AWS Sem-2

Prerequisite:

> **CloudFront:** A content delivery network (CDN) service that securely delivers content such as video, data, applications, and so on.

> **Content Delivery Network:** A network of distributed services that delivers webpages and other web content securely based on the geographic location of the user.

> **Edge Location:** A location where web content is temporarily stored (cached).

> **Origin:** The location where all the objects associated with the webpage are permanently stored.

> **TTL:** The minimum and maximum length of time to cache content at an edge location.

> **Amazon EC2:** A web service that provides secure, resizable compute capacity in the cloud. Think of it as renting a computer in the cloud.

Module 3:

```
One of the main benefits of cloud technology is the ability to pay for just what you n
eed and to pay for things as you use them. To make this possible, the cost for VMs in
 AWS is broken down into the states that instances go through during their lifetime. W
hen you launch an instance in the cloud, a server will progress through various state
s.

When you initially launch an instance, it enters the pending state. This means that AW
S is getting your instance ready, and you cannot access it yet. After the instance is
 ready for you, it enters the running state. You can connect to your running instance
 and use it the way that you would use a computer in front of you.

As soon as your instance transitions to the running state, you are billed for each sec
ond, with a 1-minute minimum, that you keep the instance running, even if the instance
remains idle and you do not connect to it.

When you no longer need an instance, you can stop or terminate it. As soon as the stat
us of an instance changes to shutting down or terminated, you stop incurring charges f
or that instance. You can stop or hibernate an instance and restart it later. However,
some temporary data might be lost when you do so. Following is a table showing the eff
ects of stopping and restarting an instance.

The following table summarizes the key differences between rebooting, stopping, hibern
ating, and terminating your instance.
```

Characteristic	Reboot	Stop/Start (Amazon EBS Backed Instances Only)	Hibernate (Amazon EBS Backed Instances Only)	Terminate
Host computer	The instance stays on the same host computer.	In most cases, we move the instance to a new host computer. Your instance can stay on the same host computer if there are no problems with the host computer.	In most cases, we move the instance to a new host computer. Your instance can stay on the same host computer if there are no problems with the host computer.	None
Private and public IPv4 addresses	These addresses stay the same.	The instance keeps its private IPv4 address. The instance gets a new public IPv4 address, unless it has an Elastic IP address, which doesn't change during a stop or start.	The instance keeps its private IPv4 address. The instance gets a new public IPv4 address, unless it has an Elastic IP address, which doesn't change during a stop or start.	None
Elastic IP address (IPv4)	The Elastic IP address remains associated with the instance.	The Elastic IP address remains associated with the instance.	The Elastic IP address remains associated with the instance.	The Elastic IP address is disassociated from the instance.
IPv6 address	The address stays the same.	The instance keeps its IPv6 address.	The instance keeps its IPv6 address.	None
Instance store volumes	The data is preserved.	The data is erased.	The data is erased.	The data is erased.
Root device	The volume is	The volume is	The volume is	The volume is

volume	preserved.	preserved.	preserved.	deleted by default.
Random access memory (RAM) (contents of memory)	The RAM is erased.	The RAM is erased.	The RAM is saved to a file on the root volume.	The RAM is erased.
Billing	The instance billing hour doesn't change.	You stop incurring charges for an instance as soon as its state changes to stopping. Each time an instance transitions from stopped to running, we start a new instance billing period, billing a minimum of 1 minute every time you restart your instance.	You incur charges while the instance is in the stopping state but stop incurring charges when the instance is in the stopped state. Each time an instance transitions from stopped to running, we start a new instance billing period, billing a minimum of 1 minute every time you restart your instance.	You stop incurring charges for an instance as soon as its state changes to shutting down.

Module 5:

Cloud Front:

```
CloudFront is a fast CDN service that securely delivers data, videos, applications, an
d application programming interfaces (APIs) to customers globally with low latency and
high transfer speeds, all within a developer-friendly environment. CloudFront is integ
rated with Amazon Web Services (AWS)—physical locations that are directly connected to
the AWS global infrastructure and other AWS services. CloudFront works seamlessly with
services including AWS Shield for distributed denial of service (DDoS) mitigation, Ama
zon Simple Storage Service (Amazon S3), Elastic Load Balancing (ELB) or Amazon Elastic
Compute Cloud (Amazon EC2) as origins for your applications, and Lambda@Edge to run cu
stom code closer to customers’ users and customize the user experience.

The CloudFront CDN is massively scaled and globally distributed. The CloudFront networ
k has 200 points of presence (PoPs), and leverages the highly resilient AWS backbone n
etwork for superior performance and availability for end users.
```

```
CloudFront is a highly secure CDN that provides network- and application-level protect
ion. You can also use configurable features such as AWS Certificate Manager (ACM) to c
reate and manage custom Secure Sockets Layer (SSL) certificates.

CloudFront features can be customized for your specific application requirements. Lamb
da@Edge functions, initiated by CloudFront events, extend custom code across AWS locat
ions worldwide, so users can move even complex application logic closer to end users t
o improve responsiveness.

CloudFront is integrated with AWS services such as Amazon S3, Amazon EC2, ELB, Amazon
 Route 53, and AWS Elemental Media Services. They are all accessible through the same
 console, and all features in the CDN can be programmatically configured by using APIs
or the AWS Management Console:
```

Module 6:

AWS Lambda:

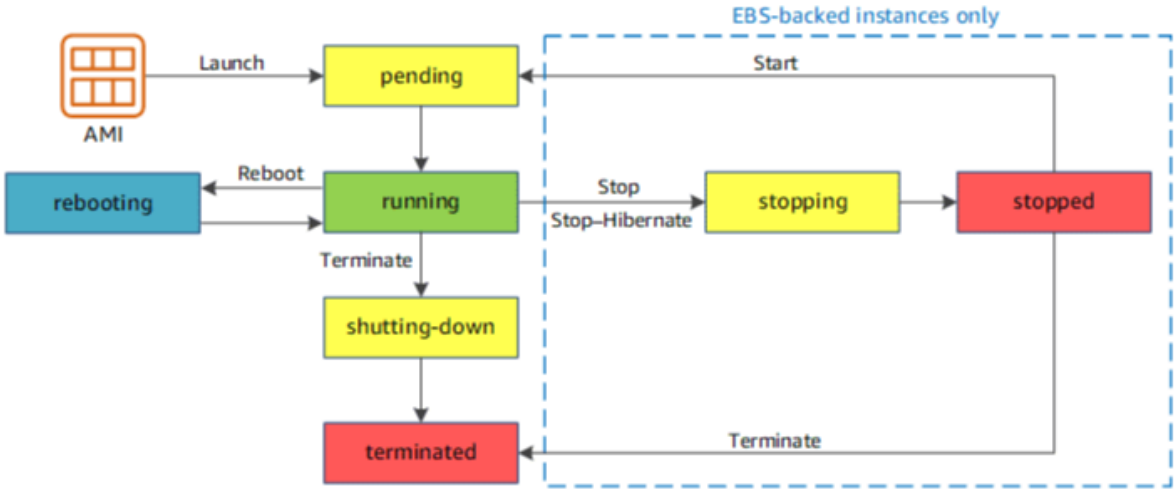
```
Lambda lets you run code without provisioning or managing servers. You pay only for th
e compute time you consume—there is no charge when your code is not running (and you s
tart paying only after the first million requests per month on the AWS Free Tier). Wit
h Lambda, you can run code for virtually any type of application or backend service—al
l with zero administration. Upload your code and Lambda takes care of everything requi
red to run and scale your code with high availability. You can set up your code to aut
omatically be initiated from other AWS services or events, or it can be set up to resp
ond directly to an HTTP or HTTPS request.
```

Difference Between EC2 and Lambda:

```
One major difference between Lambda and Amazon EC2 is the cost of operation. The prici
ng models of the two services are quite different. Lambda cost is based on usage of th
e Lambda functions and the amount of storage used by the functions, whereas Amazon EC2
cost is based on the type of machine image used and the amount and type of storage use
d. For light and medium workloads, Lambda is a dramatically less expensive solution th
an Amazon EC2.

By working with Amazon EC2 to manage your instances from the moment you launch them th
rough their termination, you ensure that your customers have the best possible perform
ance and resulting experience with the applications or sites that you host on your ins
tances.
```

The following illustration represents the transitions between instance states:



EC2 Purchase Options:

```
Amazon EC2 provides the following purchasing options for you to make purchases based o
n your needs:

On-Demand Instances: Pay by the second for the instances that you launch.

Reserved Instances: Purchase, at a significant discount, instances that are always ava
ilable, for a term of 1 to 3 years.

Scheduled Instances: Purchase instances that are always available on the specified rec
urring schedule, for a 1-year term.

Spot Instances: Request unused EC2 instances, which can lower your Amazon EC2 costs si
gnificantly.

Dedicated Hosts: Pay for a physical host that is fully dedicated to running your insta
nces, and bring your existing per-socket, per-core, or per-virtual machine (VM) softwa
re licenses to reduce costs.

Dedicated Instances: Pay by the hour for instances that run on single-tenant hardware.
Capacity Reservations: Reserve capacity for your EC2 instances in a specific Availabil
ity Zone for any duration.
```

```
If you require a capacity reservation, purchase reserved instances or capacity reserva
tions for a specific Availability Zone, or purchase scheduled instances. Spot instance
s are a cost-effective choice if you can be flexible about when your applications run
 and if they can be interrupted. Dedicated hosts or dedicated instances can help you a
ddress compliance requirements and reduce costs by using your existing server-bound so
ftware licenses. For more information, see “Instance purchasing options” (https://doc
s.aws.amazon.com/AWSEC2/latest/UserGuide/instance-purchasing-options.html).

Lambda is a serverless computing environment that makes it easy to run code in respons
e to events such as changes to Amazon Simple Storage Service (Amazon S3) buckets, upda
tes to an Amazon DynamoDB table, or custom events generated by your applications or de
```

vices. With Lambda, you do not have to provision your own instances. Lambda performs a ll the operational and administrative activities on your behalf, including capacity pr ovisioning, monitoring fleet health, applying security patches to the underlying compu te resources, deploying your code, running a web service front end, and monitoring and logging your code. Lambda provides scaling and high availability to your code without additional effort on your part.

Module 7:

Automatic Scaling:

Automatic scaling services monitor your applications and automatically adjust capacity to maintain steady, predictable performance at the lowest possible cost. The term scal ing means that new instances can be created as needed, depending on network traffic, o r existing instances can be augmented with additional storage or compute power to hand le increased network traffic. The latter type of scaling is called vertical scaling.

Amazon EC2 Auto Scaling helps maintain application availability and permits users to a utomatically add or remove EC2 instances according to conditions they define. Users ca n use the fleet management features of Amazon EC2 Auto Scaling to maintain the health and availability of a fleet. They can also use the dynamic and predictive scaling fea tures of Amazon EC2 Auto Scaling to add or remove EC2 instances. Dynamic scaling respo nds to changing demand, and predictive scaling automatically schedules the right numbe r of EC2 instances based on predicted demand. Dynamic scaling and predictive scaling c an be used together to scale faster.

Amazon EC2 Auto Scaling can detect when an instance is unhealthy, terminate it, and re place it with a new one.
Amazon EC2 Auto Scaling helps to ensure that your application always has the right amo unt of compute power and proactively provisions capacity with predictive scaling.
Amazon EC2 Auto Scaling adds instances only when needed and can scale across purchase options to optimize performance and cost.
Whether they are running one EC2 instance or thousands, users can use Amazon EC2 Auto Scaling to detect impaired EC2 instances and unhealthy applications, and replace the instances without direct intervention. This helps to ensure that their application is getting the compute capacity that they expect. Amazon EC2 Auto Scaling will perform th ree main functions to automate fleet management for EC2 instances:

Monitors the health of running instances
Amazon EC2 Auto Scaling helps to ensure that your application is able to receive traffic and that EC2 instances are working properly. Amazon EC2 Auto Scaling periodica lly performs health checks to identify any instances that are unhealthy.

Replaces impaired instances automatically
When an impaired instance fails a health check, Amazon EC2 Auto Scaling automati cally terminates it and replaces it with a new one. That means that you don’t need to respond manually when an instance needs replacing.

Balances capacity across Availability Zones

Amazon EC2 Auto Scaling can automatically balance instances across zones, and it always launches new instances so that they are balanced between zones as evenly as possible across your entire fleet. Load balancing distributes load across existing instances, whereas Auto Scaling creates or removes instances based on demand.

Scheduled Scaling:

When scaling based on a schedule, users can scale their application ahead of known load changes. For example, every week, the traffic to the web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday. Users can plan their scaling activities based on the known traffic patterns of the web application.

Dynamic Scaling:

With Amazon EC2 Auto Scaling, users can follow the demand curve for their applications closely, reducing the need to manually provision Amazon EC2 capacity in advance. For example, users can use target-tracking scaling policies to select a load metric for their application, such as CPU usage. Or they can set a target value using the new Request Count Per Target metric from Application Load Balancer, a load-balancing option for the Elastic Load Balancing service. Amazon EC2 Auto Scaling will then automatically adjust the number of EC2 instances as needed to maintain your target. An example of when dynamic scaling would be useful is when music or video goes viral. Dynamic scaling would sense the increased traffic and spin up new instances to keep pace with the increased demand.

Predictive Scaling:

Predictive scaling, a feature of Amazon EC2 Auto Scaling, uses machine learning (ML) to schedule the right number of EC2 instances in anticipation of approaching traffic changes. Predictive scaling predicts future traffic, including regularly occurring spikes, and provisions the right number of EC2 instances in advance. Predictive scaling’s ML algorithms detect changes in daily and weekly patterns, automatically adjusting their forecasts. This removes the need for manual adjustment of Auto Scaling parameters as cyclicality changes over time, making Auto Scaling simpler to configure. Auto Scaling enhanced with Predictive scaling delivers faster, simpler, and more accurate capacity provisioning, resulting in lower cost and more responsive applications.

Auto Scaling Features:

Automatic resource discovery

AWS Auto Scaling scans and automatically discovers the scalable cloud resources underlying an application, so there’s no need to manually identify these resources one by on

e through individual service interfaces.

Built-in scaling strategies

Using AWS Auto Scaling, users can select one of three predefined optimization strategies designed to optimize performance and costs, or balance the two. If preferred, users can set their own target resource usage. Using their selected scaling strategy, AWS Auto Scaling will create the scaling policies for each resource.

Smart scaling policies

AWS Auto Scaling continually calculates the appropriate scaling adjustments and immediately adds and removes capacity as needed to keep metrics on target. Amazon Web Services (AWS) target tracking scaling policies are self-optimizing and learn actual load patterns to minimize fluctuations in resource capacity. This results in smoother, smarter scaling, and the user pays only for the resources they need.

Launch Template:

A launch template is similar to a launch configuration in that it specifies instance configuration information. Included are the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and the other parameters that are used to launch EC2 instances. However, defining a launch template instead of a launch configuration facilitates users having multiple versions of a template. With versioning, users can create a subset of the full set of parameters and then reuse it to create other templates or template versions. For example, a user can create a default template that defines common configuration parameters such as tags or network configurations, and permit the other parameters to be specified as part of another version of the same template.

Module 8:

AWS DeepLense:

A fully programmable video camera, with tutorials, code, and pretrained models designed to expand ML skills

AI can enhance the customer experience in a contact center, automate content moderation in media, improve health care analytics, forecast demand more accurately, and much more. With AI services from AWS, you can add capabilities such as image and video analysis, natural language, personalized recommendations, virtual assistants, and forecasting to your applications without deep expertise in ML. Each of the services can be used on their own or you can use them in concert to create sophisticated human-like functionality. Either way, you get instant access to fast, high-quality AI tools based on the same technology used to power Amazon’s own businesses.

AI services from AWS include the following:

```
Amazon Comprehend
      Discover insights and relationships in text
Amazon Forecast
      Increase forecast accuracy using ML
Amazon Lex
      Build voice and text chatbots
Amazon Personalize
      Build real-time recommendations into your applications
Amazon Polly
      Turn text into lifelike speech
Amazon Rekognition
      Analyze image and video
Amazon Textract
      Extract text and data from documents
Amazon Translate
      Translate texts with higher accuracy
Amazon Transcribe
      Translate audio files to text

AWS DeepLens helps put ML in the hands of developers—literally—with a fully programmab
le video camera, plus tutorials, code, and pretrained models designed to expand deep-l
earning skills.

AWS DeepLens lets developers of all skill levels get started with deep learning in les
s than 10 minutes by providing sample projects with practical, hands-on examples that
 can start running with a single click.

AWS developers can run any deep-learning framework, including TensorFlow and Caffe. AW
S DeepLens comes preinstalled with a high-performance, efficient, optimized inference
 engine for deep learning using Apache MXNet.

AWS DeepLens integrates with Amazon Rekognition for advanced image analysis, Amazon Sa
geMaker for training models, and Amazon Polly to create speech-enabled projects. The d
evice also connects securely to Amazon Simple Queue Service (Amazon SQS), Amazon Simpl
e Notification Service (Amazon SNS), Amazon Simple Storage Service (Amazon S3), Amazon
DynamoDB, and more.

AWS DeepLens is easy to customize and is fully programmable using AWS Lambda. The deep
-learning models in AWS DeepLens even run as part of a Lambda function, providing a fa
miliar programming environment to experiment with.
```

Module 9:

Impact Of AI:

```
As AI and machine learning (ML) become more prevalent in society, ethical questions ha
ve been posed based on actual and predicted events. In a broader sense, there are conc
erns about how humans treat robots, how robots interact with humans, how AI services m
ake decisions ethically, and how individuals and groups use AI services.

Some examples of specific AI ethical dilemmas are:

The human rights of AI robots
```

```
Ethical considerations caused by robotic replacement of care positions
  AI medical diagnoses and treatments
  AI police work

The transparency and accountability of AI makers

Biases placed into AI facial and voice recognition
  Racial bias
  Gender bias

Biases in algorithms, such as language learning

Liability of automated vehicles

  You might be confused by examples of AI you have seen in movies and television, whic
h often depict robots as highly intelligent and self-aware beings. Current AI products
and services mimic human conversation, but are not at the level portrayed in movies. T
his is exemplified by the ELIZA effect, where humans tend to perceive computer systems
as having greater human-like qualities than they actually possess. For example, a pers
on using an automated chatbot to troubleshoot phone hardware might begin to treat the
 chatbot as a real person and ascribe emotional motivations to them.
```

Module 10:

```
ML is a specialized form of AI. ML refers to the ability of computers or networks to c
reate and update algorithms to make predictions or perform tasks with increasingly bet
ter ability. By applying more data to models, the computer system increases ability an
d seemingly learns on its own. ML can be contrasted to computer systems where the prog
rammer enters more information and directions for the computer system to follow. A dif
ferent category of ML is called reinforcement learning. In this type of ML, the algori
thm is initially coded to achieve a goal. Then the computer system experiences rewards
or positive reinforcement as it learns to become more efficient.

The applications of ML cut across many sectors of the economy. Some current areas of u
se include:

Recommendations: This entails using user data patterns to personalize recommendations.
These are familiar as advertising, news articles, or any type of suggestion to a user.

Forecasting: Current conditions are compared to data trends to make predictions.

Recognition: Computer systems are used to recognize patterns in images, video, text, o
r sound.

Fraud detection: Banks use ML algorithms to detect credit card fraud by analyzing purc
hasing patterns.

Transportation: ML is used in many ways in transportation. Mapping apps analyze data f
rom traffic to suggest best routes. Self-driving cars adapt to recognize information f
rom their sensors and react.
```

```
Health care: ML systems can be trained to diagnose illness.

Procedures in ML vary based on the task. However, some general principles are followed
in most programming.

Data: First, acquire quality data from a data lake or other sources.

Model: Create an algorithm or model to make a decision or complete an action.

Training: Give the computer system data to apply to the model and learn from.

Evaluation: After the computer system has had sufficient data points to get the model
 to
operate, it is time to test the algorithm. Give the computer new data and see if it ca
n reach the correct result.

Fine tuning: Continue to give the system more data for training and update the algorit
hm as necessary.

Software companies create applications that can help you to create ML programming. Ama
zon SageMaker is the main Amazon Web Services (AWS) ML service. For information about
 SageMaker, see https://aws.amazon.com/sagemaker/.

You might have some misconceptions about ML. Let’s correct these errors in reasoning.

AI and ML might be mistaken for living or sentient beings. However, ML is an effort to
get computers to function somewhat like a human brain—in fact, sometimes ML systems ar
e called neural networks because of this similarity. Nonetheless, the core of ML is di
gital and nonliving.

Computer systems are sometimes considered superior learners or thinkers to people. Tho
ugh this can be true in speed of processing, common human errors can also exist in ML.
The garbage in, garbage out principle is important here. The ML system still relies on
algorithms and data that are affected by underlying flaws. For example, a bank might b
e trying to teach a machine to recognize fraud. If customers do a poor job of reportin
g fraud or file untruthful reports, the data will cause the computer system to incorre
ctly identify fraud.
```

Module 11:

> C**hrun:** A mathematical model that helps a business determine how long a customer will stay and how much they might purchase during that time.

```
ML is a specialized form of AI. ML refers to the ability of computers or networks to c
reate and update algorithms to make predictions or perform tasks with increasingly bet
```

ter ability. By applying more and more data to models, the computer system increases a bility and seemingly learns on its own. ML can be contrasted to computer systems where the programmer enters more information and directions for the computer system to follo w.

The applications of ML cut across many sectors of the economy. Some current areas of u se include:

Recommendations: Using user data patterns to personalize recommendations. These are fa miliar as advertising, news articles, or any type of suggestion to a user.

Forecasting: Comparing current conditions to data trends to make predictions.
Recognition: Using computer systems to recognize patterns in images, video, or sound.

Pattern detection: Banks use ML algorithms to detect credit card fraud by analyzing pu rchasing patterns.

Customer churn: Analyzing customer information to determine how to attract and keep lo ng-term customers, or predict the lifetime spending of customers.

ML is a form of AI. AI is any computer function that mimics human behavior. The ML cla ss of AI is special because ML computer systems continue to learn from data or feedbac k as they operate. The strongest form of ML is called deep learning. Deep learning inc ludes many layers of analysis that make sense of unstructured data. This method often imitates the patterns of human brains, so they can be called artificial neural networ ks (ANN).

ML plays a very important role in the business world. For example:

Marketing: Companies use ML to create personalized advertisements and recommendations to customers.

Financial: Financial companies use ML to predict trends in markets and prices.

Healthcare: Corporations can use ML and customer data to help make decisions about ris k
factors, diagnosis, treatment, and follow-up care.

Security: ML can be used to detect fraud or malicious bots. ML recognition models can use features such as fingerprints or iris scans to unlock secured sites.

Module 12:

What is the IoT?

The IoT is a term used to describe the network of connected devices capable of working together to improve services for consumers and businesses. IoT cloud services record a nd analyze data from sensors on connected devices to improve function, increase effici ency, or provide help to the user.

The types of devices are wide ranging, including phones, watches, automobiles, machine ry, and appliances. Any of these can be outfitted with sensors and a connection to the cloud. The sensors record information about the activity of the device and transmit th

```
e information to the cloud. Then advanced analytic algorithms are used to make decisio
ns and recommendations or to make changes in the device function.

Examples that you might be familiar with include linked devices, smart home features,
 and wearables. You might wear a device that uses sensors to collect data about your p
ersonal health. Additionally, you might be familiar with phone applications that share
location data and let family members keep track of one another’s whereabouts. More com
plex examples include home systems that connect entertainment devices, lights, securit
y, or appliances. Such consumer products are only a small segment of the IoT—there are
also many industrial and commercial uses.

The IoT handles vast amounts of consumer and business data and devices. Therefore, sec
urity and privacy concerns must be considered along with increasing IoT usage. Cyberse
curity companies often specialize in keeping businesses safe from hacking and cyberatt
acks through the IoT and keeping customer data secure.

What IoT services does AWS offer?

AWS IoT provides device software, control services, and data services. Device software
lets you securely connect devices, gather data, and take intelligent actions locally,
 even when internet connectivity is not available. Control services let you control, m
anage, and secure large and diverse device fleets. Data services help you extract valu
e from IoT data.

Industrial services: AWS IoT customers are building industrial IoT applications for pr
edictive quality and maintenance and to remotely monitor operations.

Connected home services: AWS IoT customers are building connected home applications fo
r home automation, home security and monitoring, and home networking.

Commercial: AWS IoT customers are building commercial applications for traffic monitor
ing, public safety, and health monitoring.
What misconceptions might you have about the IoT?

Some might mistake the IoT for individualized programming or individual device sensing
or programming. The IoT refers to the connected interaction between devices. The IoT i
s not simply smart functionality of a device. The concept includes the ability of the
 device to communicate with other nodes in the cloud and use and connect data across t
he network.
```

Module 13:

> **Infrastructure as Code:** The process of provisioning and managing cloud resources by writing a template file that is human readable and machine consumable. For Amazon Web Services (AWS) cloud development, the built-in choice for IaC is AWS CloudFormation.

> **Stack:** A collection of Amazon Web Services (AWS) resources that you can manage as a single unit. In other words, you can create, update, or delete a collection of resources by creating, updating, or deleting stacks.

```
CloudFormation is a service that lets users easily build and manage a stack of AWS res
ources, such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Se
rvice (Amazon S3). It is a form of IaC. With CloudFormation, users can accomplish this
process by using prebuilt AWS coding and scripts, thereby making the process very user
friendly. Students should conceptualize CloudFormation as the virtual way that cloud c
omputing networks have a structure, just as previous generations of computer systems h
ad a literal architecture of hardware, cables, and physical tools.

CloudFormation simplifies provisioning and management on AWS. You can create templates
for the service or application architectures you want, and have CloudFormation use tho
se templates for quick and reliable provisioning of the services or applications. Thes
e collections of services are known as stacks. Users can also easily update or replica
te the stacks as needed.

CloudFormation provides a common language for you to model and provision AWS and third
-party application resources in your cloud environment. With CloudFormation, you can u
se programming languages or a simple text file to model and provision, in an automated
and secure manner, the resources needed for your applications across all Regions and a
ccounts. This gives you a single place where information about the state of your appli
cation is kept (source of truth) for your AWS and third-party resources.

The CloudFormation templates are large preset blocks of code. The code is usually writ
ten in the JavaScript Object Notation (JSON) coding language or in YAML (standing for
 the whimsical phrase YAML Ain't Markup Language. The AWS console permits users to wor
k with the code or use a simplified form of the templates that are visual icons. Eithe
r service is vastly easier than having a DevOps team that actually writes the code.

Misconceptions

You might think that a cloud-based network literally exists in the sky, such as in sat
ellites or some unknown amorphous idea of a cloud. You should understand that cloud co
mputing networks are connections to applications on AWS servers and not on local devic
es.
```

Module 14:

> **Big Data:** Big data can be described in terms of data management challenges that—because of vastly increased

> volume, velocity, and variety of data—cannot be solved with traditional databases. While there are plenty of definitions for big data, most of them include the concept of what is commonly known as the three Vs of big data: volume, variety, and velocity.

> **Data Processing:** The step in which data is transformed from its raw state into a consumable format—usually by means of cleaning, sorting, aggregating, joining, and even performing more advanced functions and algorithms. The resulting datasets are then stored for further processing or made available for consumption through business intelligence and data visualization tools.

> **Apache Hadoop:** A technology for managing large volumes of data in a distributed environment. Hadoop was developed by Apache and is now available to run on clusters of virtual servers in an AWS Cloud.

```
You know what data is, but what about big data? You have probably heard of it, but do
 you know how it differs from more traditional data? Big data is more than just a larg
e group of data. Big data has certain characteristics that require the use of special
 tools to process and analyze it. Big data is differentiated by its enormous size (vol
ume), wide range of data types (variety), and the lightning-fast speed at which it is
 collected and processed (velocity). These three Vs are the basis of what is considere
d big data. Veracity (the accuracy or reliability of the data) and value (the usefulne
ss of the data) are commonly used to describe big data as well.

Volume refers to the amount of data.
Variety refers to the range of data types within the dataset.
Velocity refers to the speed at which the data is collected and processed.
Veracity refers to the reliability and accuracy of the data.
Value refers to the usefulness of the data.
When these characteristics are in play, the data is big data, and it is stored and han
dled differently. Big data is being used every day to provide tremendous benefits in a
lmost every industry.
```

```
Collecting raw data—transactions, logs, mobile devices, images, and more—is the first
 challenge that many organizations face when dealing with big data. A good big data pl
atform makes this step easier, letting developers ingest a wide variety of data—from s
tructured to unstructured—at any speed, from batch to real-time to batch.
```

Of course, as with any technology, there are risks. With big data, these can be magnified because of how the datasets are processed and by the vast reach of the technologies. Some of the primary risks with big data relate to privacy and bias. Of course, whenever data is collected and stored, there is a privacy risk. However, with traditional data, the details that are sensitive are relatively well known and can be protected, or at least monitored. With big data, this is not always the case. Often big data is aggregated or combined from multiple sources, and steps are taken to remove identifiers and make the data anonymous. That should work, right? It is not that simple. While the information about you that is stored in a big dataset probably does not have your name or address, it has other clues to your identity and can be traced back to you with a little work and the use of AI. Security and regulations often lag behind advancements, and this is true with big data technologies as well.

Another primary risk with big data is that we might unknowingly rely on biased results. What if the biased results kept you from being able to get a loan to go to college or get a driver's license or board a plane? Social media data is being used in some industries to assign scores to people based on their connections, likes, posts, and other data. If the analysis model wrongly assumes that a certain profile of a person would not be as good a credit risk or would be more likely to be violent, then the results of any future analysis will be inaccurate. But we might not realize the error or be able to prove it, because the algorithms are often hidden and we cannot see the logic used. So these models must be designed very thoughtfully.

Regardless of the risks, big data is everywhere and it is a hot topic. You can learn more about the products and services provided by AWS for big data at these links:

AWS Certified Big Data – Specialty (https://aws.amazon.com/certification/certified-big-data-specialty/)
AWS Customer Success Stories (in the filters along the left of the screen, choose Tech Use Case > Analytics) (https://aws.amazon.com/solutions/case-studies/)
With all this in mind, turn your attention to how modern computer technology can be applied to satisfy the requirements of the three Vs.

Approach 1: One approach would be to have a supercomputer with enormous amounts of storage attached to an extremely fast network.
Approach 2: A second approach would be to have a lot of smaller computers, each with a modest amount of storage, connected by networking.
Approach 3: A third approach would be to have the second approach running on a cloud using web services, such as AWS, that are provided by the cloud.
So much for the hardware. We will discuss the pros, cons, and costs of each of these approaches later in this module.

Now we need to think of the software required to run in whichever hardware approach we have chosen. If Approach 1 is chosen, then standard relational or nonrelational database software will probably work. However, remember that Approach 1 is very expensive in terms of both dollar cost and the cost of maintaining the hardware and software.

Approach 2 will require software that can handle the distribution of tasks across a collection of smaller physical machines with their associated storage. This software will need to split up the data across the machines and then do parallel processing to manage the data in a coordinated way that produces reliable results. Hadoop is software that can do that.

Approach 3 will require the same software, but the burden of provisioning the machines and maintaining them with upgrades to operating systems and database software will now be the responsibility of the web services provider, such as AWS. Furthermore, this approach scales out and in, which means that virtual servers are added and removed as the

```
size of the data requires, saving the cost of having servers provisioned when they are
not needed.

This module will focus on Approach 3, although we will consider how big data is handle
d in a traditional environment and in the cloud.
```

Module 15:

> **Dashboard:** A read-only snapshot of an analysis that you can share with other users for reporting purposes. The data used for the analysis isn't captured as part of the dashboard. When you view the dashboard, it reflects the current data in the datasets used by the analysis.

Big Data Processing Cycle:

```
Model of the common flow of big data—from the collection of raw data to consumption of
actionable information. This generally consists of four phases:

Collect: Collecting the raw data—such as transactions, logs, and mobile devices—is the
first challenge that many organizations face when dealing with big data. A good big da
ta solution makes this step easier, permitting developers to ingest a wide variety of
 data—from structured to unstructured—at any speed from real time to batch.
Store: Any big data solution needs a secure, scalable, and durable repository to store
data before or after processing tasks. Depending on your specific requirements, you mi
ght also need temporary stores for data in transit.

Process and analyze: This step is where data is transformed from its raw state into a
 consumable format—usually by means of sorting, aggregating, joining, and performing m
ore advanced functions and algorithms. The resulting datasets are then stored for furt
her processing or made available for consumption with business intelligence and data v
isualization tools.

Consume and visualize: Big data is all about getting high-value, actionable insights f
rom your data assets. Ideally, data is made available to stakeholders through self-ser
vice business intelligence and agile data visualization tools. These allow for fast an
d easy exploration of datasets. Depending on the type of analytics, users might also c
onsume the resulting data in the form of statistical predictions, in the case of predi
ctive analytics, or recommended actions, in the case of prescriptive analytics.

The big data processing cycle includes four main phases: collect, store, process and a
nalyze, and consume and visualize. Data may move back and forth between the store and
 process and analyze phases during the cycle.
```

As a quick review, here’s a reminder of what big data means. It is the big part that is important because the amount of data in the world today and the rate at which it is growing is staggering. We are currently at 1 zettabyte of data and growing at an annual compound growth rate of 42 percent.

What is a zettabyte? In terms of exponents, it is 1021. Said another way, it is a trillion gigabytes.

However, the growth rate is truly staggering. At 42 percent annually, it means that by 2025, there will be 175 zettabytes of storage worldwide.

So what does all this mean? For one thing, solving the big data problem by getting bigger and bigger storage devices will soon not be possible. Therefore, using a distributed approach, where data is spread across multiple devices, will be necessary, and this is what big data is all about. It is not just spreading out the data, but also extracting information from the data, which will require special software and networking solutions.

Big data projects have lifecycles, just like any other project. You can see a depiction of the big data processing cycle here.

Models like this can help us understand the steps in a process, the types of inputs and outputs expected for each phase, and how to communicate about the project with team members. They provide general guidelines for how the project should progress. This particular model was provided by AWS.

The phases of the cycle seem simple: collect, store, process, and consume. However, the activities in each phase that help to ensure the quality and usefulness of the data can be complex.

A wide variety of raw data can be collected. The solutions you choose for the collection phase must be able to handle the types of data you are using, whether it is structured, semi-structured, unstructured, or a combination of the three. The data is likely to be received and processed in real time or by streaming, which must be supported by the technologies chosen for the project. After the data is collected, it must be stored in a structure that will support the types of data and the processing that needs to occur. Organizations have several options for big data storage, including nonrelational databases, data warehouses, and data lakes.

Big data storage: Architecture that collects and manages large datasets and facilitate

```
s real-time data analytics.

Nonrelational databases: A database that does not use rows and columns found in tradit
ional database systems. Instead, they use a storage model that is optimized for the ty
pe of data being stored.

Data warehouses: Central stores of integrated data from one or more sources.

Data lake: Repository of data stored in its raw format.

The quality of big data can be low, depending on the types and sources, so it must be
 cleansed and transformed often to improve the accuracy of the data and resulting anal
ysis. When the data has been prepared and stored in an accessible data store, it is av
ailable for analysis.

Organizations use key performance indicators, or specific measure of success, to drive
what data they collect and what questions they ask of their data, after it is availabl
e for analysis. Some businesses will track indicators related to safety and manufactur
ing defects, whereas others will track data points related to website activity and the
effectiveness of specific advertising. Analytics can help organizations answer the fol
lowing questions about their business:

What happened?
Why did it happen?
What will happen?
What should I do?

The results of the analysis should be presented in a way that conveys the information
 and findings quickly and clearly. Dashboards and visualizations are often used for th
at purpose.

Big data solutions might involve the use of AI or ML. Both technologies rely on large
 amounts of high-quality data to learn and generate predictions.
```

Module 16:

> **Hardening:** Hardening means to reduce vulnerabilities in a technology. The goal is to reduce security risk.

AWS Blockchain products:

```
AWS provides three blockchain products to make cloud-based blockchain solutions.

Amazon Quantum Ledger Database (Amazon QLDB): A fully managed ledger database that pro
vides a transparent, immutable, and cryptographically verifiable transaction log owned
by a central trusted authority. Amazon QLDB tracks every application data change and m
aintains a complete and verifiable history of changes over time. For more information
 about Amazon QLDB, see these resources:
Amazon Quantum Ledger Database (https://aws.amazon.com/qldb/)
“AWS QLDB Overview: Animated Explainer Video” (https://www.youtube.com/embed/jcZ_rsLJr
qk)
```

Amazon Managed Blockchain: A fully managed service that lets you set up and manage a scalable blockchain network with just a few clicks using the popular open-source frameworks Hyperledger Fabric and Ethereum. Managed Blockchain eliminates the overhead required to create the network and automatically scales to meet the demands of thousands of applications running millions of transactions. Managed Blockchain makes it possible to build applications where multiple parties can run transactions without the need for a trusted, central authority. Today, building a scalable blockchain network with existing technologies is complex to set up and hard to manage.
Review Amazon Managed Blockchain (https://aws.amazon.com/managed-blockchain/) to learn more.
For a thorough overview, see “Deep Dive on Amazon Managed Blockchain” (https://www.youtube.com/watch?v=5j5rhCXw8fE).
AWS Blockchain Templates: Lets you deploy the blockchain framework you choose as containers on an Amazon Elastic Container Service (Amazon ECS) cluster, or directly on an Amazon Elastic Compute Cloud (Amazon EC2) instance running Docker. Your blockchain network is created in your own Amazon Virtual Private Cloud (Amazon VPC), letting you use your VPC subnets and network access control lists. You can assign granular permissions using AWS Identity and Access Management (IAM) to restrict which resources an Amazon ECS cluster or EC2 instance can access.