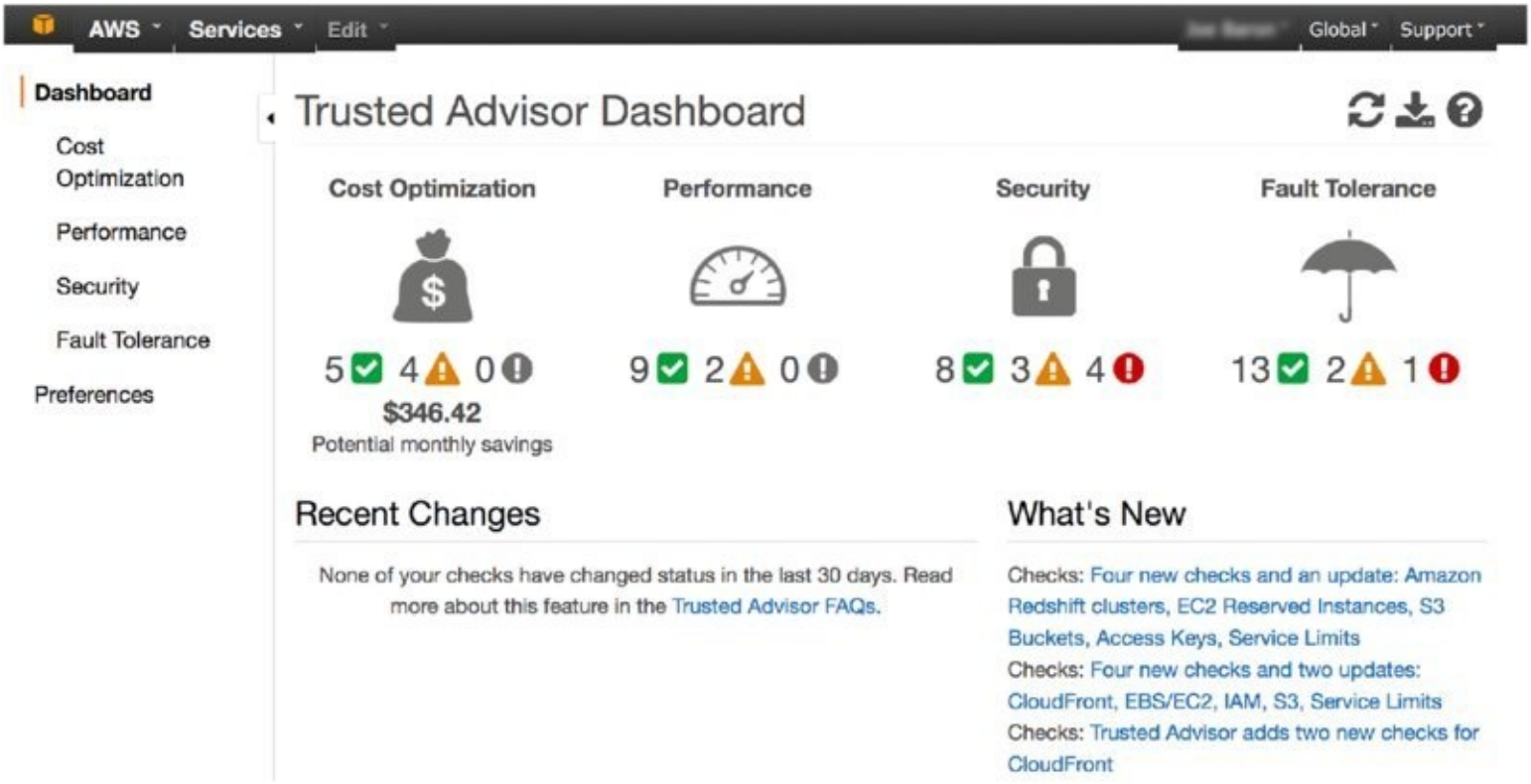

FIGURE 11.10 AWS Trusted Advisor Console dashboard

The color coding reflects the following information:

- *Red*: Action recommended
- *Yellow*: Investigation recommended
- *Green*: No problem detected

For each check, you can review a detailed description of the recommended best practice, a set of alert criteria, guidelines for action, and a list of useful resources on the topic.

All AWS customers have access to four AWS Trusted Advisor checks at no cost. The four standard AWS Trusted Advisor checks are:

Service Limits Checks for usage that is more than 80 percent of the service limit. These values are based on a snapshot, so current usage might differ and can take up to 24 hours to reflect changes.

Security Groups–Specific Ports Unrestricted Checks security groups for rules that allow unrestricted access (0.0.0.0/0) to specific ports

IAM Use Checks for your use of AWS IAM

MFA on Root Account Checks the root account and warns if MFA is not enabled

Customers with a Business or Enterprise AWS Support plan can view all AWS Trusted Advisor checks—over 50 checks.

There may be occasions when a particular check is not relevant to some resources in your AWS environment. You have the ability to exclude items from a check and optionally restore them later at any time. AWS Trusted Advisor acts like a customized cloud expert, and it helps organizations provision their resources by following best practices while identifying inefficiencies, waste, potential cost savings, and security issues.

AWS Config

AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance. With AWS Config, you can discover existing and deleted AWS resources, determine your overall compliance against rules, and dive into configuration details of a resource at any point in time. These capabilities enable compliance auditing, security analysis, resource change tracking, and troubleshooting.

Overview

AWS Config provides a detailed view of the configuration of AWS resources in your AWS account. This includes how the resources are related and how they were configured in the past so that you can see how the configurations and relationships change over time. AWS Config defines a resource as an entity you can work with in AWS, such as an Amazon EC2 instance, an Amazon EBS volume, a security group, or an Amazon VPC.

When you turn on AWS Config, it first discovers the supported AWS resources that exist in your account and generates a *configuration item* for each resource. A configuration item represents a point-in-time view of the various attributes of a supported AWS resource that exists in your account. The components of a configuration item include metadata, attributes, relationships, current configuration, and related events.

AWS Config will generate configuration items when the configuration of a resource changes, and it maintains historical records of the configuration items of your resources from the time you start the *configuration recorder*. The configuration recorder stores the configurations of the supported resources in your account as configuration items. By default, AWS Config creates configuration items for every supported resource in the region. If you don't want AWS Config to create configuration items for all supported resources, you can specify the resource types that you want it to track.

Organizations often need to assess the overall compliance and risk status from a configuration perspective, view compliance trends over time, and pinpoint which configuration change caused a resource to drift out of compliance. An *AWS Config Rule* represents desired configuration settings for specific AWS resources or for an entire AWS account. While AWS Config continuously tracks your resource configuration changes, it checks whether these changes violate any of the conditions in your rules. If a resource violates a rule, AWS Config flags the resource and the rule as noncompliant and notifies you through Amazon SNS.

AWS Config makes it easy to track resource configuration without the need for up-front investments and while avoiding the complexity of installing and updating agents for data collection or maintaining large databases. Once AWS Config is enabled, organizations can view continuously updated details of all configuration attributes associated with AWS resources.

Use Cases

Some of the infrastructure management tasks AWS Config enables include:

Discovery AWS Config will discover resources that exist in your account, record their

current configuration, and capture any changes to these configurations. AWS Config will also retain configuration details for resources that have been deleted. A comprehensive snapshot of all resources and their configuration attributes provides a complete inventory of resources in your account.

Change Management When your resources are created, updated, or deleted, AWS Config streams these configuration changes to Amazon SNS so that you are notified of all configuration changes. AWS Config represents relationships between resources, so you can assess how a change to one resource may affect other resources.

Continuous Audit and Compliance AWS Config and AWS Config Rules are designed to help you assess compliance with internal policies and regulatory standards by providing visibility into the configuration of a resource at any time and evaluating relevant configuration changes against rules that you can define.

Troubleshooting Using AWS Config, you can quickly troubleshoot operational issues by identifying the recent configuration changes to your resources.

Security and Incident Analysis Properly configured resources improve your security posture. Data from AWS Config enables you to monitor the configurations of your resources continuously and evaluate these configurations for potential security weaknesses. After a potential security event, AWS Config enables you to examine the configuration of your resources at any single point in the past.

Key Features

In the past, organizations needed to poll resource APIs and maintain their own external database for change management. AWS Config resolves this previous need and automatically records resource configuration information and will evaluate any rules that are triggered by a change. The configuration of the resource and its overall compliance against rules are presented in a dashboard.

AWS Config integrates with AWS CloudTrail, a service that records AWS API calls for an account and delivers API usage log files to an Amazon S3 bucket. If the configuration change of a resource was the result of an API call, AWS Config also records the AWS CloudTrail event ID that corresponds to the API call that changed the resource's configuration. Organizations can then leverage the AWS CloudTrail logs to obtain details of the API call that was made—including who made the API call, at what time, and from which IP address—to use for troubleshooting purposes.

When a configuration change is made to a resource or when the compliance of an AWS Config rule changes, a notification message is delivered that contains the updated configuration of the resource or compliance state of the rule and key information such as the old and new values for each changed attribute. Additionally, AWS Config sends notifications when a *Configuration History* file is delivered to Amazon S3 and when the customer initiates a *Configuration Snapshot*. These messages are all streamed to an Amazon SNS topic that you specify.

Organizations can use the AWS Management Console, API, or AWS CLI to obtain details of what a resource's configuration looked like at any point in the past. AWS Config will also automatically deliver a history file to the Amazon S3 bucket you specify every six hours that

contains all changes to your resource configurations.

Summary

In this chapter, you learned about additional key AWS cloud services, many of which will be covered on your AWS Certified Solutions Architect – Associate exam. These services are grouped into four categories of services: storage and content delivery, security, analytics, and DevOps.

In the storage and content delivery group, we covered Amazon CloudFront and AWS Storage Gateway. Amazon CloudFront is a global CDN service. It integrates with other AWS products to give developers and businesses an easy way to distribute content to end users with low latency, high data transfer speeds, and no minimum usage commitments. AWS Storage Gateway is a service that connects an on-premises software appliance with cloud-based storage. It provides seamless and secure integration between an organization's on-premises IT environment and AWS storage infrastructure. The AWS Storage Gateway appliance maintains frequently accessed data on-premises while encrypting and storing all of your data in Amazon S3 or Amazon Glacier.

The services we covered in security focused on Identity Management (AWS Directory Service), Key Management (AWS KMS AWS CloudHSM), and Audit (AWS CloudTrail). AWS Directory Service is a managed service offering, providing directories that contain information about your organization, including users, groups, computers, and other resources. AWS Directory Service is offered in three types: AWS Directory Service for Microsoft Active Directory (Enterprise Edition), Simple AD, and AD Connector.

Key management is the management of cryptographic keys within a cryptosystem. This includes dealing with the generation, exchange, storage, use, and replacement of keys. AWS KMS is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data. AWS KMS lets you create keys that can never be exported from the service and that can be used to encrypt and decrypt data based on policies you define. AWS CloudHSM helps you meet corporate, contractual, and regulatory compliance requirements for data security by using dedicated HSM appliances within the AWS cloud. An HSM is a hardware appliance that provides secure key storage and cryptographic operations within a tamper-resistant hardware module.

Rounding out the security services is AWS CloudTrail. AWS CloudTrail provides visibility into user activity by recording API calls made on your account. AWS CloudTrail records important information about each API call, including the name of the API, the identity of the caller, the time of the API call, the request parameters, and the response elements returned by the AWS service. This information helps you to track changes made to your AWS resources and to troubleshoot operational issues.

The analytics services covered help you overcome the unique list of challenges associated with big data in today's IT world. Amazon Kinesis is a platform for handling massive streaming data on AWS, offering powerful services to make it easy to load and analyze streaming data and also providing the ability for you to build custom streaming data applications for specialized needs. Amazon EMR provides you with a fully managed, on-demand Hadoop framework. The reduction of complexity and up-front costs combined with the scale of AWS means you can instantly spin up large Hadoop clusters and start processing

within minutes.

To supplement the big data challenges, orchestrating data movement comes with its own challenges. AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, and also on-premises data sources, at specified intervals. With AWS Data Pipeline, you can regularly access your data where it's stored, transform and process it at scale, and efficiently transfer the results to AWS services such as Amazon S3, Amazon RDS, Amazon DynamoDB, and Amazon EMR. Additionally, AWS Import/Export helps when you're faced with the challenge of getting huge datasets into AWS in the first place or retrieving them back to on-premises when necessary. AWS Import/Export is a service that accelerates transferring large amounts of data into and out of AWS using physical storage appliances, bypassing the Internet. The data is copied to a device at the source, shipped via standard shipping mechanisms, and then copied to the destination.

AWS continues to evolve services in support of organizations embracing DevOps. Services such as AWS OpsWorks, AWS CloudFormation, AWS Elastic Beanstalk, and AWS Config are leading the way for DevOps on AWS. AWS OpsWorks provides a configuration management service that helps you configure and operate applications using Chef. AWS OpsWorks works with applications of any level of complexity and is independent of any particular architectural pattern. AWS CloudFormation allows organizations to deploy, modify, and update resources in a controlled and predictable way, in effect applying version control to AWS infrastructure the same way one would do with software. AWS Elastic Beanstalk allows developers to simply upload their application code, and the service automatically handles all of the details such as resource provisioning, load balancing, Auto Scaling, and monitoring. AWS Config delivers a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance. With AWS Config, organizations have the information necessary for compliance auditing, security analysis, resource change tracking, and troubleshooting.

The key additional services covered in this chapter will help you form a knowledge base to understand the necessities for the exam. As you continue to grow as a Solutions Architect, diving deeper into the AWS cloud services as a whole will expand your ability to define well architected solutions across a wide variety of business verticals and use cases.

Exam Essentials

Know the basic use cases for amazon CloudFront. Know when to use Amazon CloudFront (for popular static and dynamic content with geographically distributed users) and when not to (all users at a single location or connecting through a corporate VPN).

Know how amazon CloudFront works. Amazon CloudFront optimizes downloads by using geolocation to identify the geographical location of users, then serving and caching content at the edge location closest to each user to maximize performance.

Know how to create an amazon CloudFront distribution and what types of origins are supported. To create a distribution, you specify an origin and the type of distribution, and Amazon CloudFront creates a new domain name for the distribution. Origins supported include Amazon S3 buckets or static Amazon S3 websites and HTTP servers located in Amazon EC2 or in your own data center.

Know how to use amazon CloudFront for dynamic content and multiple origins. Understand how to specify multiple origins for different types of content and how to use cache behaviors and path strings to control what content is served by which origin.

Know what mechanisms are available to serve private content through amazon CloudFront. Amazon CloudFront can serve private content using Amazon S3 Origin Access Identifiers, signed URLs, and signed cookies.

Know the three configurations of AWS storage gateway and their use cases. Gateway-Cached volumes expand your on-premises storage into Amazon S3 and cache frequently used files locally. Gateway-Stored values keep all your data available locally at all times and also replicate it asynchronously to Amazon S3. Gateway-VTL enables you to keep your current backup tape software and processes while eliminating physical tapes by storing your data in the cloud.

Understand the value of AWS Directory Service. AWS Directory Service is designed to reduce identity management tasks, thereby allowing you to focus more of your time and resources on your business.

Know the AWS Directory Service Directory types. AWS Directory Service offers three directory types:

- AWS Directory Service for Microsoft Active Directory (Enterprise Edition), also referred to as Microsoft AD
- Simple AD
- AD Connector

Know when you should use AWS Directory Service for Microsoft Active Directory. You should use Microsoft Active Directory if you have more than 5,000 users or need a trust relationship set up between an AWS hosted directory and your on-premises directories.

Understand key management. Key management is the management of cryptographic keys within a cryptosystem. This includes dealing with the generation, exchange, storage, use,

and replacement of keys.

Understand when you should use AWS KMS. AWS KMS is a managed service that makes it easy for you to create and control the symmetric encryption keys used to encrypt your data. AWS KMS lets you create keys that can never be exported from the service and which can be used to encrypt and decrypt data based on policies you define.

Understand when you should use AWS CloudHSM. AWS CloudHSM helps you meet corporate, contractual, and regulatory compliance requirements for data security by using dedicated hardware security module appliances within the AWS cloud.

Understand the value of AWS CloudTrail. AWS CloudTrail provides visibility into user activity by recording API calls made on your account. This helps you to track changes made to your AWS resources and to troubleshoot operational issues. AWS CloudTrail makes it easier to ensure compliance with internal policies and regulatory standards.

Know the three services of Amazon kinesis and their use cases. Amazon Kinesis Firehose allows you to load massive volumes of streaming data into AWS. Amazon Kinesis Analytics enables you to easily analyze streaming data real time with standard SQL. Amazon Kinesis Streams enables you to build custom applications that process or analyze streaming data real time for specialized needs.

Know what service Amazon EMR provides. Amazon EMR provides a managed Hadoop service on AWS that allows you to spin up large Hadoop clusters in minutes.

Know the difference between persistent and transient clusters. Persistent clusters run continuously, so they do not lose data stored on instance-based HDFS. Transient clusters are launched for a specific task, then terminated, so they access data on Amazon S3 via EMRFS.

Know the use cases for Amazon EMR. Amazon EMR is useful for big data analytics in virtually any industry, including, but not limited to, log processing, clickstream analysis, and genomics and life sciences.

Know the use cases for AWS data pipeline. AWS Data Pipeline can manage batch ETL processes at scale on the cloud, accessing data both in AWS and on-premises. It can take advantage of AWS cloud services by spinning up resources required for the process, such as Amazon EC2 instances or Amazon EMR clusters.

Know the types of AWS import/export services and the possible sources/destinations of each. AWS Snowball is Amazon shippable appliances supplied ready to ship. It can transfer data to and from your on-premises storage and to and from Amazon S3. AWS Import/Export Disk uses your storage devices and, in addition to transferring data in and out of your on-premises storage, can import data to Amazon S3, Amazon EBS, and Amazon S3; it can only export data from Amazon S3.

Understand the basics of AWS opsworks. AWS OpsWorks is a configuration management service that helps you configure and operate applications of all shapes and sizes using Chef. You can define an application's architecture and the specification of each component including package installation, software configuration, and resources such as storage.

Understand the value of AWS cloudformation. AWS CloudFormation is a service that

helps you model and set up your AWS resources. AWS CloudFormation allows organizations to deploy, modify, and update resources in a controlled and predictable way, in effect applying version control to AWS infrastructure the same way you would do with software.

Understand the value of AWS elastic beanstalk. AWS Elastic Beanstalk is the fastest and simplest way to get an application up and running on AWS. Developers can simply upload their application code, and the service automatically handles all the details such as resource provisioning, load balancing, Auto Scaling, and monitoring.

Understand the components of AWS elastic beanstalk. An AWS Elastic Beanstalk application is the logical collection of environments, versions, and environment configurations. In AWS Elastic Beanstalk, an application is conceptually similar to a folder.

Understand the value of AWS config. AWS Config is a fully managed service that provides organizations with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance. With AWS Config, organizations can discover existing and deleted AWS resources, determine their overall compliance against rules and dive into configuration details of a resource at any point in time. These capabilities enable compliance auditing, security analysis, resource change tracking, and troubleshooting.

Review Questions

1. What origin servers are supported by Amazon CloudFront? (Choose 3 answers)
   A. An Amazon Route 53 Hosted Zone
   B. An Amazon Simple Storage Service (Amazon S3) bucket
   C. An HTTP server running on Amazon Elastic Compute Cloud (Amazon EC2)
   D. An Amazon EC2 Auto Scaling Group
   E. An HTTP server running on-premises
2. Which of the following are good use cases for Amazon CloudFront? (Choose 2 answers)
   A. A popular software download site that supports users around the world, with dynamic content that changes rapidly
   B. A corporate website that serves training videos to employees. Most employees are located in two corporate campuses in the same city.
   C. A heavily used video and music streaming service that requires content to be delivered only to paid subscribers
   D. A corporate HR website that supports a global workforce. Because the site contains sensitive data, all users must connect through a corporate Virtual Private Network (VPN).
3. You have a web application that contains both static content in an Amazon Simple Storage Service (Amazon S3) bucket—primarily images and CSS files—and also dynamic content currently served by a PHP web app running on Amazon Elastic Compute Cloud (Amazon EC2). What features of Amazon CloudFront can be used to support this application with a single Amazon CloudFront distribution?
4. (Choose 2 answers)
   A. Multiple Origin Access Identifiers
   B. Multiple signed URLs
   C. Multiple origins
   D. Multiple edge locations
   E. Multiple cache behaviors
5. You are building a media-sharing web application that serves video files to end users on both PCs and mobile devices. The media files are stored as objects in an Amazon Simple Storage Service (Amazon S3) bucket, but are to be delivered through Amazon CloudFront. What is the simplest way to ensure that only Amazon CloudFront has access to the objects in the Amazon S3 bucket?
   A. Create Signed URLs for each Amazon S3 object.
   B. Use an Amazon CloudFront Origin Access Identifier (OAI).

C. Use public and private keys with signed cookies.

D. Use an AWS Identity and Access Management (IAM) bucket policy.

6. Your company data center is completely full, but the sales group has determined a need to store 200TB of product video. The videos were created over the last several years, with the most recent being accessed by sales the most often. The data must be accessed locally, but there is no space in the data center to install local storage devices to store this data. What AWS cloud service will meet sales' requirements?

   A. AWS Storage Gateway Gateway-Stored volumes

   B. Amazon Elastic Compute Cloud (Amazon EC2) instances with attached Amazon EBS Volumes

   C. AWS Storage Gateway Gateway-Cached volumes

   D. AWS Import/Export Disk

7. Your company wants to extend their existing Microsoft Active Directory capability into an Amazon Virtual Private Cloud (Amazon VPC) without establishing a trust relationship with the existing on-premises Active Directory. Which of the following is the best approach to achieve this goal?

   A. Create and connect an AWS Directory Service AD Connector.

   B. Create and connect an AWS Directory Service Simple AD.

   C. Create and connect an AWS Directory Service for Microsoft Active Directory (Enterprise Edition).

   D. None of the above

8. Which of the following are AWS Key Management Service (AWS KMS) keys that will never exit AWS unencrypted?

   A. AWS KMS data keys

   B. Envelope encryption keys

   C. AWS KMS Customer Master Keys (CMKs)

   D. A and C

9. Which cryptographic method is used by AWS Key Management Service (AWS KMS) to encrypt data?

   A. Password-based encryption

   B. Asymmetric

   C. Shared secret

   D. Envelope encryption

10. Which AWS service records Application Program Interface (API) calls made on your account and delivers log files to your Amazon Simple Storage Service (Amazon S3) bucket?

    A. AWS CloudTrail

B. Amazon CloudWatch

C. Amazon Kinesis

D. AWS Data Pipeline

11. You are trying to decrypt ciphertext with AWS KMS and the decryption operation is failing. Which of the following are possible causes? (Choose 2 answers)

A. The private key does not match the public key in the ciphertext.

B. The plaintext was encrypted along with an encryption context, and you are not providing the identical encryption context when calling the Decrypt API.

C. The ciphertext you are trying to decrypt is not valid.

D. You are not providing the correct symmetric key to the Decrypt API.

12. Your company has 30 years of financial records that take up 15TB of on-premises storage. It is regulated that you maintain these records, but in the year you have worked for the company no one has ever requested any of this data. Given that the company data center is already filling the bandwidth of its Internet connection, what is an alternative way to store the data on the most appropriate cloud storage?

A. AWS Import/Export to Amazon Simple Storage Service (Amazon S3)

B. AWS Import/Export to Amazon Glacier

C. Amazon Kinesis

D. Amazon Elastic MapReduce (AWS EMR)

13. Your company collects information from the point of sale registers at all of its franchise locations. Each month these processes collect 200TB of information stored in Amazon Simple Storage Service (Amazon S3). Analytics jobs taking 24 hours are performed to gather knowledge from this data. Which of the following will allow you to perform these analytics in a cost-effective way?

A. Copy the data to a persistent Amazon Elastic MapReduce (Amazon EMR) cluster, and run the MapReduce jobs.

B. Create an application that reads the information of the Amazon S3 bucket and runs it through an Amazon Kinesis stream.

C. Run a transient Amazon EMR cluster, and run the MapReduce jobs against the data directly in Amazon S3.

D. Launch a d2.8xlarge (32 vCPU, 244GB RAM) Amazon Elastic Compute Cloud (Amazon EC2) instance, and run an application to read and process each object sequentially.

14. Which service allows you to process nearly limitless streams of data in flight?

A. Amazon Kinesis Firehose

B. Amazon Elastic MapReduce (Amazon EMR)

C. Amazon Redshift

D. Amazon Kinesis Streams

15. What combination of services enable you to copy daily 50TB of data to Amazon storage, process the data in Hadoop, and store the results in a large data warehouse?
    A. Amazon Kinesis, Amazon Data Pipeline, Amazon Elastic MapReduce (Amazon EMR), and Amazon Elastic Compute Cloud (Amazon EC2)
    B. Amazon Elastic Block Store (Amazon EBS), Amazon Data Pipeline, Amazon EMR, and Amazon Redshift
    C. Amazon Simple Storage Service (Amazon S3), Amazon Data Pipeline, Amazon EMR, and Amazon Redshift
    D. Amazon S3, Amazon Simple Workflow, Amazon EMR, and Amazon DynamoDB
16. Your company has 50,000 weather stations around the country that send updates every 2 seconds. What service will enable you to ingest this stream of data and store it to Amazon Simple Storage Service (Amazon S3) for future processing?
    A. Amazon Simple Queue Service (Amazon SQS)
    B. Amazon Kinesis Firehose
    C. Amazon Elastic Compute Cloud (Amazon EC2)
    D. Amazon Data Pipeline
17. Your organization uses Chef heavily for its deployment automation. What AWS cloud service provides integration with Chef recipes to start new application server instances, configure application server software, and deploy applications?
    A. AWS Elastic Beanstalk
    B. Amazon Kinesis
    C. AWS OpsWorks
    D. AWS CloudFormation
18. A firm is moving its testing platform to AWS to provide developers with instant access to clean test and development environments. The primary requirement for the firm is to make environments easily reproducible and fungible. What service will help the firm meet their requirements?
    A. AWS CloudFormation
    B. AWS Config
    C. Amazon Redshift
    D. AWS Trusted Advisor
19. Your company's IT management team is looking for an online tool to provide recommendations to save money, improve system availability and performance, and to help close security gaps. What can help the management team?
    A. Cloud-init
    B. AWS Trusted Advisor

C. AWS Config

D. Configuration Recorder

20. Your company works with data that requires frequent audits of your AWS environment to ensure compliance with internal policies and best practices. In order to perform these audits, you need access to historical configurations of your resources to evaluate relevant configuration changes. Which service will provide the necessary information for your audits?

    A. AWS Config

    B. AWS Key Management Service (AWS KMS)

    C. AWS CloudTrail

    D. AWS OpsWorks

21. All of the website deployments are currently done by your company's development team. With a surge in website popularity, the company is looking for ways to be more agile with deployments. What AWS cloud service can help the developers focus more on writing code instead of spending time managing and configuring servers, databases, load balancers, firewalls, and networks?

    A. AWS Config

    B. AWS Trusted Advisor

    C. Amazon Kinesis

    D. AWS Elastic Beanstalk