**Definition and Essential Characteristics of Cloud Computing**

Cloud computing, also referred to as “the cloud,” is the delivery of on-demand computing

resources. Everything from applications to data centers, over the internet on a pay-for-use

basis.

To get a common understanding of cloud computing, let’s start with the US National Institute

of Standards and Technology (NIST’s) definition of cloud computing.

NIST defines cloud computing as a model for enabling convenient, on-demand network access

to a shared pool of configurable computing resources, that can be rapidly provisioned

and released with minimal management effort or service provider interaction.

Examples of computing resources include networks, servers, storage, applications, and services.

This cloud model is composed of five essential characteristics, three deployment models,

and three service models.

Let’s start with understanding the five essential characteristics of the cloud, which

include on-demand self-service, broad network access, resource pooling, rapid elasticity,

and measured service.

On-demand Self-service, the 1st characteristic, means that you get access to cloud resources

such as the processing power, storage, and network you need, using a simple interface,

without requiring human interaction with each service provider.

The 2nd characteristic, Broad Network Access, means that cloud computing resources can be

accessed via the network through standard mechanisms and platforms such as mobile phones,

tablets, laptops, and workstations.

The 3rd characteristic, Resource Pooling, is what gives cloud providers economies of

scale, which they pass on to their customers, making cloud cost-efficient.

Using a multi-tenant model, computing resources are pooled to serve multiple consumers; cloud

resources are dynamically assigned and reassigned, according to demand, without customers needing

to concern themselves with the physical location of these resources.

Rapid Elasticity, the 4th characteristic, implies that you can access more resources

when you need them, and scale back when you don’t, because resources are elastically

provisioned and released.

And the 5th characteristic, Measured Service, means that you only pay for what you use or

reserve as you go. if you’re not using resources, you’re not paying.

Resource usage is monitored, measured, and reported transparently based on utilization.

As we see, cloud computing is really about utilizing technology “as a service”, leveraging

remote systems on-demand over the open internet, scaling up and scaling back, and paying for

what you use.

It is a revolution in that it has changed the way the world consumes compute services,

by making them more cost-efficient while also making organizations more agile in responding

to changes in their markets.

As I mentioned earlier in this talk, the cloud model is composed of five essential characteristics,

three deployment models, and three service models.

We just went over the five essential characteristics.

While we will go into greater depth of the deployment and the service models in the later

videos, let me leave you with a brief overview of these models.

There are three types of cloud deployment models—Public, Private, and Hybrid.

Public cloud is when you leverage cloud services over the open internet on hardware owned by

the cloud provider, but its usage is shared by other companies.

Private cloud means that the cloud infrastructure is provisioned for exclusive use by a single

organization.

It could run on-premises or it could be owned, managed, and operated by a service provider.

And when you use a mix of both public and private clouds, working together seamlessly,

that is classified as the Hybrid model.

Now, let’s look at the three service models that are based on the three layers in a computing

stack - Infrastructure, Platform, and Applications.

These cloud computing models are aptly referred to as Infrastructure as a Service (Iaas),

Platform as a Service (PaaS), and Software as a Service (SaaS).

In an Infrastructure as a Service model, you get access to infrastructure and physical

computing resources such as servers, networking, storage, and data center space - without the

need to manage or operate them.

In a Platform as a Service model, you get access to the platform, that is the hardware

and software tools, usually those needed to develop and deploy applications to users over

the Internet.

Software as a Service is a software licensing and delivery model in which software and applications

are centrally hosted and licensed on a subscription basis, and sometimes also referred to as "on-demand

software."

**History and Evolution of Cloud Computing**

Cloud computing is an evolution of technology over time.

The concept of cloud computing dates to the 1950s when large-scale mainframes with high-volume

processing power became available.

In order to make efficient use of the computing power of mainframes, the practice of time

sharing, or resource pooling, evolved.

Using dumb terminals, whose sole purpose was to facilitate access to the mainframes, multiple

users were able to access the same data storage layer and CPU power from any terminal.

In the 1970s, with the release of an operating system called Virtual Machine or VM, it became

possible for mainframes to have multiple virtual systems, or virtual machines, on a single

physical node.

The virtual machine operating system evolved the 1950s application of shared access of

a mainframe by allowing multiple distinct compute environments to exist on the same

physical hardware.

Each virtual machine hosted guest operating systems that behaved as though they had their

own memory, CPU, and hard drives, even though these were shared resources.

Virtualization thus became a technology driver and a huge catalyst for some of the biggest

evolutions in communications and computing.

Even 20 years ago, physical hardware was quite expensive.

With the internet becoming more accessible, and the need to make hardware costs more viable,

servers were virtualized into shared hosting environments, virtual private servers, and

virtual dedicated servers, using the same types of functionality provided by the virtual

machine operating system.

So, for example, if a company needed ‘x’ number of physical systems to run their applications,

they could take one physical node and split it into multiple virtual systems.

This was enabled by hypervisors.

A hypervisor is a small software layer that enables multiple operating systems to run

alongside each other, sharing the same physical computing resources.

A hypervisor also separates the Virtual Machines logically, assigning each its own slice of

the underlying computing power, memory, and storage, preventing the virtual machines from

interfering with each other.

If, for example, one operating system suffers a crash or a security compromise,

the others keep working.

As technologies and hypervisors improved and were able to share and deliver resources reliably,

some companies decided to make the cloud’s benefits accessible to users who didn’t

have an abundance of physical servers to create their own cloud computing infrastructure.

Since the servers were already online, the process of spinning up a new instance was

instantaneous.

Users could now order cloud resources they needed from a larger pool of available resources

and they could pay for them on a per-use basis, also known as Pay-As-You-Go.

This pay-as-you-go or utility computing model, became one of the key drivers behind cloud

computing taking off.

The pay-per-use model allowed companies and even individual developers to pay for the

computing resources as and when they used them, just like units of electricity.

This allowed them to switch to a more cash-flow friendly OpEx model from a CapEx model.

This model appealed to all sizes of companies, those who had little or no hardware, and even

those that had lots of hardware, because now, instead of making huge capital expenditures

in hardware, they could pay for compute resources as and when needed.

It also allowed them to scale their workloads during usage peaks, and scale down when usage

subsided.

And this gave rise to modern-day cloud computing.

The impact of the evolution of the cloud has been immense.

**Key Considerations for Cloud Computing**

Every organization’s transformation journey is unique, and therefore every organization’s

cloud adoption strategy is also unique to them.

Agility, flexibility, and competitiveness are key drivers for moving to the cloud, provided

it is done without creating business disruption or issues related to security, compliance

and performance.

Let’s look at some key considerations that organizations can use as a guide while working

through their cloud strategy.

The first consideration is infrastructure and workloads.

The cost of building and operating data centers can become astronomical.

On the other hand, low initial costs and pay-as-you-go attributes of cloud computing can add up to

significant cost savings.

Also, a point to consider is that not all workloads may be ready for the cloud, as is.

The second consideration is around SaaS and development platforms.

Organizations need to consider if paying for application access is a more viable option

than purchasing off-the-shelf software and subsequently investing in upgrades.

Organizations also need to consider speed and productivity, what it means for them

to get a new application up and running in ‘x’ hours on the cloud versus a couple

of weeks, even months on traditional platforms.

And the person-hour cost efficiencies they gain from using cloud dashboards, real-time

statistics, and active analytics.

Lastly, organizations need to consider the impact of making a wrong decision, their

risk exposure.

Is it riskier, for example, for them to invest in the hardware and software or rent by the

hour?

Is it safer for them to work on a 12-month plan to build, write, test, and release the

code if they’re uncertain about adoption?

And is it better for them to “try” something new paying-as-you-go rather than making long-term

decisions based on little or no trial or adoption?

Let’s look at some of the benefits of cloud adoption, categorized broadly into Flexibility,

Efficiency, and Strategic Value.

Cloud gives us flexibility.

Users can scale back or scale up services to fit their needs, customize applications,

and access cloud services from anywhere with an internet connection.

Cloud infrastructure scales on demand to support fluctuating workloads.

Organizations can determine their level of control with as-a-service options.

Users can select from a menu of pre-built tools and features to build a solution that

fits their specific needs.

And Virtual Private Clouds, encryption, and API keys help keep data secure.

Cloud also brings great efficiency.

Enterprise users can get applications to market quickly without worrying about underlying

infrastructure costs or its maintenance.

Cloud-based applications and data are accessible from virtually any internet-connected device.

Hardware failures do not result in data loss because of networked backups.

Cloud computing uses remote resources, saving organizations the cost of servers and other

equipment, and paying on use-basis.

Cloud services give enterprises a competitive advantage by providing the most innovative

technologies available while managing the underlying infrastructure, thus enabling organizations

to focus on their priorities.

While cloud brings great opportunity, it also introduces challenges for business leaders

and IT departments.

Some of these perceived risks include: Data security, associated with loss or unavailability

of data causing business disruption; Governance and sovereignty issues;

Legal, regulatory, and compliance issues; Lack of standardization in how the constantly

evolving technologies integrate and interoperate; Choosing the right deployment and service

models to serve specific needs; Partnering with the right cloud service providers;

Concerns related to business continuity and disaster recovery.

Organizations can no longer think of cloud adoption as something that is to be looked

at in the future.

With the right cloud adoption strategies, technologies, services, and service providers,

these risks can be mitigated.

**Key Cloud Service Providers and Their Services**

Let’s look at some numbers and predictions that help us see the scale, cloud is predicted

to achieve in the coming years.

Gartner predicts: The worldwide public cloud service market

to grow from $182.4 B in 2018 to $331.2 B in 2022, attaining

a compound annual growth rate (CAGR) of 12.6%.

Spending on Infrastructure-as-a-Service to increase from $30.5 B in 2018

to $76.6 B in 2022, growing 27.5% in a year.

Platform-as-a-Service spending to grow from $15.6 B in 2018 to $31.8B

2022, growing 21.8% in a year.

With software investments shifting from cloud-first to cloud-only, spending on Sofware-as-a-Service

is expected to grow from $80 B in 2018 to $143.7 B in 2022.

What is clear is that cloud is accelerating faster than predicted, adoption is high, and

revenues are soaring.

The question for businesses today is no longer “if” they need to adopt the cloud, rather

“what” their cloud adoption strategy should be to best serve their businesses and customers.

Keeping up with this technological wave, and driving it forward, are the Cloud Service

Providers with a wide range of services.

We will now talk about some of the major Cloud Service Providers in alphabetical order.

Alibaba Cloud, also known as Aliyun, while relatively new, is the largest Chinese cloud

computing service provider.

Aliyun provides a comprehensive suite of global cloud computing services to power not just

their customers’ online businesses but also the Alibaba Group’s own e-commerce ecosystem.

It offers a host of products and services such as compute, network, storage, security,

monitoring and managing, communication, analytics, IoT, application development, data migration,

web hosting, and more.

One of the first to enter the cloud computing space, Amazon Web Services, or AWS Cloud,

offers an extensive range of Infrastructure and Platform services to individuals, companies,

and governments on a metered pay-as-you-go basis.

The Amazon Cloud provides a wide range of products, services, and solutions ranging

from Compute, DevOps, Data, Analytics, IoT, Machine Learning, Networking, Content Delivery,

Robotics, Serverless Computing, and much more.

Google Cloud Platform, or GCP, is a suite of cloud computing services, providing Infrastructure,

Platform, and Serverless Computing environments.

Google also uses GCP internally for their end-user products such as Google Search and

YouTube.

Google Cloud includes G Suite with products for communication, productivity, collaboration,

storage, and more.

The Google App Engine is a platform for developing and hosting web applications in Google-managed

data centers, automatically allocating and de-allocating resources to handle demand.

IBM cloud is a full stack cloud platform that spans public, private, and hybrid environments

with products and services covering compute, network, storage, management, security, DevOps,

and databases.

Some of their prominent offerings include their Bare Metal Servers, VMWare, Cloud Paks

for Application Modernization, Virtual Private Cloud, and the suite of emerging technologies

such as AI, IoT, Blockchain, Data and Analytics.

With the acquisition of Red Hat, IBM is also positioning itself as the leading hybrid cloud

provider of our times.

Microsoft Azure is a flexible cloud platform for building, testing, deploying, and managing

applications and services through Microsoft managed data centers.

With its data centers spread out in many regions, Azure provides a global reach with a local

presence.

It provides Software, Platform, and Infrastructure services supporting Microsoft-specific and

third-party languages, tools, and frameworks.

Oracle Cloud is primarily known for Software as a Service and Database as a Service (also

known as the Oracle Data Cloud).

Oracle’s SaaS offering includes wide-ranging applications such as ERP, SCM, HCM, Marketing,

Sales, and CX running in the cloud.

And the Oracle Data Cloud provides one of the largest cloud-based data management platforms

helping customers personalize their online, offline, and mobile marketing campaigns, for

targeted audiences.

Oracle Cloud also provides some cloud Infrastructure and Platform services.

Salesforce specializes in their Software as a Service offering that focuses on customer

relationship management, supporting businesses to better connect with their customers, partners,

and potential customers.

Salesforce offers multiple cloud services such as Sales Cloud, Service Cloud, and Marketing

Cloud, helping customers track analytics in real-time, customer success and support, customer

complaints, even listening in to customers across social platforms to automatically route

them to appropriate agents for resolution.

SAP is known for Enterprise software and applications such as ERP, CRM, HR, and Finance, running

in the cloud.

There is also an SAP Cloud Platform for building and extending business applications with rapid

innovation cycles in a secure cloud computing environment managed by SAP.

**Cloud Adoption - No longer a choice**

Cloud adoption is no longer a thing of the future.

From a single individual to a global multi-billion-dollar enterprise, anybody can access the computing

capacity they need on the cloud.

The lag time from decision to value is no longer a journey of years with high upfront

capital. Cloud makes it possible for businesses to experiment, fail, and learn much faster

than ever before, with low risk exposure.

Businesses today have greater freedom to change course than to live with the consequences

of expensive decisions taken in the past.

According to an IBM Institute for Business Value study, more than three-quarters of enterprises

today are using cloud computing to expand into new industries.

74% have adopted cloud to improve customer experience, and 71% use cloud to create enhanced

products and services while simultaneously downsizing legacy systems and reducing costs.

To remain competitive, businesses need to be able to respond quickly to marketplace

changes, use analytics to understand customer experience, and apply that understanding to

adapt their products and services based on what they learn.

Product lifecycles have shortened, and barriers to entry have become lower.

Cognitively-enabled workflows,

Applied exponential technologies such as AI, Automation, IoT, and Blockchain,

applications that span new and legacy solutions, and

open hybrid and secure Multicloud infrastructures, are today's

enablers for growth, agility and innovation. The power, scalability, flexibility, and pay-as-you-go economics of cloud

has made it the underpinning foundation for digital transformation.

The International Data Corporation, IDC, predicts that by 2025, the total amount of digital

data created worldwide will rise to 163 zettabytes (where one zettabyte is equivalent to a trillion

gigabytes).

And 30% of this data will be real-time information.

Considering the unprecedented amounts of data being produced daily, and the ability to make

data-driven decisions crucial to any business, cloud computing becomes essential for businesses

to succeed, sustain, and compete in today’s markets.

A cloud strategy, more than just an IT strategy, is the core component of any business strategy

today.

Businesses that haven’t already, or are not currently, integrating cloud into their

business strategy, run the risk of lacking the speed, agility, innovation, and decision-making

capacities needed to be competitive, as also their ability to respond to digital disruption.

**Cloud Adoption - Some case studies**

In this video, we will look at how some of the leading businesses have transformed the

way they work to provide better customer service, remove barriers to innovation, achieve enterprise

scale, and accelerate growth, using cloud technologies.

Although the case studies we’ll look at are curated from IBM Cloud, similar stories

with dramatic impact to business can be found across the spectrum of companies utilizing

other Cloud Service providers as well.

BETTER CUSTOMER SERVICE: In the highly competitive airline industry,

customer experience is a major point of differentiation, and digital channels are increasingly important.

To become more responsive to customer needs, American Airlines needed a new technology

platform and a new approach to development that would help it deliver digital self-service

tools and customer value more rapidly across its enterprise.

The airline recognized the opportunity to remove the constraints of their existing customer-facing

applications, based on monolithic code into cloud-native based microservices architecture

on the cloud.

The results: Faster development and release of new apps.

Improved operational reliability, productivity, and end customer response times.

Cost savings by avoiding existing upgrade costs via migration to the IBM Cloud.

REMOVING BARRIERS TO INNOVATION: As a lean organization with a self-imposed

limit on headcount, UBank excels at finding innovative ways to meet demands.

Continually challenged to find more efficient ways to operate, UBank’s IT team explored

a Platform as a Service (PaaS) cloud development model.

Their need was to give more control to their developers, reduce the need for additional

resources, faster speed to market, and removing barriers in going from an idea to production.

UBank launched new initiatives in an IBM Cloud Platform environment, including a virtual

assistant that incorporates IBM Watson technology to support the bank’s online home loan application.

The results: Faster time to market made possible through

the Cloud Platform framework, that streamlines development and empowers product teams.

Foster greater innovation with cloud-based development resources that are quick, easy,

and cost-effective to deploy

more efficient operations.

DEMAND FOR ENTERPRISE SCALE: Since its inception in 2008, Bitly has journeyed

from a startup that offered intelligent link-shortening technology adopted by users to compress lengthy

URLs for social media posts, to an enterprise product.

Seeking an agile, cost-effective IT infrastructure to support this transition, Bitly started

planning for cloud migration.

Their need was to have a cloud-based model with pay-as-you-go pricing, the ability to

scale up and down, a more global presence, and the ability to geodistribute into more

POPs.

And they wanted it to be low-risk.

Bitly migrated to an IBM Cloud environment, establishing a scalable hosting platform for

low-latency delivery to enterprise customers around the world.

The results: 25 billion data-infused links migrated from

one hosting site to Cloud infrastructure with data center locations worldwide.

1 billion user interaction data set stored and managed in a flexible, cost-effective

Cloud Object Storage environment.

Transformed IT operations to scale for growth, control costs and focus valuable resources

on new product development.

ACCELERATING GROWTH: Financial traders demand extreme speed and

availability from trading systems.

Profitability depends on split-second decisions.

As a leading online broker in forex, commodities, equities, cryptocurrencies, indices, and other

financial instruments, ActivTrades enables investors to buy and sell on numerous financial

markets.

Investors need reliable access to accurate market information, combined with the ability

to move rapidly to execute trades.

As its client base grew, Active Trades wanted to cut latency, accelerate execution, and

streamline the delivery of new functions.

ActivTrades migrated three major trading systems from on-premises infrastructure to IBM Cloud

for VMware solutions, backed by data storage, networking, and security offerings on the

IBM Cloud.

The results: Up to 3X performance boost, helping clients

seize fleeting opportunities for profit.

Security-rich cloud platform with ultra-high availability protects client investments.

Hours, not days to fire up new resources, for faster response to emerging requirements.

**Internet of Things in the Cloud**

In this new era, technologies such as Internet of Things, Big Data, artificial

intelligence, and blockchain are disrupting existing business models and

industries while creating unprecedented opportunities for businesses to

differentiate themselves and create value for their clients. The power, scale,

dynamic nature, and economics of the cloud resources make cloud computing a

key enabler for adoption and evolution of these emerging technologies. In this

video, we will look at how the Internet of Things, or IoT, powered by cloud, is

making a big difference for businesses today. The Internet of Things, or IoT, is a

giant network of connected things and people that have changed much of how we

live our daily lives - from the way we drive, to how we make purchases,

monitoring our personal health, and even how we get energy for our homes. Smart

devices and sensors are continuously tracking and collecting data. For example,

a smart building could have thousands of sensors measuring all kinds of data

related to thermal, optical, structural, and environmental stimuli. An

unprecedented amount of data is being generated, putting a tremendous strain on

the Internet. That is where the cloud comes in, by

connecting the IoT device user to the cloud - be it for device registration,

device identity, storing data, or accessing enterprise data. Data collected

through IoT devices is stored and processed on the cloud since IoT devices

can be in a state of motion, the cloud serves as a collection point in closest

proximity, minimizing the latency in reporting up the data points and

providing a response back to the IoT application.

So, from IOT platforms running entirely on the cloud to the interfaces used by

customers to interact these devices, to the backend analytics

platforms - cloud computing supports and enables IoT. Cloud service providers also

offer specialized IoT services designed to help speed up the development of IoT

solutions. Let's look at a case study that demonstrates the use of the IoT on

the cloud to combat the poaching of endangered rhinos at Welgevonden by

making poaching predictable.

The rhinos have become one of the the key species that is becoming endangered due

to poaching throughout Africa. But now especially in South Africa.

Up until now, poachers have been increasing in numbers, and they become

more militarized with weapons. And so of course we've had to do the same. This is

not sustainable. The only way to do this better, is to bring in technology and

things that they do not have. This endangered species is getting help from

some unexpected friends, the zebra and antelope. They're wearing IoT sensors

connected to the IBM cloud. When poachers enter the area, the animals run for it,

which alerts Rangers who can track their emotions and help stop them before any

harm is done. It's a smart way to help increase the Rhino population and turn

the poachers into the endangered species.

**Artificial Intelligence on the Cloud**

Making sense of the endless streams of data is where Artificial Intelligence, or AI, comes

in.

Many of the applications where we apply AI today simply wouldn't have been possible without

the scalable, on-demand computing offered by the cloud.

There is a three-way relationship between AI, IoT, and Cloud.

Just as AI consumes the data produced by IoT devices, the IoT devices’ behavior can be

dictated based on responses from AI.

For example, Smart Assistants, a common type of IoT device, continues to learn about the

user’s preferences as usage grows, such as the songs they like, their home temperature

settings, preferred meal times, and over time they anticipate their actions based on the

user’s past history.

So, what we see is a symbiotic relationship between IoT, AI, and Cloud.

IoT delivers the data, AI powers the insights, and both these emerging technologies leverage

cloud’s scalability and processing power to provide value to individuals and businesses

alike.

Let’s look at how the United States Tennis Association, USTA, is using AI on the Cloud

to deliver unique digital experiences to millions of fans around the world.

For two weeks at the end of every summer, tennis fans around the world turn their eyes

to New York city, and the US Open.

Hundreds of thousands onsite, and millions more online.

But where you see tennis, IBM sees data;

the scores and statistics, the sights, the sounds.

IBM integrates and analyzes the data flowing from the court.

And delivers unique digital experiences to more than ten million tennis fans around the

world.

And we do it all in the IBM Cloud.

The IBM Cloud is the digital foundation of the US Open.

It scales rapidly to meet a five thousand percent increase in web traffic

and it delivers a consistant experience to our fans all around the globe.

And with Watson on the IBM Cloud, we can engage fans in unique ways, year after year.

Slam Tracker analyzes more than twenty-six million historical data points.

It gives fans deep insight into featured matches, and it can see the momentum of a match shifting

in real time.

AI Highlights uses Watson to process thousands of hours of US Open video.

It can hear the cheers of the crowd.

It can see a player celebrating

and it knows what makes a great tennis highlight.

And this year we're putting the power of AI Highlights into the hands of US players and

coaches.

Watson is analyzing match video,

so coaches can quickly find the footage they need to guide the development of their players.

And if you need to know where to park, find a good burger, or grab the latest US Open

gear, you can find the answers with the Guest Information feature in the US Open app

and mobile web, using Watson.

We work with IBM because they keep us on the cutting edge of the fan experience.

They help us to adopt the latest technology, like Cloud and AI.

And they bring data to life in a way that's accessible and engaging for our fans.

**Blockchain and Analytics in the Cloud**

Blockchain is a secure, distributed, open technology that can help speed up

processes, lower costs, and build transparency and traceability in

transactional applications. It is an immutable Network allowing members to

view only those transactions that are relevant to them. The more open, diverse,

and distributed the network, the stronger the trust and transparency in the data

and transactions. 85% of businesses today rely on multiple clouds to meet their IT

needs, with more than 70% using more than three. These businesses need to be able

to move applications and data across multiple clouds easily and securely,

leading to the emerging demand to build and manage business applications such as

blockchain for the multi cloud environment. Blockchain and AI, much like

IoT and AI, powered by the cloud, also have a three-way relationship. Where

blockchain technology provides the trusted, decentralized source of truth, AI

powers the analytics and decision-making from the data collected, and cloud

provides globally distributed, scalable, and cost-efficient computing resources

to support both the unprecedented amounts of data being collected and the

processing power required to draw insights from this data. Blockchain

serves to make AI more understandable by recording the data and variables that go

into a decision made in an AI algorithm, leading to greater trust and

transparency in the conclusions and decisions made by these algorithms. Let's

look at how blockchain on the cloud is helping farmers reduce waste at times of

recall by building traceability and transparency in the food supply chain.

For farmers here, this is our life's work. 60% of the nation's lettuce is

grown right here in Salinas. When it comes down to how plants are looked at, I

really relate back to how humans are raised and nurtured. I want to make sure

that everything is safe before it leaves the ranch. But when a recall happens,

perfectly good food goes to waste. You have to take all product off your shelf,

no matter what age, no matter where it came from. And it takes resources to grow

those things. Now we're actually eating into our future food supply. But

we've gotta way around that. With Blockchain technology on the IBM cloud

we're able to track our product within seconds. Giving the consumers instant

access where the product came from, in case there's any recall so we don't

have to take all the food off the shelves.

Having that instant access allows you to reduce the waste. There's a lot of

starving people in this world. I want to be a generation that

fixes that.

Analytics technologies on the cloud leverage the flexibility, scalability, and

computing resources available on the cloud. From tracking trends on social

media to predict future events, to analyzing data to build machine learning

models that can be deployed in cognitive applications, cloud provides the

integrated environment that is required to leverage data for continuous

improvement and accelerated business growth. Let's look at how KONE has

invested in cloud and IoT technologies to power a data analytics and predictive

maintenance solution for city infrastructure used by more than 1

billion people daily. At KONE we manufacture elevators, escalators, auto

walks, and doors All of these devices are streams of data that we are collecting.

In order to process those streams, we need a scalable way of handling the amount

of data that is coming in. And that's where cloud function fits in perfectly.

We handle that data with event-driven architecture. We use functions to persist

that data, and to generate further events on that data, that are then utilized and

consumed by applications, our customers and users. In our analytics

platform, we analyzed the set of data and we generate value predictive in a sense

that we can predict the failure rate to a certain percentage that is about to

happen in the future for our equipment. And this allows us to perform predictive

maintenance. And this is kind of the whole concept that we have behind our

24/7 connected services, which is a promise to our customers that, the

equipment is connected to the cloud and we are monitoring it and that's where we

generate the real a value for our customers. At the

moment we use almost all aspects of the IBM cloud. We use storage from the cloud.

We use cloud function. We use messaging services. We

use IoT services. So a number of services already in use and platform and that use

will only grow as our digital footprint in the industry grows.

**Overview of Cloud Service Models**

In our introductory cloud video we briefly talked about the three service

models available on the cloud. Infrastructure as a Service, Platform as

a Service, and Software as a Service. Before we dive into the details of each

of these service models in the subsequent videos, let's get started with

a quick overview of these models. Hi, I'm Tessa Rhodes, and I'm a designer on IBM

Cloud. So let's start with the foundation here with IaaS. IaaS is a set of compute

networking and storage resources that have been virtualized by a vendor so

that a user can access and configure them any way they want. In design we

have a concept of talking about users, called personas, and the persona for IaaS

is a system admin, or an IT admin. Let's jump up to the top with Software as a

Service that's the easy one. Software as a Service is just software that you

don't have to install on your machine and you don't have to manually update.

And so the user for Software as a Service could be anyone. In fact, if you're

watching this on YouTube right now, then you're a user of Software as a Service.

It's usually charged on a subscription model rather than a one-time license fee.

And that brings us to Platform as a Service. PaaS takes advantage of all the

virtualized resources from Iaas and then just abstracts them away, so the

user doesn't have to worry about managing any of those virtualized

resources. The user for PaaS is not a system admin, usually. It's usually a dev.

In IBM we call this dev, Jane.

That's the name of our persona. And so this whole metaphor, the pyramid metaphor,

is meant to indicate that as you move down, you're increasing complexity in

terms of your knowledge and management of infrastructure resources

and you're increasing the ease of use.

Another metaphor I use when I'm talking to the designers on my team about PaaS,

is having to do with a car. So in this metaphor IaaS is like leasing

a car. So if you've ever leased a car, you probably did a lot of research, and

you care about the specs of the car and their performance. You care about the

color of the car, what kind of car it is. You're the one driving and you're the one paying

for it. You're also paying for the gas and any tolls or maintenance. With

Platform as a Service in this metaphor, that's more like renting a car. So say

you're on vacation and you just got off at the air Airport and you're going to pick

up your rental car. You don't really care what color it is. You don't even care

about the specs of it, but you're still driving and you're paying for the gas

and any tolls you go through. Software as a Service is again the easiest one. That

one's more like getting a taxi or an uber. So with the taxi or an uber, you

don't care at all about what kind of car it is, what color it is and in fact,

you're not even the one driving, or paying for gas, or any tolls because

that's baked into the price. So let's see what it means in terms of cloud

computing and its three service models. With IaaS, the cloud provider manages the

physical resources, data centers, cooling power, Network and security, as well as

computing resources that include servers and storage. With PaaS, the provider, in

addition to the computing resources, also manages the platform infrastructure

which includes the operating systems, development tools, databases, and business

analytics. In the SaaS model, in addition to the infrastructure and the platform

resources, the provider also hosts and manages the applications and data.

**IaaS - Infrastructure as a Service**

In this video, we will discuss the Infrastructure-as-a-Service model in more detail.

Infrastructure-as-a-Service, commonly referred to as “IaaS,” – or simply “eye-es”

-is a form of cloud computing that delivers fundamental compute, network, and storage

resources to consumers on-demand, over the internet, on a pay-as-you-go basis.

The cloud provider hosts the infrastructure components traditionally present in an on-premises

data center as well as the virtualization or hypervisor layer.

In an IaaS Cloud environment, customers can create or provision virtual machines (or VMs)

in their choice of Region and Zone available from the Cloud Provider.

These VMs typically come pre-installed the customer’s choice of operating system.

The customers can then deploy middleware, install applications, and run workloads on

these VMs.

They can also and create storage for their workloads and backups.

Cloud providers often provide customers the ability to track and monitor the performance

and usage of their cloud services and manage disaster recovery.

Let’s look at the key components of cloud infrastructure:

Physical data centers: IaaS providers manage large data centers that contain the physical

machines required to power the various layers of abstraction on top of them.

In most IaaS models, end users do not interact directly with the physical infrastructure

but experience it as a service provided to them.

Compute: IaaS providers manage the hypervisors and end-users programmatically provision virtual

instances with desired amounts of compute, memory, and storage resources.

Cloud compute typically comes with supporting services like auto scaling and load balancing

that provide scalability and high performance.

Network: Users get access to networking resources on the cloud through virtualization or programmatically,

through APIs.

Storage: There are three types of cloud data storage: object, file, and block storage.

Object storage is the most common mode of storage in the cloud, given that it is highly

distributed and resilient.

IaaS supports a wide array of use cases.

We’ll look at some typical use cases here.

Organizations today are using cloud infrastructure services to enable their teams to set up test

and development environments faster, helping create new applications more quickly.

By abstracting the low-level components, cloud infrastructure is helping developers focus

more on business logic than infrastructure management.

Business continuity and disaster recovery require a significant amount of technology

and staff investments.

IaaS is helping organizations reduce this cost and make applications and data accessible

as usual during a disaster or outage.

Organizations are using cloud infrastructure to deploy their web applications faster and

also scale infrastructure up and down as demand fluctuates.

Organizations are leveraging the high-performance computing capabilities of cloud infrastructure

to solve complex problems involving millions of variables and calculations such as climate

and weather predictions and financial modeling.

Mining massive data sets to locate valuable patterns, trends, and associations requires

a huge amount of processing power.

Cloud infrastructure not only provides the required high-performance computing but also

makes it economically viable.

While there are some concerns regarding the lack of transparency in the cloud infrastructure’s

configuration and management and dependency on a third-party for workload availability

and performance, Infrastructure-as-a-Service is the fastest growing cloud model today.

**PaaS - Platform as a Service**

Platform-as-a-Service, commonly referred to as “PaaS” or simply "pass", is a a cloud computing model

that provides customers a complete platform to develop, deploy

manage, and run applications created by them or acquired from a third-party.

The PaaS provider hosts everything—servers, networks, storage, operating system, application

runtimes, APIs, middleware, databases, and other tools at their data center.

The provider also takes responsibility for the installation, configuration, and operation

of the application infrastructure, leaving the user responsible for only the application

code and its maintenance.

Customers pay for this service on a usage basis and purchase resources on-demand.

With IaaS, the cloud provider offers access to ‘raw’ computing resources, such as

servers, storage, and networking, while the user is responsible for the platform and application

software.

With PaaS, the cloud service provider delivers and manages the entire platform infrastructure,

abstracting users from the lower-level details

of the environment.

Let’s look at some essential characteristics of Platform-as-a-Service or PaaS :

PaaS clouds are distinguished by the high level of abstraction they provide to the users,

eliminating the complexity of deploying applications, configuring infrastructure, and provisioning

and configuring supporting technologies like load balancers and databases.

PaaS clouds provide services and APIs that help simplify the job of developers in delivering

elastically scalable and highly available cloud applications.

These services typically include a variety of capabilities such as APIs for distributed

caching, queuing and messaging, file and data storage, workload management, user identity,

and analytics, thus eliminating the need to integrate disparate components.

The PaaS runtime environment executes end-user code according to policies set by the application

owner and cloud provider.

Many of the PaaS offerings provide developers with rapid deployment mechanisms, or “push

and run” mechanism, for deploying and running applications.

PaaS offerings support a range of application infrastructure or middleware capabilities,

such as application servers, database management systems, business analytics servers, mobile

back-end services, integration services, business process management systems, rules engines,

and complex event processing systems.

Such an application infrastructure assists developers by reducing the amount of code

that must be written while expanding the application’s functional capabilities.

The most important use case for PaaS is strategic—build, test, deploy, enhance, and scale applications

rapidly and cost-effectively.

Let’s look at some more use cases for PaaS: API development and management: Organizations

are using PaaS to develop, run, manage, and secure APIs and microservices, which are loosely

coupled, independently deployable components and services.

Internet of Things, or IoT: PaaS clouds support a broad range of application environments,

programming languages, and tools used for IoT deployments.

Business analytics/intelligence: PaaS tools allow organizations to analyze their data

to find business insights that enable more informed business decisions and predictions.

Business Process Management, or BPM: Organizations are using the PaaS cloud to access BPM platform

delivered as a service.

Master data management, or MDM: Organizations are leveraging the PaaS cloud to provide a

single point of reference for critical business data such as information about customer transactions

and analytical data to support decision making.

Let’s look at some advantages of using PaaS: Scalability, made possible because of the

rapid allocation and deallocation of resources with a pay-as-you-use model offered by PaaS.

The APIs, support services, and middleware capabilities that PaaS clouds provide assist

developers in focusing their efforts on application development and testing, resulting in faster

time to market for their products and services.

Middleware capabilities also reduce the amount of code that needs to be written while expanding

the application’s functional capabilities.

Greater agility and innovation because using PaaS platforms means that you can experiment

with multiple operating systems, languages, and tools without having to invest in these

resources.

You can evaluate and prototype ideas with very low risk exposure resulting in faster,

easier, less-risky adoption of a wider range of resources.

Some of the key PaaS offerings available in the market today include AWS Elastic Beanstalk,

Cloud Foundry, IBM Cloud Paks, Windows Azure, RedHat OpenShift, Magento Commerce

Cloud, Force.com, and Apache Stratos.

PaaS clouds do come with some risks—risks that all cloud offerings have in general,

such as information security threats and dependency on the service provider’s infrastructure.

Services can get impacted when a service provider’s infrastructure experiences downtime.

Customers also don’t have any direct control over the changes that may take place when

a provider makes changes in its strategy, service offerings, or tools.

But the benefits can far outweigh these risks. PaaS continues to experience strong growth and

is predicted to become the prevailing platform delivery model moving forward.

**SaaS - Software as a Service**

Software-as-a-Service, “SaaS”, is a cloud offering that provides

users with access to a service provider’s cloud-based software.

SaaS providers maintain the servers, databases, and code that constitute an application.

They also manage access to the application, including security, availability, and performance.

Applications reside on a remote cloud network, and users use these applications without having

to maintain and update the infrastructure.

Core business processes supported by SaaS today include

email and collaboration via offerings such as Microsoft's Office 365 and Google's Gmail,

Customer Relationship Management via services such as NetSuite CRM and Salesforce,

Human Resource Management via services from Workday and SAP SuccessFactors,

financial management, billing and collaboration, and many more.

According to Forrester Research, SaaS has overtaken on-premises solutions in categories

such as human capital management (HCM), customer relationship management (CRM), and collaboration.

Solutions once available with several different deployment options are now SaaS-only.

Let’s look at key characteristics of Software-as-a-Service. SaaS clouds have a multitenant architecture.

Infrastructure and code are maintained centrally and accessed by all users.

SaaS makes it easy for users to manage privileges, monitor data use, and ensure everyone sees

the same information at the same time.

Security, compliance, and maintenance are all part of the offering.

Users can customize applications to fit their business processes with point-and-click ease.

Users can customize the UI to work with their branding guidelines; they can modify data

fields and enable or disable features within the business process.

These customizations are preserved through upgrades.

Users pay for the use of the services via a subscription model.

The use of resources can be scaled easily, depending on service needs.

Key benefits of adopting SaaS: Businesses can directly procure solutions

without upfront capital and assistance from IT, greatly reducing the time from decision

to value from months to days.

SaaS greatly increases workforce productivity and efficiency.

Users can access core business apps from wherever they are. They can also buy and deploy apps in

minutes, reducing the typical obstacles enterprises have to test the products they they might

use.

Using SaaS applications, individuals and small enterprises can spread out their software

costs over time.

Let’s look at some use cases for SaaS: Organizations are moving to SaaS for their

core business needs as part of their strategic transformation to reduce on-premises IT infrastructure

and reduce capital expenditure.

Oragnzaitions are leveraging SaaS to avoid the need for ongoing upgrades, maintenance,

and patching, done traditionally by internal IT resources; applications run reliably with

minimal input, for example, email servers and office collaboration and productivity

tools.

Organizations are increasingly opting for SaaS eCommerce Platforms to manage their websites,

marketing, sales, and operations.

With SaaS, organizations are able to take advantage of the resilience and business continuity

of the cloud provider.

Enterprises are now developing SaaS integration platforms (or SIPs) for building additional

SaaS applications, moving SaaS beyond standalone software functionality to a platform for mission-critical

applications.

SaaS does evoke some concerns, as well.

Primary among them being data ownership and data safety.

Security is an important consideration when you’re allowing a third-party to maintain

business-critical data.

And application access relies on a good internet connection. If you’re not connected, you

cannot access the apps.

But the benefits far outweigh the concerns, with SaaS making up the largest segment of

the cloud market today.

**Public Cloud**

In our introductory cloud video, we briefly mentioned the three deployment models for

cloud.

In this video, we will discuss the Public Cloud deployment model in more detail.

Deployment models indicate where the infrastructure resides, who owns and manages it, and how

cloud resources and services are made available to users.

The three cloud deployment models include—Public Cloud, Private Cloud, and Hybrid Cloud.

In a public cloud model, users get access to servers, storage, network, security, and

applications as services delivered by cloud service providers over the internet.

Using web consoles and APIs, users can provision the resources and services they need.

The cloud provider owns, manages, provisions, and maintains the infrastructure, renting

it out to customers either for a subscription charge or usage-based fee.

Users don’t own the servers their applications run on or storage their data consumes, or

manage the operations of the servers, or even determine how the platforms are maintained.

In very much the same way that we consume and pay for utilities such as water, electricity,

or gas in our everyday lives, we don’t own any of these cloud resources—we make an

agreement with the service provider, use the resources, and pay for what we use within

a certain period.

Public clouds offer significant cost savings as the provider bears all the capital, operational,

and maintenance expenses for the infrastructure and the facilities they are hosted in.

It makes scalability as easy as requesting more capacity.

However, with a public cloud, the user does not have any control over the computing environment

and is subject to the performance and security of the cloud provider’s infrastructure.

There are several public cloud providers in the market today, such as Amazon Web Services,

Microsoft Azure, IBM Cloud, Google Cloud Platform, and Alibaba Cloud.

While all providers include a common set of core services, such as servers, storage, network,

security, and databases, they also offer a wide spectrum of niche services with varied

payment options.

Let’s talk about some of the characteristics of a public cloud:

A public cloud is a virtualized multi-tenant architecture enabling tenants or users to

share computing resources, residing outside their firewalls.

The cloud providers pool of resources, including infrastructure, platforms, and software, are

not dedicated for use by a single tenant or organization.

Resources are distributed on an as-needed basis offered through a variety of subscription

and pay-as-you-go models.

Public clouds have significant benefits. We’ll go over some of these benefits here:

Vast on-demand resources are available, allowing applications to respond seamlessly to fluctuations

in demand.

Considering the large number of users that share the centralized cloud resources on-demand,

the public cloud offers the most significant economies of scale.

The sheer number of server and network resources available on the public cloud means that a

public cloud is highly reliable. If one physical component fails, the service still runs unaffected

on the remaining available components.

It’s also important to note some concerns users have regarding public clouds—key among

them being security and data sovereignty compliance.

Security issues such as data breaches, data loss, account hijacking, insufficient due

diligence, and system and application vulnerability seem to be some of the fears users continue

to have concerning security in the public cloud.

With data being stored in different locations and accessed across national borders, it has

also become increasingly critical for companies to be compliant with data sovereignty regulations

governing the storage, transfer, and security of data.

A service provider’s ability to not just keep up with the regulations, but also the

interpretation of these regulations, is a concern shared by many businesses.

Let’s look at some common use cases for public cloud.

Organizations are increasingly opting to access cloud-based applications and platforms so

their teams can focus on building and testing applications, and reducing time-to-market

for their products and services.

Businesses with fluctuating capacity and resourcing needs are opting for the public cloud.

Organizations are using public cloud computing resources to build secondary infrastructures

for disaster recovery, data protection, and business continuity.

More and more organizations are using cloud storage and data management services for greater

accessibility, easy distribution, and backing up their data.

IT departments are outsourcing the management of less critical and standardized business

platforms and applications to pubic cloud providers.

**Private Cloud**

The National Institute of Standards and Technology defines Private Cloud as cloud infrastructure

provisioned for exclusive use by a single organization comprising multiple consumers,

such as the business units within the organization.

It may be owned, managed, and operated by the organization, a third party, or some combination

of them, and it may exist on or off premises.

Private cloud platforms can be implemented internally or externally.

When the platform is provisioned over an organization’s internal infrastructure, it runs on-premises

and is owned, managed, and operated by the organization.

When it is provisioned over a cloud provider’s infrastructure, it is owned, managed, and

operated by the service provider.

This external private cloud offering that resides on a cloud service provider’s infrastructure

is called a Virtual Private Cloud, or VPC.

A VPC is a public cloud offering that lets an organization establish its own private

and secure cloud-like computing environment in a logically isolated part of a shared public

cloud.

Using a VPC, organizations can leverage the dynamic scalability, high availability, and

lower cost of ownership of a public cloud, while having the infrastructure and security

tailored to the organization’s unique needs.

Virtual Private Clouds are offered by most Public Cloud providers such as IBM and Amazon.

A private cloud is a virtualized environment modeled to bring in the benefits of a public

cloud platform without the perceived disadvantages of an open and shared public platform.

Users of a private cloud, such as Developers and Business Units in an organization, still

get to leverage benefits such as economies of scale, granular scale, operational efficiencies,

and user self-service, while exercising full control over access, security, and compliances

specific to their organization and business.

Private clouds provide you with 1.The ability to leverage the value of cloud

computing using systems that are directly managed or under perceived control of the

organization’s internal IT.

2.The ability to better utilize internal computing resources, such as the organization’s existing

investments in hardware and software, thereby reducing costs.

3.Better scalability through virtualization and “cloud bursting,” i.e., leveraging

public cloud instances for a period of time but returning to the private cloud when the

surge is met.

4.Controlled access and greater security measures customized to specific organizational needs.

5.The ability to expand and provision things in a relatively short amount of time, providing

greater agility.

Organizations may choose to opt for private cloud because of various reasons—because

their applications provide a unique competitive advantage, there are security and regulatory

concerns, or because the data is highly sensitive and subject to strict industry or governmental

regulations.

Let’s look at some common use cases for a private cloud:

A private cloud is an opportunity for organizations to modernize and unify their in-house and

legacy applications.

Moving these applications from their dedicated hardware to the cloud also allows them to

leverage the power of the compute resources and multiple services available on the cloud.

2.Using the private cloud, organizations are integrating data and application services

from their existing applications with public cloud services.

This allows them to leverage their private cloud’s compute capability for the larger

jobs while pulling data into an application on a private cloud to leverage public cloud

services — essentially opening their data centers to work with cloud services.

3.Application portability is a key feature of cloud platforms.

Using the private cloud gives organizations the ability to build applications anywhere,

and move them anywhere, without having to compromise security and compliance in the

process.

4.Some of the key reasons that may prevent an organization from moving to a public cloud

include security and regulatory concerns, and data sensitivity.

A private cloud offers these organizations the benefits of on-demand enterprise resources

while exercising full control over critical security and compliance issues from within

the environment of their dedicated cloud.

**Hybrid Cloud**

Hybrid Cloud is a computing environment that connects an organization's on-premise

private cloud and third-party public cloud into a single flexible

infrastructure for running the organization's applications and workloads.

The mix of public and private cloud resources gives organizations

the flexibility to choose the optimal cloud for each application or

workload, workloads move freely between the two clouds as needs change.

Organizations can choose to run the sensitive highly regulated and

mission-critical applications or workloads with reasonably constant performance and

capacity requirements on private Cloud infrastructure.

While deploying the less sensitive and

more dynamic workloads on the public cloud.

With proper integration and orchestration between the public and

private clouds, you can leverage both clouds for the same workload.

For example, you can leverage additional public cloud capacity to accommodate

a spike in demand for a private cloud application also known as cloud bursting.

The key tenants of a hybrid cloud are interoperability,

scalability and portability.

Hybrid cloud is interoperable, which means, that the public and

private cloud services can understand each other's APIs, configuration,

data formats, and forms of authentication and authorization.

When there is a spike in demand a workload running on the private cloud

can leverage the additional public cloud capacity making it scalable.

A hybrid cloud is also portable, since you're no longer locked in with a specific

vendor, you can move applications and data not just between on-premise and

cloud systems, but also between cloud service providers.

Hybrid is about taking the best of both worlds.

There are two common types of hybrid clouds, hybrid monocloud and

hybrid multi-cloud.

A Hybrid Monocloud is a hybrid cloud with one cloud provider,

while a Hybrid Multicloud is an open standards-based stack that can

be deployed on any public cloud infrastructure.

The difference lies in the flexibility that the hybrid multicloud offers

organizations to move workloads and environments from one vendor to another.

There is also a and of hybrid multicloud called the, Composite Multicloud,

which makes this flexibility even more granular as it distributes

single applications across multiple providers, allowing you to

move application components across cloud services and vendors as needed.

Hybrid cloud offers significant benefits in areas of security and compliance,

scalability and resilience, resource optimization, and cost saving.

A hybrid cloud lets organizations deploy highly regulated or

sensitive workloads in a private cloud while running

the less-sensitive workloads on a public cloud.

Using a hybrid cloud you can scale up quickly, inexpensively, and

even automatically using the public cloud infrastructure,

all without impacting the other workloads running on your private cloud.

Because you're not locked-in with a specific vendor and also don't have to

make either-or decisions between the different cloud models,

you can make the most cost-efficient use of your infrastructure budget.

You can maintain workloads where they are most efficient, spin-up environments using

pay-as-you-go in public cloud, and rapidly adopt new tools as you need them.

A typical organization will have a range of applications and

workloads spread across private public and traditional IT environments.

This represents a range of opportunities for

optimization via a hybrid cloud approach.

Let's look at some increasingly common hybrid cloud use cases.

Software-as-a-Service integration,

through hybrid integration organizations are connecting software-as-a-service

applications available in the public cloud to their existing public cloud,

private cloud, and traditional IT applications to deliver new solutions.

Data and AI integration, organizations today are creating richer and

more personal experiences by combining new data sources on the public cloud.

Such as weather, social, the Internet of things, CRM, and ERP,

with existing data in analytics, machine learning, and AI capabilities.

Enhancing legacy apps, an increasing number of organizations are using

public cloud services to upgrade their user experience of their

on-premises applications and deploy them globally to new devices,

while incrementally modernizing their Core Business Systems.

VM ware migration, more and more organizations are lifting and shifting

their on-premises virtualized workloads to a public cloud without conversion or

modification to reduce their on-premises data center footprint and

position themselves to scale without added capital expense.

Hybrid cloud is a concept that's been around for quite some time, but

we're finding that it's becoming increasingly used to architect and

modernize existing or legacy applications.

According to research we found that 75% of non-cloud applications

will be moving to the cloud in the next three years.

This goes to say that if you're not already thinking about your

hybrid cloud strategy, you may be falling behind.

**Overview of Cloud Infrastructure**

After choosing the cloud service model and the cloud type offered by vendors, customers

need to plan the infrastructure architecture.

The infrastructure layer is the foundation of the cloud.

This layer consists of physical resources that are housed in Regions, Zones and Data

Centers.

A Cloud provider’s IT environment is typically distributed across many regions around the

world.

A cloud region, is a geographic area or location where a cloud provider’s infrastructure

is clustered, and may have names like NA South or US East.

The cloud regions are isolated from each other so that if one region was impacted by a natural

disaster like an earthquake, the cloud operations in other regions would keep running.

Each Cloud Region can have multiple Zones (or Availability Zones or AZ for short), which

are typically distinct Data Centers with their own power, cooling and networking resources.

These Zones can have names like DAL-09 or us-east-1.

The isolation of zones improves the cloud’s overall fault tolerance, decreases latency,

and avoids creating a single shared point of failure.

The Availability Zones (and DataCenters within them) are connected to other AZs and regions,

private datacenters and the Internet using very high bandwidth network connectivity.

A cloud data center is a huge room or a warehouse containing cloud infrastructure.

These data centers contain pods and racks or standardized containers of computing resources

such as servers, as well as storage, and networking equipment - virtually everything that a physical

IT environment has.

Computing Resources: Cloud providers offer several compute options

– Virtual Servers, Bare Metal Servers, and “Serverless” computing resources.

Most of the servers in a cloud datacenter run hypervisors to create virtual servers

or virtual machines (also called VMs for short), that are software-based computers, based on

virtualization technologies.

Other servers in the racks are bare metal servers that are physical servers that aren’t

virtualized.

Customers can provision VMs and Bare Metals servers as and when they need them and run

their workloads on them.

Cloud users can also run their workloads on serverless computing resources, which are

an abstraction layer on top of virtual machines.

We will talk about all three compute options in greater detail in subsequent videos.

Storage: Information and data can consist of files,

code, documents, images, videos, backups, snapshots, and databases and can be stored

in many different types of storage options on the Cloud.

Bare Metal Servers and Virtual Servers are provisioned with default storage in local

drives.

Since these cloud servers can be provisioned and decommissioned by customers on demand

and freed up for use by other users, any information stored in a local drive can be lost when you

delete or decommission a cloud server.

However there are other storage options available on the cloud to persist data that you can

choose depending on factors like how important your data is, how quickly you want to be able

to access it, how often you access it, and how secure you need it to be.

These additional storage options include Block storage, File storage, and Object storage.

Block and file storage modes are commonly used in traditional data centers, but often

struggle with scale, performance and distributed characteristics of cloud.

Object storage is the most common mode of storage in the cloud as it’s both highly

distributed and resilient.

We will examine Object Storage and the other storage options in more detail in later videos.

Networking: Networking infrastructure in a cloud data center

includes traditional networking hardware like routers and switches, but more importantly

for users of the Cloud, the Cloud providers have Software Defined Networking (or SDN)

options where certain networking resources are virtualized or made available programmatically,

through APIs.

This allows for easier network provisioning, configuration, and management in the cloud.

When servers in the cloud are provisioned, you need to setup their public and private

network interfaces.

The public network interfaces, as the name suggests, connect the servers to the public

internet, whereas the private ones provide connectivity to your other cloud resources

and help keep them secure.

As in the physical IT world, network interfaces in the cloud need to have IP addresses and

subnets either assigned automatically or configured.

In a cloud environment it is even more important to configure which network traffic and users

can access your resources, which can be done by setting up Security Groups and Access Control

Lists (or ACLs).

For further security and isolation of your resources in the cloud, most Cloud providers

provide Virtual Local Area Networks (VLANs), Virtual Private Clouds (VPCs), and Virtual

Private Networks (VPNs).

Some of the traditional hardware appliances such as firewalls, load balancers, gateways

and traffic analyzers can also be virtualized and made available as services in the cloud.

Another networking capability provided by the Cloud Providers is Content Delivery Networks

or CDNs, that distribute content to multiple points throughout the world so users accessing

the content can access it more quickly by getting it from a point nearest to them.

We will learn more about some of these cloud networking options and terminology in subsequent

videos.

Cloud infrastructure is constantly advancing and improving.

**Virtualization and Virtual Machines Explained**

Hi. My name is Kaleigh Bovey, with the IBM Cloud team and today we're going to be

talking about virtualization. As you know virtualization is a fairly old

technology, but it's still super relevant to building our cloud computing strategy

today. So, first off, what is virtualization? Simply put, virtualization

is the process of creating a software based, or virtual, version of something,

whether that be compute, storage, networking, servers, or applications and

what makes virtualization feasible, is something called the hypervisor. We're

going to write that here. What a hypervisor is, is it's simply a piece of

software that runs above the physical server, or host. And there are a couple

different types of hypervisors out there. What they do is essentially pull the

resources from the physical server and allocate them to your virtual

environments. There are two main types of hypervisors out there. One being Type 1.

Very simple to remember. And 2, you guessed it, Type 2. So let's start

with Type 1. A Type 1 hypervisor is a hypervisor that

is installed directly on top of the physical server. They're also called

bare-metal hypervisors. So we'll write that up here. Remember these are the most

frequently typed of use hypervisors and they're most secure, they lower the

latency, and these are the ones that you'll see in the market the most. Some

examples would be VMware, ESXi, or Microsoft Hyper-v, or even open-source

KVM. The other type of hypervisor is a Type 2 hypervisor, over here. And what

makes these different is that there is a layer of host OS that sits between the

physical server and the hypervisor. So, by that nature they are also called, Hosted.

These are a lot less frequent. They're mostly used for end-user virtualization and

you might see some in the market that are called: Oracle, VirtualBox,

or VMware Workstation. Again, they're a lot less frequent. They're a bit more...

They have a higher latency than a Type 1 hypervisor. So once you have your

hypervisor installed, you can build virtual environments, or virtual machines,

or simply put, VMs. So let's spin up some environments.

So, what makes a VM a VM? A VM is simply a software based computer. They're run like

a physical computer. They have an operating system and applications, and

they're completely independent of one another, but you can run multiple of them

on a hypervisor and the hypervisor manages the resources that are allocated

to these virtual environments from the physical server. So, because they're

independent you can run different operating systems on different virtual

machines. You could run Windows here, or Linux here, or UNIX here for example.

Because they're independent they're also extremely portable. You can move a

virtual machine from one hypervisor to another hypervisor on a completely

different machine almost instantaneously, which gives you a lot of

flexibility and a lot of portability within your environment. So looking at

all of this - this is the core virtualization as a process. So let's

talk about a couple key benefits that you want to take away from this. 1) Cost

savings.

When you think about this and the fact that you can run multiple virtual

environments from one piece of infrastructure, means that you can

drastically reduce your physical infrastructure footprint. This is

consolidation at its core. And the fact that you don't have to maintain nearly

as many servers, run as much electricity, save on maintenance costs, means that you

save on your bottom line at the end of the day. 2) Would

be agility and speed. Like I said, spinning up a virtual machine is

relatively easy and quick - a lot more simple than provisioning an entire new

environment for your developers if they say they want to spin up a new

environment so that they can run a test scenario. Whatever it might be,

virtualization makes that process a lot simpler and quicker and 3) lowers

your downtime.

So, let's say that this host goes out unexpectedly. The fact that you can move

virtual machines from one hypervisor to another, on a different physical server,

means that you have a great backup plan in place right? So, if this host goes down

you can simply move your VMs very quickly to another hypervisor on a

machine that is working. Virtualization and VMs are at the center of cloud

computing and provide many benefits.

**Types of Virtual Machines**

Virtual Machines or VMs are also known as Virtual Servers or Virtual Instances, or simply

Instances, depending on the cloud provider.

The various cloud providers make VMs available in a variety of configurations and deployment

options to serve different use cases.

When you create a virtual server in the cloud, you specify the Region and Zone or Data Center

you want the server to be provisioned in and the Operating System you want on it.

You can choose between shared (that is, a multi-tenant) VMs or dedicated (that is,

a single-tenant) VMs.

You can also choose between hourly or monthly billing, and select storage and networking

options for the virtual server.

Now let’s look at a few different types of VMs that can be provisioned in the cloud.

Shared or Public Cloud VMs are provider-managed, multi-tenant deployments that can be provisioned

on-demand with predefined sizes.

Being multi-tenant means that the underlying physical server is virtualized and is shared

across other tenants or users.

To satisfy different workloads, cloud providers offer predefined sizes and configurations

ranging from a single virtual core and a small amount of RAM to multiple virtual cores and

much larger amounts of RAM.

For example there can be configurations for Compute Intensive workloads, Memory intensive

workloads, or High Performance I/O. Rather than pick from only pre-defined sizes, some

providers also offer custom configurations that allow users to define the number of cores

and RAM and local storage characteristics.

Public VMs are usually priced by the hour (or in some cases even seconds) and configurations

start as low as pennies per hour.

Some providers also let you get monthly VMs, which can result in some cost savings if you

know you will run the VM for at least a month, but if you decide to de-commision the VM in

the middle of the month, you will still be charged for the full month.

Transient or Spot VMs take advantage of unused capacity in a cloud data center.

Cloud providers make this unused capacity available to users at a much lower cost than

regular VMs of similar sizes.

Although the Transient VMs are available at a huge discount, the Cloud provider can choose

to de-provision them at any time and reclaim the resources for provisioning regular, higher-priced,

VMs.

Because you run the risk of losing these VMs when capacity in the data center decreases,

these VMs are great for non-production workloads such as testing and developing applications.

They are also useful for running stateless workloads, testing scalability, or running

big data and high performance computing (HPC) workloads at a low cost.

Reserved virtual server instances allow you to reserve capacity and guarantee resources

for future deployments.

You reserve desired amount of virtual server capacity, provision instances from that capacity

when you need them, and choose a term, such as 1 year or 3 years, for your reserved capacity.

You're guaranteed this capacity within the data center of your choice for the life of

the contract term.

By committing to a longer term, you can also lower your costs compared to hourly or monthly

instances.

This can be useful when you know you require at least a certain level of cloud capacity

for a specific duration.

If you exceed your reserved capacity, you can always choose to supplement your unplanned

usage and capacity requirements with hourly or monthly VMs.

Note however that not all predefined VMs families or configurations may be available as reserved.

Dedicated hosts offer single-tenant isolation.

This means that only your VMs run on a given host so they can make exclusive use of full

capacity and resources of the underlying hardware.

When provisioning a dedicated host you to specify the data center and POD in which you

want your host placed.

You then assign instances, or virtual machines, to a specific host.

This allows for maximum control over workload placement.

Dedicated hosts are typically used for meeting compliance and regulatory requirements or

meet specific licensing terms.

Virtualization and VMs are at the center of cloud computing and provide many benefits.

**Bare Metal Servers**

A bare metal server is a single-tenant, dedicated physical server.

In other words, it's dedicated to a single customer.

The cloud provider actually takes the physical server and plugs it into a rack in a data

center for customers.

The cloud provider manages the server up to the operating system or OS, which means if

anything goes wrong with the hardware or rack connection, they will fix or replace it and

then reboot the server.

The customer is responsible for administering and managing everything else on the server.

Bare metal servers are either preconfigured by the cloud provider to meet workload packages

or they can be custom-configured as per customer specifications.

This includes the processors, RAM, hard drives, specialized components, and the OS.

Customers can also install their own OS and can install certain hypervisors that aren't

available from the cloud provider, and thus create their own virtual machines and farms.

With bare metal servers you can also add GPUs, which are designed for accelerating scientific

computation, data analytics, and rendering professional grade virtualized graphics.

Because bare metal servers are physical machines, they take longer to provision than virtual

servers.

Pre-configured builds of bare metal can take 20 to 40 minutes to provision and custom-builds

can take around three or four hours, but these provisioning times can vary by Cloud provider.

As Bare Metal servers are dedicated for use by a single client at any given time, they

tend to be more expensive than similarly sized Virtual Machines.

Also note that unlike virtual servers, not all cloud providers provide Bare Metal servers.

Since bare metal servers are fully customizable, they can do what a customer wants in the most

demanding environments.

Bare metal servers are dedicated and intended for long term, high performance use in highly

secure and isolated environments.

Clients have full access and control of bare metal servers because there’s no hypervisor

required.

As there is no sharing underlying server hardware with other customers, Bare metal servers fulfil

the demanding needs of high-performance computing or HPC and data intense applications that require

minimal latency-related delays.

These servers also excel in big data analytics applications and GPU-intensive solutions.

Some workload examples that bare metal servers satisfy are ERP, CRM, AI, Deep Learning, and

virtualization.

If you use any applications that require high degrees of security control or apps that you’ve

typically run in an on-premises environment, then a bare metal server is a good alternative

in the cloud.

When comparing bare metal servers to virtual servers, some of the most important considerations

are found in customer need.

Bare metal servers work best for: CPU and I/O intensive workloads, excel with highest

performance and security, satisfy strict compliance requirements, and offer complete flexibility,

control and transparency but come with added management and operational overhead.

Whereas, virtual servers are rapidly provisioned, provide an elastic and scalable environment,

and are low cost to use, however since they share underlying hardware with other virtual

servers, they can be limited in throughput and performance.

**Secure Networking in Cloud**

As cloud environments gain greater adoption, and digital data invites rapidly increasing

cybersecurity threats, building secure networks on the cloud is crucial.

Let’s look at how we can build a secure cloud networking presence.

As one might expect, the notion of building a cloud network is not much different from

deploying a network in an on-premises data center.

The main difference stems from the fact, that in the cloud, we use logical instances of

networking elements as opposed to physical devices.

For example, Network Interface Controllers (NICs) would be represented by vNICs in cloud

environments.

In the cloud, networking functions are delivered as a service rather than in the form of rack-mounted

devices.

To create a network in the cloud, one starts by defining the size of the network, or the

IP address range that establishes the boundaries or the cloud network.

Cloud networks are deployed in networking spaces that are logically separated segments

of the networks using options, including Virtual private Cloud (VPC) that in turn can be divided

into smaller segments called subnets.

Logically segmented cloud networks are private carveout of the cloud that offer customers

the security of private clouds and the scalability of public clouds.

Cloud resources, such as VMs or Virtual Server Instances (VSIs), storage, network connectivity

and load balancers are deployed into subnets.

Using subnets allows users to deploy enterprise applications using the same multi-tier concepts

used in on-premises environments.

Subnets are also the main area where security is implemented in the cloud.

Every subnet is protected by Access Control Lists (ACLS) that serve as a subnet-level

fire wall.

Within the subnet, one could create Security Groups that provide security at the instance

level such as VSIs.

Once you build a subnet, then it is time to add some VSIs and storage to it so that you

could run your applications.

Let’s say you have a 3-tier application that requires web access VSIs, applications

tier VSIs and backend database VSIs.

In this case, we would place the web facing VSIs into one Security Group, the Application

VSIs in a second Security Group, while the database VSIs in a third Security Group.

It goes without saying that the web-facing VISs need Internet access.

A public Gateway instance is added to the network to enable users’ access to the application

in the internet tier.

While public gateways are great for Internet access to the cloud, enterprises are interested

in extending their on-premises resources to the cloud by securely connecting them using

Virtual Private Networks, or VPNs.

When building many subnets and deploying several workloads, it becomes necessary to ensure

that applications continue to be responsive.

That is achieved with Load Balancers that ensure availability of bandwidth for the different

applications.

Enterprises with hybrid cloud environment find using dedicated high-speed connections

between clouds and on-premises resources is a more secured and more efficient way than

public connectivity solutions.

Some cloud service providers offer such connectivity, such as IBM Cloud and its Direct Link solution

that enables extending on-premises resources to the cloud as needed.

Building a cloud network entails creating a set of logical constructs that deliver networking

functionality that is akin to the data center networks that all IT professionals have come

to rely on for securing their environments and ensuring high performing business applications.

**Containers**

Containers are an executable unit of software in which application code is packaged, along

with its libraries and dependencies, in common ways so that it can be run anywhere, whether

it be on desktop, traditional IT, or the cloud.

Containers are small, fast, and portable, and unlike virtual machines, they do not need

to include a guest OS in every instance and can, instead, simply leverage the features

and resources of the host OS.

In the rest of this video, we will see how container-based technology really works.

Hi everyone.

My name is Sai Vennam and I'm a developer advocate with IBM.Today I want to talk about

containerization.

Whenever I mention containers, most people tend to default to something like Docker or

even Kubernetes these days.

But container technology has actually been around for quite some time.

It's actually back in 2008 that the Linux kernel introduced C groups, and control groups,

that basically paved the way for all the different container technologies we see today.

So that includes Docker, but also things like Cloud Foundry, as well as Rocket and other

container runtimes out there.

Let's get started with an example, and we'll say that I was a developer.

I've created a node.js application and I want to push it into production.

We'll take two different form factors to kind of explain the advantages of containerization.

Let's say that first we'll talk about VMs, and then we'll talk about containers.

So, first things first, let's introduce some of the things that we've got here.

We've got the hardware itself, just a big box.

We've got the guest, or rather, the host, operating system, as well as a hypervisor.

Hypervisor is actually what allows us to spin up VMs.

Let's take a look at this shared pool of resources with the host OS and hypervisor.

We can assume that some of these resources have already been consumed.

Next, let's go ahead and take this .js application and push it in.

And to do that, I need a Linux VM. So let's go ahead and sketch out that Linux VM.

In this VM there's a few things to note here.

So, we've got another operating system, in addition to the host OS, it's gonna be the guest OS,

as well as some binaries and libraries.

That's one of the things about Linux VMs, that even though we're working with a really

lightweight application, to create that Linux VM, we have to put that guest OS in there,

in a set of binaries and libraries.

That really bloats it out.

In fact, you know, I think the smallest node .js VM that I've seen out there is 400 plus mega

bytes, whereas the node.js runtime and app itself would be under 15.

So we've got that and we'll go ahead and let's push that .js application into it.

Just by doing that alone, we're gonna consume a set of resources.

Next, let's think about scaling this out. Right.

So we'll create two additional copies of it, and you'll notice that even though it's the

exact same application, we have to use and deploy that separate guest OS and libraries

every time.

And so we'll do that three times.

And by doing that, essentially, we can assume that for this particular hardware we've consumed

all of the all of the resources.

And there's another thing that I haven't mentioned here, but this .js application, I developed

it on my macbook.

So when I pushed it into production to get it going on the VM, and notice that there

were some issues and incompatibilities.

This is the kind of foundations is big, he said, she said issue.

Where things might be working on your local machine, and work great, but when you try

to push it into production, things start to break.

and this really gets in the way of doing agile DevOps, and continuous integration and delivery.

That's solved when you use something like containers.

There's a three-step process when kind of doing anything container related, and then

pushing, or creating, containers.

And it almost always starts with first, some sort of a manifest.

So something that describes the container itself.

So in the Docker world, this would be something like a Docker file

and in Cloud Foundry, this would be a manifest Channel.

Next, what you'll do is create the actual image itself.

So for the image, again, if you're working with something like Docker, that could be something

like a Docker image.

If you're working with Rocket it would be an ACI or application container image.

So regardless of the different containerization technologies, this process stays the same.

The last thing you end up with is an actual container itself.

You know, that contains all of the runtimes, and libraries, and binaries needed to run

an application.

That application runs on a very similar set up to the VMS, but what we've got on this

side is, again, a host operating system.

The difference here, instead of a hypervisor, we're gonna have something like a runtime

engine.

So if you're using Docker this would be the Docker engine and, you know, different containerization

technologies would have a different engine.

Regardless, it's something that runs those containers.

Again, we've got this shared pool of resources, so we can assume that that alone consumes

some set of resources.

Next, let's think about actually containerizing this technology.

We talked about the three-step process.

We create a docker file.

We build out the image.

We push it to a registry, and we have our container and we can start pushing this out

as containers.

The great thing is, these are going to be much more lightweight.

So deploying out multiple containers, since you don't have to worry about a guest OS this

time, you really just have the libraries, as well as the the application itself.

So we've scaled that out three times, and because we don't have to duplicate all of those operating

system dependencies and create bloated VMs, we actually will use less resources.

So use a different color here... and scaling that out three times, we still have a good

amount of resources left.

Next, let's say that my coworker decides, hey, for this .js application, let's take

advantage of a third party you know let's say a cognitive API - to do something like image recognition.

Let's say that we've got our third party service, and we want to access that using maybe a Python

application.

So he's created that service that acts as that third party APIs

and with our node.js application, we want to access that Python app, to then access

that service.

If we wanted to do this in VMs, I'm really tempted to basically create a VM out of both

the .js application and the Python application

because essentially that would allow me to continue to use the VMs that I have.

But that's not truly cloud native, right?

Because if I wanted to scale out the .js, but not the Python app, I wouldn't be able

to if they were running in the same VM.

So to do it in a truly cloud native way, essentially I would have to free up some of these resources.

Basically get rid of one of these VMs, and then deploy the Python application in it instead.

And you know, that's not ideal.

But with the container based approach what we can do is simply say, since we're modular,

we can say, okay, just deploy one copy of the Python application.

So we'll go ahead and do that.

There's a different color here.

And that consumes a little bit more resources and

then with those those remaining resources, the great thing about container technology,

that actually becomes shared between all the processes running.

In fact, another advantage if some of these container processes aren't actually utilizing

the CPU or memory, all of those shared resources become accessible for the other containers

running within that hardware.

So with container-based technology, we can truly take advantage of cloud native based

architectures.

We talked about things like portability of the containers.

We talked about how it's easier to scale them out.

And then overall, with this kind of three-step process and the way we push containers, allows

for more agile devops and continuous integration and delivery.

Containers streamline development and deployment of Cloud Native applications.

**Basics of Storage on Cloud**

Cloud storage is where you save data and files in the cloud.

Certain storage must be attached to a compute node before the storage can be accessed, whereas

other storage types can be directly accessed either through the public Internet or a dedicated

private network connection.

Cloud providers host, secure, manage, and maintain the cloud storage and associated

infrastructure to ensure you have access to your data when you need it.

Cloud storage services allow you to scale your capacity as you need so you only pay

for what you provision, usually on a ‘per gigabyte’ basis.

The cost of storage will vary by type but in general, the faster the read / write speed

of the storage, the higher the per gigabyte cost.

Cloud storage is available in four main types – Direct Attached, File Storage, Block Storage

and Object Storage.

Direct Attached storage, sometimes referred to as ‘Local Storage’, is storage which

is presented directly to a cloud-based server and is effectively either within the host

server chassis or within the same rack.

This storage is fast and normally only used to store a server’s operating system, although

it can have other use cases.The main two reasons why direct attached storage is not so great

for other uses besides to store the operating system is that it’s typically ‘Ephemeral’

, meaning that it only lasts as long at the compute resource it’s attached to – it

cannot be shared with other nodes and while you can use RAID techniques, it’s not as

resilient to failure as other types of storage.

File storage is typically presented to compute nodes as ‘NFS Storage’.

NFS stands for Network File System and means that the storage is connected to compute nodes

over a standard ethernet network.

NFS-mounted storage is common-place but it tends to be slower than either direct-attached

storage or block storage because the data travels over an ethernet network.

It also tends to be lower cost than either direct attached or block storage.

One advantage of File Storage is that it can be mounted or used on multiple servers at

once.

File-based storage is a simple, straightforward approach to data storage and works well for

organizing data in a hierarchical folder structure, that desktop users are familiar with.

Block storage is presented to compute nodes using high-speed fibre connections, which

means that read and write speeds are typically much faster and reliable than with file storage,

making block storage suitable for use with databases and other applications where disk

speed is important.

You typically provision block storage in ‘volumes’, which can then be mounted onto a compute node,

which it then effectively sees as another hard drive.

Volumes can normally only be mounted onto one compute node at a time.

With both File and Block storage, you may also hear the term ‘IOPS’.

IOPS stands for ‘Input/Output Operations Per Second’ and relates to the speed of

the storage or to put it another way, how quickly data can be read from or written to

the storage.

We’ll cover this in a little more detail in a later video.

Persistence is a term that is used when provisioning File or Block storage and relates to what

happens to the storage once the compute node it is attached to is terminated.

If the storage is set to ‘persist’ then it will not be deleted along with the compute

node, meaning that it and its data are preserved and available to mount onto another compute

node, though you will continue to pay for the storage.

You can also, in some cases, set the storage so that it is automatically deleted with the

compute node that it is mounted onto – in this case, as we know, it becomes Ephemeral

Storage.

Here, you will also stop paying for the storage but you will lose any data unless it is backed

up somewhere.

There are several ways to backup data in the cloud but one way to back up both File and

Block storage is to take a Snapshot.

As the term implies, this is a point in time image of the storage.

Snapshots are usually fast to create (they don’t actually write any data, or rather

they create metadata), don’t require downtime and subsequent snapshots record only changes

to the data.

They are great for returning storage to the way it was at a particular snapshot, though

note, they cannot be used to recover individual files.

The fourth kind of storage is Object storage.

This is a different type of storage in so much as it’s not attached to a compute node,

rather it is accessed via an API.

Of all the storage types, Object Storage is by far the cheapest and also the slowest in

terms of read and write speeds, but it is infinite in size to the end user.

Unlike File and Block storage where you provision a certain storage capacity and it fills up

over time, with Object Storage you can keep adding files to it and it never fills up,

you just pay for what you use.

This makes Object Storage a fantastic repository for all sorts of unstructured data types,

large and small, including documents, video, logs, backups, data from IoT, application

binaries and virtual machine images.

**File Storage**

In this video, we’re going to talk about File Storage in greater detail.

Like direct attached storage, file storage must be attached to a compute node before

it can be accessed and have data stored on it.

However, File Storage can be less expensive, more resilient to failure, and involve lesser

disk management and maintenance for you as the user to do , as compared to direct attached

storage.

You can also provision much larger amounts of File Storage and present it as a disk to

a server.

File storage is mounted from remote storage appliances.

That is, the physical disks are contained in a separate, specialised piece of hardware

and they are then connected to the compute node via the underlying infrastructure in

the datacenter.

These storage appliances are not only extremely resilient to failure, the data is also far

more secure in them as these storage appliances offer services such as encryption in transit

and encryption at rest.

These appliances are all managed by the service provider.

File Storage is mounted to compute nodes via an ethernet network – the same kind of network

that you might receive email or browse the internet over, although this ethernet network

is normally dedicated to the task.

This means it can sometimes be referred to as ‘Network Attached Storage’, ‘Network

File Storage’ or simply ‘NFS.

One of the issues with ethernet networks is that their speed can vary – the more loaded

an ethernet network is, the more likely it becomes that it’s speed or bandwidth will

be affected.

Of course, Cloud Providers build their storage networks to handle very high volumes of traffic.

But even so, consistent speed cannot be guaranteed.

Therefore, File storage tends to be used for workloads where consistently high network

speeds are not a requirement.

In terms of workloads, File Storage can typically be mounted onto more than one compute node

at a time, where the mounted disk or volume looks just like another drive on the compute

node.

The ability for File Storage to be mounted to multiple compute nodes at a time make it

an ideal solution where some sort of common storage is required – for example, a departmental

file share, a ‘landing zone’ for incoming files that need to be processed by an application,

or a repository of files that a web service might access.

In these applications, the potential variance in the speed of the connecting network is

not really an issue.

Of course, where cost is an issue, you can use file storage for other applications such

as databases, but the trade-off is speed.

When you provision file storage, one consideration you need to take into account is the IOPS

capacity of the storage.

IOPS stands for Input/Output Operations Per Second and refers to the speed at which the

disks can write and read data (note this is not the speed of the network between the storage

and the compute node).

The higher the IOPS value, the faster the speed of the underlying disk.

A higher IOPS will also normally cost more.

Understanding IOPS is important because if the IOPS value is too low for your application,

the storage can become a bottleneck and cause your application to run slowly.

Alternatively, if the IOPS is too high, you will probably be paying more that you need

to for your storage.

For example, a file share may be mounted on 30 different compute nodes and an application

writes and requests data to and from that share 60 times per minute.

You can average that out to 1 operation per second.

With this simple example, you can see that each application has different IOPS requirements.

**Block Storage**

In this video, we will discuss Block Storage and how it compares to File Storage in the

Cloud.

Block storage breaks files into chunks (or blocks) of data and Stores each block separately

under a unique address.

Like direct attached storage and file storage, block storage also must be attached to a compute

node before it can be utilized for your workloads.

Block storage, like file storage, can be mounted from remote storage appliances, making it

extremely resilient to failure, and keeping data far more secure in them, on account of

encryption in transit, and encryption at rest services, available on these appliances.

Block storage is mounted as a volume to compute nodes using a dedicated network of fibres,

through which signals move at the speed of light.

These fibre optic networks are more expensive to build than the ethernet ones which deliver

File Storage, which is one reason why Block Storage tends to have a higher price-point.

However, since the traffic is moving faster and with speed consistency, they are perfect

for workloads that need low-latency storage to work effectively.

In terms of workloads, it is important to note that unlike File Storage, which can be

mounted onto 80 computer nodes or more, Block storage is normally mounted onto only one

compute node at a time.

Since these disks run at a consistent high speed, they are perfect for workloads that

need consistently fast storage, such as databases and mail servers.

Block storage is not suitable for workloads where there needs to be some level of disk

sharing between compute nodes.

For block storage, as it is for file storage, you need to take the IOPS capacity of the

storage into account.

Most cloud providers will allow you to specify IOPS characteristics when you provision storage

and, in some cases, adjust the IOPS of your storage as you need, so if the requirements

or usage behaviour of an application changes, you can adjust accordingly.

So, to summarise the commonalities and differences between these two storage types:

Block and File Storage is taken from appliances which are maintained by the service provider.

Both are normally highly available and resilient and will often include data encryption at

rest and in transit.

File storage is attached to compute nodes using an ethernet network, so it is sometimes

called Network attached or NFS Storage.

File storage is very reliable, but the speed of the connecting network can vary, based

on load.

Block storage is attached via a high-speed fibre network, which is very reliable and

consistent.

File storage can be attached to multiple compute nodes at once.

Block storage can only be attached to one node at a time.

File storage is a good choice where file shares are required, where workloads do not require

lightning fast connectivity to storage, or where cost is a factor.

Block storage is a good choice when supporting an application that needs consistent fast

access to disk, such as databases.

Remember to consider the IOPS requirements of the application when provisioning either

file or block storage.

**Object Storage Overview**

In this video, we’re going to start to understand what Object Storage is, how data is stored

in Object Storage, and how it differs from the more traditional storage types such as

File and Block Storage.

The first thing to note about Object Storage is that you do not connect it to a particular

compute node in order to use it.

Instead, you provision an Object Storage service instance and use an API (or Application Program

Interface) to upload, download, and manage your data.

This means you can directly use Object Storage with anything that can call an API and you

don’t need an underlying compute node.

The second thing to note about Object Storage is that it’s less expensive that other cloud

storage options.

It’s per gigabyte cost is typically a couple of US cents per month and in some cases, even

less, depending on the storage tier used.

More on storage tiers later.

The third and possibly most important thing to note about Object Storage is that it’s

effectively infinite.

With file and block storage, you specify the size of the storage you want in gigabytes

or terabytes and then pay a fee based on the size you provisioned.

With Object Storage, you just consume the storage you need and pay per gigabyte cost

for what you use.

You can keep uploading files and the storage will never run out.

So, when would you use Object Storage?

Well, Object Storage is great for storing large amounts of unstructured data.

By unstructured this means that the data is not stored in any kind of hierarchical folder

or directory structure – Object Storage uses ‘buckets’, and objects are stored

within these buckets in a structurally flat way.

A bucket is a bit like a folder, in the sense that you can give them meaningful names, and

of course have different buckets for different object-types but you cannot place a bucket

within a bucket.

When an object is placed in a bucket, it also has some metadata (data about the data) added

to it, such as an object ID.

This metadata helps applications to both locate and access the object, as well as provide

information on the time that the data was stored or last accessed.

When you create a bucket, you don’t need to provide or define any sizing information. The

bucket will just hold the data that you place inside it and the service provider ensures

that there is sufficient storage capacity available.

Buckets can hold as little as a few bytes of data, right up to multiple petabytes and

you can build up the amount of data stored as slowly or quickly as you like, as well

as shrink it back down again.

The service provider also takes care of resilience and making sure that the Object Storage solution

is highly available.

Some cloud providers offer different types of buckets with different levels of resilience.

For example, they offer buckets which are resilient, but the data is only stored in

one data centre.

This is a good option where data needs to reside in a particular geographical location

or in situations where high availability is less of an issue.

They will then offer buckets which are highly available across regions, where the data is

stored multiple times in different datacentres (or zones) in the same region or even in multiple

regions.

These options usually cost more but they provide both the highest level of resilience as well

as availability for your data.

Object Storage has a very ‘flat’ storage structure, which we’ll explain in the next lesson.

This data can be anything from text files to audio and video files, from IOT data to virtual machine images, from backup files to data archives.

Pretty much any data which is static and where fast read and write speeds are not necessary

would make a good fit for object storage.

Object Storage would, however, not be suitable for running operating systems, nor applications

such as databases or anything else where the contents of the files changes.

So, to summarize what we have learned in this lesson:

Object Storage is used to store files or Objects which are static.

The data that you can store using Object Storage can be anything from text files to audio and

video files, from IOT data to virtual machine images, from backup files to data archives.

You cannot run operating systems or other applications such as databases using Object

Storage.

Objects are stored in Buckets.

You can have multiple buckets, but you cannot place buckets within buckets.

You do not need to specify a size for a bucket, you can just use as little or as much space

as you need.

Many providers offer different types of buckets with different charges for each.

Some are based on resilience and availability, while others are based on the frequency at

which the objects inside are accessed.

**Object Storage - Tiers and APIs**

In this video, we're going to look more closely at Object Storage data tiers and

Object Storage APIs.

Object Storage buckets also have storage tiers, or classes, associated with them,

and these tiers are based on how frequently the data is accessed.

A standard tier bucket is where you would store objects that are frequently

accessed.

This tier tends to have the highest per gigabyte cost associated with it.

A vault or archive tier is where you might store documents that are only

accessed perhaps only once or twice a month, or less, and

this will be offered at a lower storage cost.

Whereas there may also be cold vault tier, where you would store data that is

typically accessed only once or twice a year.

This storage often costs just a fraction of a US cent per gigabyte per month.

Often, you can also set up automatic archiving rules for

your data, meaning that if an object isn't accessed for

a period of time, it will automatically be moved to a cheaper storage tier.

The rule uses some of the object's metadata to determine when it should

be archived.

Note that, Object Storage does not come with IOPS options.

Object Storage tends to be very slow in comparison with file or block storage,

where downloads typically takes seconds if not longer to complete.

Where providers offer cold vault buckets, data retrieval from these tiers

can sometimes even take hours because the storage is kept offline.

If your application needs fast access to files,

then object storage may not be a good option.

We've mentioned that object storage is priced per gigabyte used, but

there can also be other costs related to retrieval of the data.

These costs are similarly low, but access charges can be higher for

data that is in a vault or cold vault tiers, so it is important to

ensure that the data is in the correct tier based on its frequency of access.

Object Storage does not need to be attached to a compute node for

you to access it,

rather you access object storage through an application program interface, or API.

The most common API for object storage is called the S3 API,

which is a standard based on the S3 object storage offered by AWS.

Many providers offer APIs to their object storage,

which is S3 compatible, which is useful, because it means developers can write code

which is able to access multiple vendors object storage.

The API itself is an HTTP based RESTful API, or RESTful web service.

The API call allows applications to manage object storage and buckets,

as well as put, upload, or get download objects to and from them.

Object Storage is not just for new applications,

but can be used to meet requirements for existing ones.

It can also be used as an effective solution for backup and

disaster recovery as a replacement for off-site tape based solutions,

reducing the time to restore data.

Many backup packages now include the ability to backup data

up into the Cloud using object storage.

Object Storage is more efficient than tape backup solutions,

which require tapes that need to be physically loaded into, and

removed from, tape drives and moved off-site for geographical redundancy.

So, to summarize what we have learned in this lesson,

object storage has different tiers with different charges for each.

Some are based on the frequency at which the objects inside are accessed.

Object Storage is priced per gigabyte of storage used per month plus some charges

for data retrieval.

Object Storage is much cheaper than file or block storage.

Object Storage is very slow in comparison with file and block storage.

You can often create rules which allow the automatic

archiving of objects to cheaper tiers when they are in frequently accessed.

Object Storage is accessed using an API.

Many Object Storage providers have an S3 compatible API which means developers

can create code that will work against multiple vendor Object Storage solutions.

Object Storage in the Cloud offers an effective Backup and

Disaster Recovery Solution.

**CDN - Content Delivery Networks**

A content delivery network, or CDN, is a distributed server network that delivers temporarily stored, or cached, copies of website content to users, based on the user's geographic location. A CDN stores this content in distributed locations and reduces the distance between your website visitors, and your website server. In the rest of the video, we'll learn more about Content Delivery Networks. Hi. I'm Ryan Sumner, I'm a Chief Network Architect with IBM cloud, and today I'm going to help you answer: what is a content delivery network? So, in short, a content delivery network, or CDN, is a service that accelerates Internet content delivery. In other words, the main benefit of a CDN is that it makes your website faster. Before I get into describing to you how it accomplishes that, and some of the other benefits, first I want to talk to you about some of the challenges that we have where we have users all around the world, but we don't have servers all around the world, and the experience that those users have due to that dynamic. So, I've got a simple diagram here showing a server hosted down in Dallas. This is my website. And then I have users all around the world. So, in Sidney I might have five.In London I've got five. New York I might have ten. LA I might have ten. I've got 30 users around the world that are accessing my server and my website down in Dallas. Let's kind of follow a set of these users in their journey. Let's look at their users down in Sydney. They make a request to the website. They've got an 8,600 mile hike to Dallas, and then an 8,600 mile hike back. The amount of time that that takes is usually measured and measured in milliseconds, and just that round-trip might be about 170 milliseconds. For our users up in London, that might be about 100 milliseconds. Our users in New York City can probably experience about a 40 millisecond round-trip time. And over in LA, about 30. So as you can see, the further you're away, the longer it takes ultimately, the slower the website will be for you. So this is where the the CDN comes into play, and this is how it actually accomplishes the increase in speed, which is by reducing the amount of distance between the user and the content, or the server providing the content. What it does by doing that is, it places these content delivery network endpoints in as many locations around the world as possible. And in our case, we're going to assume we've got one in just about every location where our users exist. So now when the user in Sydney, or London, or New York City, or LA tries to access some content, it's first retrieved by the content delivery network service and then distributed around the world. So we have a single request down to the Dallas server. It's now then distributed all around the world, and our users in London now instead of going all the way to Dallas, they're able to retrieve that content directly from their closest geographical location, drastically reducing the amount of time that it takes to retrieve that content. As you can see here, it's very basic how a CDN is able to provide the benefits to the end-user by reducing the amount of time that it takes to deliver the service. But what you're not seeing here, is an indirect benefit, is the reduction in the amount of traffic that actually hits the Dallas server. So the indirect benefit is that you actually see a reduction in the load, or a reduction in the amount of capacity that you need in Dallas, to serve all these users. So another indirect benefit because of there's this much less validity, and so much less stuff happening in Dallas, because all these users are having to make these trips. And I'm also not having to communicate with with users so far away. The Dallas environment may also see an increase in uptime. And then lastly, because the users are not really directly communicating with the servers down in Dallas, you have the indirect benefit of an increase in security through obscurity. It's pretty basic to understand how a CDN works in the end to provide a better benefit to the end user.

**Hybrid Multi-cloud**

Hybrid Cloud, as we covered in the previous lesson, is a computing environment that connects and organizations on-premise private cloud and third-party public cloud into a single infrastructure for running the organization's applications. Multicloud is a cloud adoption strategy that embraces a mix of cloud models from different service providers - public, private, and managed, across infrastructure, platform, or software services. For example, a business may consume email as a service from one provider, a CRM application from another, and infrastructure from yet another provider. So, essentially, a Hybrid Multi-cloud implies you're able to leverage the best of cloud models and services across different service providers, and have your applications and workloads working seamlessly across multiple different clouds. In this video we will look at some use cases for why a business may want to use a hybrid multi- cloud approach. In this video I want to touch on a few of those use cases for why a business may want to use hybrid or multicloud. Let's start with the basic one: cloud scaling. Now most of us are probably familiar with this, it's one of the main reasons for adopting the cloud. Now let's say we have a flower delivery service that is able to hit a certain bottom line of users to have on-premise infrastructure, and it can hit a certain amount of user load. So, visualizing this here throughout a calendar year, you can imagine that their load maybe goes up and down, and responds to specific holidays. Now to hit those peaks, they could scale up their on-premise architecture, but that's met with upfront costs and cost of upkeep. Now instead what they'll do, is take advantage of cloud that allows them to scale up in response to that load, and then automatically deprovision resources when they no longer need them. Now, this concept is kind of general to cloud computing, not just hybrid or multicloud. That brings me to my next topic. Here we're gonna be talking about how it can be used to build a composite cloud. Essentially this is going to be applications that are spread across multiple cloud environments. So back to the flower delivery service. Let's say they have on-premise architecture that allows them to run three major components of their app. So, let's say they have the web UI. They have some billing API's, as well as a rewards framework. Now let's say that this service is actually based in EU, and their European customers are happy. But for their North American or American customers, it's best specifically around, you know, Veterans Day or Thanksgiving, they're noticing that the system is bogging down. So they decide to take advantage of a hybrid cloud or multi- cloud architecture by composing their application across multiple cloud environments. So, they'll take advantage of data centers in America, and essentially, they've identified that although the rewards framework can stay on Prem in their European side, they want to move the billing and the UI capabilities over. So they'll move just those two to a cloud platform of their choice in a North American or American datacenter. This kinda allows them to scale up portions in response to say American holidays, while keeping their EU portions individually scaled. So in this example the flower delivery service is able to take advantage of scaling at a global level by using the hybrid or multicloud architecture. Next let's talk about the airline or travel industry. So we can first start with an example of modernization. In the past we've seen that reservation systems may have been difficult to work with, or you might have had to call in. But almost all the airline companies now have a mobile application. So most of the time, and we've actually found that, it's about - in general - not just in the travel industry, but 80% of all enterprise applications are actually still on Prem. And that's likely the case in this industry as well. So, in this specific example, let's say they have a reservation system that's running on prem. But to create new experiences for their end-users, let's say they've created a mobile application. That mobile app, of course, has a mobile backend that's maybe running in a public cloud. And that in turn works with the reservation service. So again, the mobile app can hit the mobile backend, that in turn works with the reservation capabilities. In this case, they've modernized and new user experiences are possible. But let's take that a step further. Now a source for a lot of dissatisfaction for users, is whenever their flights are delayed - so when a flight is delayed, they may have to rebook new flights. The solution is almost always the same. The traveler wants to get to his destination in the easiest way possible. What airline industries have been doing is taking advantage of the cloud to create maybe a recommendation feature. It allows them to book new flights as soon as the delay is recommended, or as soon as the delay is incurred and that's going to connect up to that mobile backend service, allowing users to be able to book flights through their phone the second of flight is delayed. This not only improves the bottom line for the airline industry, it leads to happier users. That's one way the modernization has been done. Next, let's take it even a step further and talk about data and AI. For data and AI, the airline industry has been taking advantage of lots of historical data. Over the decades that a company has been around, let's say they have historical data of when unplanned maintenance has happened on their airline. In fact, 30% of all delay time in the airline industry is actually when unplanned maintenance happens. So by taking advantage of, let's say, machine learning or AI capabilities, they could hook into all of the legacy data that they have - large volumes - and connect them up to machine learning and AI capabilities. This allows airline industries to take advantage of predictive analytics and get insights before errors, or before the unplanned maintenance ever occurs. This again improves their bottom line, leading to happier users and a more efficient airline industry. Today we talked about four major use cases for hybrid and multicloud platforms: cloud scaling and composite cloud, in the flower delivery service, as well as modernize in data and AI for the airline industry. Another reason for adopting hybrid multicloud strategy is to prevent lock-in to a specific vendor's cloud platform and having flexibility of being able to move workloads from one cloud platform to another as the need arises.

**Microservices**

Microservices architecture is an approach in which a single application is composed

of many loosely coupled and independently deployable, smaller components or services.

These services typically have their own stack running on their own containers.

They communicate with one another over a combination of APIs, event streaming, and message brokers.

For a business, what this means is that: Application components can be developed and

updated more efficiently by multiple developers working independently.

Teams can use different stacks and runtime environments for different components.

Components facing too much load can be scaled independently, reducing the waste and cost

associated with having to scale entire applications.

In this video, we’ll look at how microservices have shaped application development, and also

look at a use case that illustrates microservices in action.

The way developers work to build applications is changing.

In the past, software was built as large monolithic applications where a team of developers would

take months to construct a large application built on a common code base.

These developers would write every part of the application from start to finish.

Now, after decades of software development, there are vast amounts of code already out

there that developers can use as the base of an application, meaning they no longer

have to create every line of code from scratch.

Cloud development platforms provide developers with an ecosystem of code that can be easily

and securely integrated into applications.

Now, instead of building one huge application on one team, developers break into small independent

teams where they write smaller amounts of code called microservices.

Microservices breakdown large applications into their core functions, for example, search,

recommendations, customer ratings, or product catalogs.

Each is developed independently of one another, yet work together on the cloud development

platform to create a functioning application.

A container is the distribution method for each microservice, meaning it delivers the

code where it needs to go.

Containers are plug-and-play, so if one microservice isn’t working for an application, developers

can take it out and put in a different one without disrupting how the rest of the app

functions.

Check out microservices in action with Ron.

Ron is a soccer fan who uses an online streaming media service called Dream Game.

Last night he missed watching his team play their crucial semi-final match.

Luckily, he can watch the game tonight with Dream Game.

When he logs in, he sees the most popular content among all Dream Game users.

After some searching, he finds the match he’s looking for.

What he would really like is to find his game with one click.

Luckily, the Dream Game development team is using microservices to develop a better user

experience for viewers like Ron.

The first microservice is a content catalog housing the millions of games that Dream Game

offers.

The small team of developers organizes each piece of content with metadata that describes

them.

This metadata feeds into a second microservice, the search function, which ensures that Ron’s

search results are captured and compared to the Dream Game catalog.

The third microservice, recommendations, captures data about the most popular content among

all Dream Game users.

This is what generates the home page that Ron saw when he first logged in.

These three microservices are all in their separate containers ready to join the application.

But before they can work together, they have to find one another.

They do this by using something called service discovery, which creates a roadmap for these

and many other microservices to communicate.

When microservices find each other, they communicate using an application programming interface

or an API.

So when Ron searches for his favorite soccer team, the search microservice is communicating

to the content catalog, in an API, about what Ron is looking for.

Now back to the goal at hand.

To get Ron to a soccer game with just one click, the development team working on the

recommendations microservice is updating the code, adding an analytics algorithm.

Using analytics, the recommendations microservice will compare Ron’s viewing history and preferences

to popular content among other users, including soccer fans and viewers in Ron’s geographical

region and demographic.

Since the developers didn’t need to create the code from scratch, they are able to deploy

this new functionality in a matter of days.

These updates happen behind the scenes as the rest of the microservice containers function

normally.

The next time Ron checks DreamGame, instead of just seeing the most popular or newest

content he sees a personalized playlist that will continue to refine itself as the system

learns more about his viewing habits and preferences.

The result, Ron finds his favorite team’s latest game right away.

The microservice approach lets developers quickly innovate applications in parallel

and lets users like Ron focus on the things that really interest them.

And when those interests are changing and growing faster every day, microservices help

businesses keep up and grow with their customers.

**Serverless Computing**

Serverless is an approach to computing that offloads responsibility for common infrastructure

management tasks such as scaling, scheduling, patching, and provisioning application stacks

to cloud providers, allowing developers to focus their time and effort on the code and

business logic specific to their applications or process.

Serverless doesn’t mean there are no servers; only that the management of the underlying

physical or virtual servers is removed from their users.

The serverless computing environment allocates resources as needed for the applications.

Let’s look at some key attributes that distinguish serverless computing from other compute models.

The serverless model requires no provisioning of servers, installation of application stacks

and software, or operation of the infrastructure by the developer.

Serverless computing runs code only on-demand on a per-request basis, scaling transparently

with the number of requests being served.

Serverless enables end users to pay only for resources being used, never paying for idle

capacity, which is unlike virtual servers on the cloud—where end users pay for VMs

as long as they are running even if idle.

Effectively, serverless abstracts the infrastructure away from developers.

Code is executed as individual functions where each function runs inside a stateless container.

No prior execution context is required to serve a request; and with each new request,

a new instance of the function is invoked.

Let’s look at a scenario.

You could, for example, have a serverless platform between the front-end of your website

and your storage layer, running individual functions.

The serverless app could be translating text files and storing it in a cloud-based storage

service.

Using the front-end of your website, you send text files to a serverless app. The app creates

translations in different languages, and then stores these translated files in cloud storage,

and sends their links back to you.

Some of the key serverless computing services today include IBM Cloud Functions (which is

based on Apache OpenWhisk), AWS Lamb-da, and Microsoft Azure Functions.

It is important to note that serverless may not be the best fit for all applications or

scenarios.

You need to evaluate application characteristics and ensure that the application is aligned

to serverless architecture patterns.

Applications that qualify for a serverless architecture include some of the following

characteristics: Short-running stateless functions (seconds

or minutes).

Seasonal workloads with varying off-peak and peaks.

Production volumetric data that shows too much idle time.

Event-based processing or asynchronous request processing for implementing use cases.

Microservices that can be built as functions that are stateless.

Serverless architectures are well-suited for use cases around data and event processing,

IoT, microservices, and mobile backends.

Given its inherent and automatic scaling, rapid provisioning, and a pricing model that

does not charge for idle time, supporting microservices architecture has become one

of the most common use cases of serverless computing today.

Serverless is well-suited to working with structured text, audio, image, and video data

around tasks such as data enrichment, transformation, validation and cleansing, PDF processing,

audio normalization, thumbnail generation, and video transcoding.

Parallel tasks such as data search and processing, and genome processing, are also well-suited

to be run on a serverless runtime.

Serverless is also well-suited for working with all sorts of data stream ingestions,

including business data streams, IoT sensor data, log data, and financial market data.

And finally, let’s look at some challenges worth considering about serverless.

Serverless workloads are designed to scale up and down in response to workload, but for

workloads characterized by long-running processes managing a traditional server environment

might be simpler and more cost-effective.

The serverless application architecture can be vendor dependent, and so there is a potential

for vendor lock-in, particularly involving platform capabilities such as authentication,

scaling, monitoring, or configuration management.

Because serverless architectures scale up and down in response to workload, they also

sometimes need to start up from zero to serve a new request.

For certain applications, this delay isn’t much of an impact, but for something like

a low-latency financial application, this delay wouldn’t be acceptable.

**Cloud Native Applications**

Simply put, a cloud native application is an application developed from the outset to

work only in the cloud environment, or an existing app that has been refactored and

reconfigured with cloud native principles.

A cloud native application consists of microservices working together as a whole to comprise an

application, yet each can be independently scaled and iterated through automation and

orchestration processes.

These microservices are often packaged in containers, which are executable units of

software in which the application code is packaged along with its libraries and dependencies

so that it can be run anywhere.

This independence enables frequent, iterative improvement of cloud native applications,

without disrupting the experience of end-users.

Cloud native applications are unlike traditional, or monolithic applications, that are built

out of one huge piece of software; applications that tightly couple the user interface, business-logic

layer, and data-layer.

Let’s take the example of how a cloud native application might be used on a travel website.

Each topic covered by the site—flights, hotels, cars, specials—is its own microservice.

Each microservice may roll out new features independent of the other microservices.

Specials and discounts can also scale out independently.

While the travel site is presented to customers as a whole, each microservice remains independent

and can be scaled or updated as needed without affecting other services.

Whether creating a new cloud native application or modernizing an existing application, developers

adhere to a consistent set of development principles:

Follow the microservices architectural approach by breaking applications down to single-function

microservices.

Rely on containers for maximum flexibility, scalability, and portability.

Adopt Agile methods that speed the creation and improvement process through quick iterative

updates based on user feedback.

In this video, we’ll take a closer look at the key concepts of cloud native, its benefits,

and use cases.

Hi.

I'm Andrea Crawford and I'm with IBM Cloud.

Today we're going to talk about cloud native apps.

In the heritage world, we have our lumpy, monolithic apps.

And in the new world, we have our microservices living on the cloud.

If we take a look at this diagram here, we see we have cloud infrastructure.

This is your private, your public, and your enterprise infrastructure.

Cloud native apps apply to hybrid and multicloud situations.

We also have our scheduling and orchestration layer.

This layer is all about control planes, like our kubernetes.

We also have our application and data services layer.

This layer is all about backing services, and being able to integrate our application

code with existing services that may be available on other clouds, or even on-premise.

We have our application runtimes, these are what we're traditionally, or conventionally,

known as middleware.

And over here, well, that's where we have our

cloud native apps.

This is the sweet spot right up here.

So our application code is actually designed, built, and delivered very differently for

cloud native, than it would be for conventional, monolithic, lumpy apps over here.

Let's talk a little bit about why cloud native apps

can actually leverage benefits like: enabling innovation,

business agility, and most importantly - from a commoditization of this solution stack over

here.

So as time has progressed and technologies have matured and emerged, a lot of the services

are actually being refactored lower down in this stack.

This means that core services are starting to have a lower center of gravity, freeing

up innovation at this level over here.

So, what are our use cases for when to build a cloud native app?

\*Everything\*

Everything that lives in the cloud should have a cloud native app design and approach.

This means our application code needs to be instrumented with things like: standardized

logging, standardized events, and being able to match those logging and events to a

standard catalog, that multiple microservices and cloud native apps can use.

The last thing we want to do is have our development squads have to figure out what their log and

event messages should be.

Let's standardize that, because we want to be able to commoditize that as well.

We also need to have things like distributed tracing.

When we get over into the microservices world over here, we

have a lot of moving parts.

This means we're going to need to leverage services core to the system, like: load balancing,

service discovery, and routing.

These are the kinds of things that are commoditized in this layer here, with

things like Istio, and with the emergence of newer projects, like Knative.

And so, if we were to recognize the benefits for cloud native apps and to sum it all up,

we are all about enterprise and engineering at scale.

**DevOps on the Cloud**

Development teams need to design, develop, deliver and run software as reliably and efficiently

as possible.

Operations teams need to identify and resolve problems as soon as possible by monitoring,

predicting failure, managing the environment, and fixing issues.

Combining development and operations with the ability to monitor and analyze and optimize

bottlenecks gives us DevOps, a collaborative approach where business owners and the development,

operations, and quality assurance teams collaborate to continuously deliver software.

A DevOps approach applies agile and lean thinking principles to all stakeholders in an organization

who develop, operate, or benefit from the business’s software systems, including customers,

suppliers, partners.

By extending lean principles across the software supply chain, DevOps capabilities improve

productivity through accelerated customer feedback cycles, unified measurements and

collaboration across an enterprise, and reduced overhead, duplication, and rework.

Using the DevOps approach, developers can produce software in short iterations

on a continuous delivery schedule of new features and bug fixes in rapid cycles;

and businesses can seize market opportunities and reduce time to include customer feedback

in their products.

The DevOps process involves: Continuous Delivery, which is about delivering

small, well-designed, high-quality, increments of software to customers.

Continuous Integration; creating packaged builds of the code changes released as immutable

images; where immutable implies that when modifications are needed, the entire component

is replaced with an upgraded version.

Continuous Deployment, which involves progressing each new packaged build through the deployment

lifecycle as rapidly as possible.

Continuous Monitoring; with tools that help developers understand the performance and

availability of their applications, even before they are deployed to production.

Delivery Pipeline; which is an automated sequence of steps that involves the stages of

Ideation, Coding, Building, Deploying, Managing, and Continuous Improvement; which loops back

to the Ideation phase in the delivery pipeline.

While DevOps can apply to applications anywhere, there is especially a compelling case for

DevOps when it comes to cloud-ready, and cloud-native applications.

DevOps and Cloud share a symbiotic relationship.

With its near limitless compute power and available data and application services, cloud

computing platforms come with their own risks and challenges.

DevOps’ tools, practices, and processes are helping tackle some of the complexities

and challenges posed by the cloud and allowing solutions to be delivered—quickly and reliably.

Let’s look at some core capabilities that DevOps provides to help building and running

applications in the cloud a lot more manageable: DevOps best practices make it possible to

programmatically provision servers, build middleware, install application code, and

fully automate the installation process in a way that is documented, repeatable, verifiable,

and traceable.

Application deployments often involve considerable complexity.

The DevOps’ practices of continuous integration and continuous deployment help create a fully

automated deployment pipeline, which is important all through the application development lifecycle.

Cloud native applications form a complex distributed system with multiple moving parts, independent

tech stacks, and rapid release cycles.

DevOps principles are essential to define how people work together to build, deploy,

and manage applications in a cloud native approach.

With the DevOps best practices of automated provisioning and continuous deployment, developers,

quality professionals, and other stakeholders can test in low-cost, production-like test

environments that were previously not available—enhancing both productivity and quality.

When systems are compromised or struggling to recover from natural disasters, DevOps

best practices make it possible to rebuild these systems quickly and reliably.

DevOps provides a powerful set of principles, practices, and tools to realize the full potential

of cloud-native computing, as well as for modernizing existing applications to leverage

cloud benefits.

**Application Modernization**

Many organizations have huge investments in existing applications that are often siloed

in legacy systems and are very difficult and expensive to update and maintain.

Modernizing these applications can unlock great benefits for these organizations such

as accelerating their digital transformations, enabling them to take advantage of new technologies

and services, and becoming more responsive to their customers' needs and changing market

dynamics.

Cloud computing is one of the three main ingredients in Application Modernization.

In the rest of this video we will see how, and what else Application Modernization entails.

Hi, I'm Eric Minick, with IBM Cloud, and I want to talk about application modernization

and three huge transformations that have been going on together.

We've got three things going on.

They're interrelated, and this is what we're seeing this change in how we're doing architecture,

infrastructure, and our ways of working - how we deliver.

And if we go back in time a little bit, we saw applications that were very monolithic,

they were running on physical servers and we used waterfall style development where

we'd have long plans and we'd say okay, this is gonna be our planning phase, or development

phase, or testing phase, and we could plan out a year as a project.

And that's really what we've gone away from.

So if we look at how most organizations are working today, architecturally, they've got

some sort of distributed architecture.

It's usually related to like a service-oriented architecture; the the big buzzwords a few

years ago.

But some sort of distributed architecture.

We have a bunch of web services.

They're talking to each other.

We've got some databases on the backend.

And then some front ends that kind of go through all that.

On an infrastructure level, they're running on some sort of a virtual machine, alright.

So, we said we could probably do better than having to order a new server every time we

have a new service.

Let's virtualize this stuff, and we need a little more density along the way.

And from a way of working, you know, Agile development - pretty normal - and then trying

to figure out a little bit of what happens downstream.

So this kind of takes us up to where a lot of teams are today, but not really where they're

going.

And so if we look at, kind of, that next phase.

We're taking another pass at this service-oriented architecture, and really shrinking the sizes

of the services, taking advantage of the more dynamic infrastructure we have.

And we're calling these now microservices.

So we've got a microservice architecture, so, very small, very focused services, moving

away from a lot of the heavyweight XML based communication we saw in SOA, towards more

rest based communication, things like that.

But same idea, let's keep breaking into smaller and smaller pieces.

We have more independence of what we ship.

More rigor in saying this service needs to be independent from another service, so I

can change these things by themselves.

On the infrastructure side - cloud.

It's pretty popular.

And this could be public cloud, this could also be private cloud.

I'm painting with a very broad brush when I say cloud here.

And thhen from a delivery, in a way of working, we could say that DevOps is really key.

And I would include in this, approaches like site reliability engineering, SRE.

More the ways of working we have today.

Now that that's fine and interesting.

But what do these things have to do with each other?

I'd argue that what we're really seeing is modernization in how the applications are

delivered, and how they're built, and what they are.

And while you could walk into any large enterprise today and you'll find someone who says we

are going through a cloud transformation.

You'll often find somebody says, "yes, I'm in charge of leading the DevOps transformation."

And you'll walk into enterprise architecture and they'll say, "yes, we are pushing microservice

architectures."

Individuals think that they're going through three separate transformations.

But they're really tied.

Right?

If I'm doing microservices, and I have new microservices all the time.

In order to get a new microservice up and running, I'm over here and I have to order

a new physical server, then rack and stack it a couple months later.

I'm not gonna get any time to market benefits.

The resilience benefits that I'm gonna normally look for from microservices are gonna be modest,

at best.

Microservices want cloud infrastructure.

You want to be able to say, I've got a new microservice.

Let me put it in a container and just run that container right now and scale that dynamically.

Similarly, cloud really likes running microservices.

The benefits of being able to dynamically scale are really cool when you have a lot

of small things that you might need few of, or a lot of.

It's not as interesting when I've got a monolith that that isn't even distributed.

How do I scale that?

I get a bigger cloud server?

And then all of this is, kind of, baking in this idea of speed and resiliency

and DevOps brings that together. The developers who have always wanted speed, the operations

people who've always wanted that resiliency.

They're going to be programming that cloud. Right?

The programmable infrastructure the cloud provides, needs operations people who understand

resiliency, but bring some of that development skill in.

And to really take advantage of these new infrastructures, the new architectures, you need these new

ways of working.

And you also are gonna say, if this is going to give me time to market benefits, I can't

be back here, and say we've got a one-year project plan that we're just gonna execute.

I need to be able to be more agile, and adapt in my planning and my responsiveness to the

business.

I need to better wire up my application so they can be more easily monitored and more

resilient.

We have to have the application in a way that it knows when one of these services is failing,

and we can spin up another.

So this is really, for me, fascinating that you walk into these organizations everywhere,

and they're undergoing these three different transformations.

But they're always doing them together.

And when they don't - it doesn't quite work.

So you've got these three transformations going on at once.

And you'll hear us talk a lot about application modernization.

See it written across the top here.

When I think of application modernization, I think it's just this.

It's this transformation right here; going from these kind of monoliths, or service-oriented

architectures, to micro-services, adopting cloud, modernizing our ways working towards

DevOps and SRE. That's AppMod.

It's a really exciting time.

And it's really great when you're able to go after it in a holistic fashion.

**What is Cloud Security - Part 1**

Hi, I'm Nataraj Nagaratnam, and I'm from IBM Cloud.

Traditionally when you deploy an application, you have the entire data center, the servers

that you run. You're responsible for all of it.

In the cloud model, that's a shared responsibility between you and the cloud provider.

In a shared responsibility model, you need to rethink security; on what your responsibility

is, and what cloud provider's responsibility is.

Let's take platform as a service as an example.

When you look at PaaS, you're building applications, migrating data to the cloud, and building

applications, running them on the cloud.

So you're responsible for securing the applications, the workload, and the data, while the cloud

provider is responsible for managing the security of the platform, so that it's compliant,

it's secured from the perspective of network, the platform on down, in terms of managing

the containers, the runtime, and isolation so that you have your own space within the

platform.

Whereas, if you are adopting and migrating workloads to the cloud, and you're using infrastructure

as a service, then the cloud provider manages hypervisor on down

if you are using virtual servers or if you are using bare metal, then you can completely

control everything on up, from the operating system, the virtual servers that you run,

and the data you bring it on.

So it's very important to understand the adoption model, whether you're consuming Iaas, or PaaS,

or if you're consuming SaaS - where the club provider manages all the applications, and

the security of it and

you worry about the data that you bring in, and plan accordingly.

So that's a very important thing, because it's part of understanding your responsibility

in ultimately managing the risk and compliance of the workloads and the data that you bring

to cloud.

Now let's talk about architecture.

When you build applications and migrate applications, or modernize your apps - let's start with

data with all the risk that you deal with

and the kind of data matters.

Is it confidential data?

Is it public data or sensitive data, that may deal with private information?

Consider all those factors and make a secure design around what your data security architecture

should be.

Make sure you have data at rest encryption so that the data is always encrypted, whether

you use a database as a service, object store as a service, or other ways to store data

like block storage.

Encryption is for amateurs.

If you think about key management, is for professionals.

So, having more control of your keys, provide you the ability, in the context of shared

responsibility model, that you own your data.

You have complete control of your data.

So as you think about key management, make sure you have an approach to think about if you

are bringing confidential data, you want to bring your own keys.

Maybe sensitive data, you want to keep your own keys, so that how much control of the

keys you have and the hardware security module in which the key processing the encryption/decryption

operations happen.

More control you have, more responsibility that you can take on.

So encryption at data at rest, data in motion - as it comes from services to data stores

or applications, so that as you think about data coming all the way.

Your request and API requests coming all the way - data in motion.

And in the new world you need to start thinking about when the application is actually processing

the data, that is going to be data in this memory.

So you can actually start to protect data using hardware based technologies where you

can protect in-memory data as well.

So that when it is in use, and in memory by the applications, you can protect it.

Take a holistic approach to data protection at rest, in motion, in use, with full control

of your keys.

It can be bring your own keys, or even better, push the boundary with keep your own keys.

The application that serves the data - it's not only about which application needs to

have access.

Make sure the data access is only on need by need basis.

Do not open up your data services to the whole world, be it network access, or everybody

to access the data.

Make sure you exactly know which applications need to access, or which users need to access

the data to run your cloud applications.

From an application viewpoint, make sure there are no vulnerabilities in your application.

So scan your applications.

Have an AppSec, application security, approach so that you can do dynamic scanning or static

scanning of your application before you deploy it into the production.

In the cloud native environment, you are deploying container images.

So you can scan your images.

You can scan it for vulnerabilities before you deploy, and set your policies so that

you only have secured images in production any time.

If there is any vulnerability in the new world, you don't need to patch these systems, you

just spin up a new container and off you go.

That's the beauty of a cloud native approach; that you have security built in in every step.

So at a container level and the applications that serves the business logic, you can start

to protect them.

Then when you look at the users coming in, you want to manage access in terms of who

the user is, and from where they are coming from.

So, identity - you need to make sure who the user is, or which service it is, based on

the identity of those services or users, so that you can maintain access control to your

application or data.

And also from the perspective of network access, you want to make sure only authorized users

can get in and

if there are intruders out there, you can make sure you set it up so that they are prevented

from accessing your application and your data in the cloud, be it through web application

firewalling, network access control, or distributed denial of service protection, and have intelligence

built into these network protections as well.

So both identity and network - in essence, you are protecting your data.

You need to manage access to your apps and the workload on the data that you have deployed

on the cloud.

**What is Cloud Security - Part 2**

You need to have a continuous security monitoring so that you know at any point whether you're compliant, your policies, you can watch out for threats that you need to manage. Having an approach and set of tools to manage security and compliance posture is very important. Gaining insights about your posture, compliance, and threats. So from your deployment environment you can garner information. It can be security events, audit logs, flow logs from network or system that can be fed in so that you can figure out what your posture, and complaints, and threats are. And not only is it important for you to gain insight, you need to have actionable intelligence so that you can start to mediate. You may figure out there's a vulnerability; a container image that you have deployed is vulnerable, so you can remediate and respin up a new container. There may be a particular access from a network that seems to be coming in from a suspicious network IP address. You can block that. So ability to gain visibility and insights, and having that insight and turning it into actionable intelligence and remediating is very important. Let's talk about DevOps. DevOps is about development and operations. Traditionally we think about, okay, there's an application team that is doing the design and architecture for building code, and then you throw it over the wall for the enterprise security team to secure it and manage it. That should be rethought. Fundamentally, it's not just about dev and ops, but security needs to be a forethought, not an afterthought. So it should become SecDevOps approach to the way you build, manage, and run your applications. You need to embed security into the entire lifecycle, what we call shift left. Not only will you manage security, but shift left through the entire process. You need to have a secure design. So as you plan, as you design and say: what kind of data am I going to put what level of classification? What kind of applications am I building? Is it container-based? Is it a workload that I'm migrating? Take that into account along with what integrations you need to do so that you can plan it and architect it. Then as you build it, embed security as part of that process. Do you have security aware applications? For example, you may want to encrypt data, or the sensitive data. You may want to encrypt the data from your applications before you even store it into a data store. So, secure build - and you manage security. As part of SecDevOps, as you have secure design and architecture, you pass on that and build secure applications and deploy and manage security in continuous fashion. Then you have a closed loop so that whatever you find, you may need to remediate, or rearchitect your application, or implement certain things as the threats landscape evolves.

**Identity and Access Management**

According to the 2019 Cloud Security Report by Cybersecurity Insiders, the top cloud security

concern of cybersecurity professionals is data loss and leakage.

Unauthorized access through misuse of employee credentials and improper access controls is

the single biggest perceived vulnerability to cloud security, followed by insecure interfaces

and APIs.

In this video, we will look at how Identity and Access Management, also known as access

control, works as the first line of defense, allowing you to authenticate and authorize

users and provide user-specific access to cloud resources, services, and applications.

A comprehensive security strategy needs to encompass the security needs of a wide audience including

organizational users, internet and social-based users, third-party business partner organizations

and vendors.

There are three main types of users – Administrative Users, Developer Users, and Application Users.

Administrative users include cloud platform administrators, operators, and managers—roles

that typically create, update, and delete application and service instances, and also

need insight into their team members’ activities.

An attacker on an administrative account can steal data from production database service

instances, deploy malicious applications inside the customer's domain, or even deface or destroy

existing applications.

Developer users include cloud application developers, platform developers, and application

publishers.

Developer users are authorized to read sensitive information and to create, update, and delete

applications.

The third type of user is the Application user.

These are the users of the cloud-hosted applications.

Let’s look at the key components of identity and access management and how they work.

Authentication, or the identity service, enables applications deployed to the cloud to authenticate

users at an application level, based on a range of identity providers such as the cloud

directory, social identity providers such as Google, LinkedIn, Facebook, and Twitter,

enterprise hosted identity provider, and cloud hosted identity provider.

Sometimes API keys, or unique identifiers are passed into an API to identify the calling

application or user.

Multifactor authentication is used to combat identity theft by adding an additional layer

of authentication for application users, such as single-use passwords or pins, certificates,

tokens, risk-based authentication, such as changes in the user’s location, past activity,

and preferences.

Cloud Directory services are used to securely manage user profiles and their associated

credentials and password policy inside a cloud environment.

A directory service within a cloud means that applications hosted on the cloud do not need

to use their own user repository.

Reporting helps provide a user-centric view of access to resources or a resource-centric

view of access by users.

Reports typically give information about which users have access to which resources, which

users have changes in access rights, which access is being exploited by each user, and

under which conditions.

Audit and compliance is a critical service within identity and access management framework,

both for cloud provider, and cloud consumer.

Auditors use these processes to validate implemented controls against an organization's security

policy, industry compliance, and risk policies and to report deviations.

User and service access management capability enables cloud application and service owners

to provision and de-provision customer, partner, and vendor user profiles with minimal human

interaction.

This streamlines access control based on the role, organization, and access policies defined

by the owner.

User accounts of administrators or developers give access to sensitive information.

In order to mitigate the risks of these accounts being hacked into, you require maximum control

over the whole life cycle of these users.

Some of the controls that can help secure these sensitive accounts include:

Provisioning users by specifying roles on resources for each user.

Password policies that control the usage of special characters, minimum password lengths,

and other similar settings. Multifactor authentication like time-based

one-time passwords; and Immediate de-provisioning of access when users leave or change roles.

Cloud providers offer Identity Access and Management services, typically including

the ability to create access groups, add users to access groups, and manage access for existing

users.

An access group is a group of users and service IDs created so that the same access can be

assigned to all entities within the group with one or more access policies.

Access policies define how users, service IDs, and access groups in the account are

given permission to access account resources.

Policies include: a subject which can be users, service IDs, or access groups; a target—which

is the resource, or provisioned service offering, to which you want to provide access and role—which

defines the actions allowed on the target of the policy, that is, the resource to which

the access is being granted.

Access groups provide a more streamlined access assignment process as compared to assigning

individual access to each user and help reduce the number of policies in an account.

In this video, we learned how Identity and Access Management work as the first line of

defense to secure the cloud.

**Cloud Encryption**

Given the concerns around data security and privacy, especially in public cloud environments,

encryption plays a key role, and is often referred to as the last line of defense, in

a layered security model.

This protection not only encrypts data, but also provides robust data access control,

key management, and certificate management.

In this video, we will take a closer look at cloud encryption.

Encryption is defined as scrambling data in a way that makes it illegible.

There are two parts to an encryption system—the encryption algorithm and the decryption key.

The encryption algorithm defines the rules by which data will be transformed so that

it becomes illegible and the decryption key defines how the encrypted

data will be transformed back to legible data.

Encryption ensures that only authorized users have access to sensitive data, and when accessed

or intercepted without authorization, data is unreadable and meaningless.

Cloud providers offer various cloud encryption services.

This could be limited encryption of data that is identified as sensitive or end-to-end encryption

of all data uploaded to the cloud.

Data is encrypted upon receipt, and encryption keys are passed to the customers to decrypt

data when needed.

Keys need to be managed securely.

If you lose your keys, you will not be able to read your data.

Data needs protection in three states—at rest, in transit, and when it is in use.

Encryption at rest protects data while it is physically stored in a database or the

storage layer.

Depending on the application and business requirements, there could be multiple options

for encrypting data at rest, such as encryption for block and file storage, built-in encryption

in object storage, and database encryption services.

Encryption in transit protects data while it is transmitted from one location to another.

Encryption in transit includes encrypting the data before transmission, authenticating

endpoints, and decrypting and verifying data on arrival.

Secure Sockets Layer (or SSL) and Transport Layer Security (TLS) are commonly used protocols

for encryption in transit.

They are not only used when accessing websites securely but also for data moving between

servers and services within the cloud.

Encryption in use protects data when it is in use in memory for computations.

It allows computations to be performed on encrypted text without needing to decrypt

the data.

Cloud storage encryption could be server-side or client-side.

Server-side encryption occurs after cloud storage receives your data, but before the

data is written to disk and stored.

For server-side encryption, you can either: Create and manage your own encryption keys,

known as Customer-supplied encryption keys; or you can generate and manage your encryption

keys using key management services offered by the cloud storage provider, known as Customer-managed

encryption keys.

Client-side encryption occurs before data is sent to cloud storage.

This way, users can utilize encryption keys and algorithms that are not visible to the

cloud provider, making it virtually impossible for cloud providers to decrypt hosted data.

Given that a majority of enterprises today operate in multi-cloud environments, there

is a need to implement a singular data protection strategy across an enterprise on-premise,

hybrid, and multi-cloud deployments.

Some cloud providers offer multi-cloud data encryption services with a range of features

such as data access management, integrated key management, and sophisticated encryption

that combine to deliver the scalability and flexibility to help protect the most sensitive

workloads across the enterprise, regardless of where the data resides.

Using a multi-cloud data encryption console, you can define and manage access policies,

create, rotate, and manage encryption keys, and aggregate access logs.

Encryption does not eliminate data security risk—it separates the security risk from

the data itself by moving security to the encryption keys.

These keys need to be managed and protected against threats in order to keep the data

secure.

Key Management Services offered by some cloud providers help perform life cycle management

for encryption keys that are used in cloud services or customer-built applications.

They enable customers to encrypt sensitive data at rest and to easily create and manage

the entire lifecycle of cryptographic keys that are used to encrypt data.

Since the keys remain in possession of the customer, the data is protected from cloud

service providers as well as from other users.

Some of the best practices for encryption key management include:

Storing encryption keys separately from the encrypted data.

Taking key backups offsite and auditing them regularly.

Refreshing the keys periodically.

Implementing multi-factor authentication for both the master and recovery keys.

**Cloud Monitoring Basics and Benefits**

Cloud-based deployments can be complex.

Monitoring performance across an entire stack of applications and services can be time-consuming

and draining on internal resources.

This is where cloud monitoring solutions come in.

Cloud monitoring solutions assess data, application, and infrastructure behaviors for: performance,

resource allocation, network availability, compliance, and security risks and threats.

Cloud Monitoring is not just about automated tools. It includes the strategies, practices,

and processes that need to be in place for analyzing, tracking, and managing cloud-based

services and applications.

It also serves to provide actionable insights that can help improve availability and user

experience.

Cloud monitoring helps to: Accelerate the diagnosis and resolution of

performance incidents.

Control the cost of your monitoring infrastructure.

Mitigate the impact of abnormal situations with proactive notifications.

Get critical Kubernetes and container insights for dynamic microservice monitoring.

Troubleshoot your applications and infrastructure.

Cloud monitoring solutions are designed to give organizations visibility and control

over their entire cloud-based infrastructure.

They provide: Data in real-time with round the clock monitoring of virtual machines,

services, databases, and applications.

Multilayer visibility into application, user, and file access behavior across all cloud-based

applications and services.

Advanced reporting and auditing capabilities for ensuring regulatory standards are being

met.

Large-scale performance monitoring integrations across multicloud and hybrid cloud environments.

One way to categorize cloud monitoring tools solutions is to break them down into Infrastructure,

Database, and Application Performance monitoring.

Infrastructure monitoring tools help identify minor and large-scale hardware failures and

security gaps so that developers and administrators can take corrective action before problems

affect user experience.

Database monitoring tools help track processes, queries, and availability of services to ensure

the accuracy and reliability of database management systems.

Application Performance Monitoring, or APM, measures application availability and performance,

providing tools needed to troubleshoot issues in an application's environment.

APM solutions help improve user experience, meet application and user service level agreements

(SLAs), minimize downtime, and lower overall operational costs.

To get the most benefit from your cloud-based deployments, you can follow some standard

cloud monitoring best practices.

Leverage end-user experience monitoring solutions to capture the performance of an application

from the point of view of its end users.

These solutions monitor user journeys to track parameters such as application response time

and frequency of use under varied conditions.

These insights will help you to improve customer experience significantly.

Consider moving all aspects of your infrastructure, whether in private, public or hybrid clouds,

under one unified monitoring platform.

This can help you to manage all your KPIs in one place with complete visibility into

performance optimization.

Use monitoring tools that can help you track the usage and cost of your cloud resources

and services.

Increase cloud monitoring automation.

This can help you gain significant operational efficiencies.

Simulate outages and breach scenarios to evaluate how well your monitoring tools capture and

alert against failures.

Cloud monitoring needs to be a priority for organizations looking to leverage the benefits

of cloud technologies.

It will help you manage and optimize your cloud resources for cost and performance,

and create better customer experiences.

**Case Studies in Different Industry Verticals**

In this video, we will look at some case studies that demonstrate the use of cloud computing in real business scenarios, and the ensuing impact on these businesses and the customers they serve. The Weather Company's mission is to map the atmosphere of the earth. And based on that, generate the most accurate and hyperlocal forecasts, which can be served out to all our consumers and devices - which number in millions across the globe. Weather is unpredictable. We need to be able to spin up and spin down as fast as weather happens. Our normal day-to-day load is 30 million unique users and when we peak into severe weather, that can go north of 100 million across our products. We generate forecasts on demand. And our forecast on demand system is responsible for generating forecasts at a resolution of one square kilometer grade. We deliver forecasts on demand to the tune of 250 billion per day. On the backside of that is our API platform and that system operates at the scale about 150,000 requests per second. If our products don't work, and don't work fast, people's lives are in danger. Imagine a hurricane is approaching the East coast of the United States. As it is approaching the coast, usage of my weather data goes up tremendously. It can go up from anywhere from two times, to five times. It took us six months to migrate to IBM Kubernetes from our previous provider. What we've noticed is some real efficiencies with DevOps. It's reduced our workflow in pipeline by about 80%. With the latest hurricane, we were able to scale with IBM cloud very easily, very seamlessly. IBM's community service allows us to scale as fast as the weather happens. As we migrated our web platform to IBM Cloud Kubernetes services, one of the biggest advantages is that this is a managed service... which allows our team not to have to babysit our system; enables them to do other work. With the migration to IBM Cloud Kubernetes, we've gained the ability to have automation security baked in. This was a feature in the component that was very main for us in the past. IBM's security team proactively now notifies us of any security vulnerabilities. The tools and the investment that the company's made, gives us capabilities that we would have never had before. It gets us very excited and it gives our developers and engineers opportunities to build new things that they wouldn't have otherwise been able to do. The global reach that IBM and IBM Cloud together with our technology, with our capabilities for forecasting, give us the ability to extend our products and services to keep people safe all over the world. [Music] [In-flight announcement: "Good afternoon ladies and gentlemen. Welcome aboard. Captain speaking. Anticipating on-time arrival.] In the scenario where there's a flight cancellation or some other off-scheduled operation, we've had a system where we find new seats and flights availability for those customers. But I think oftentimes customers didn't necessarily know that that was the best option for them. What we wanted to do was create a system where they could actually see alternate options. The goal was to facilitate a better experience for our customers on how to get to their destination, in an automated fashion, on the channel of their choice. In our business, hurricanes, storms, are the natural events that impact our customers on a fairly regular basis. What happened - and this is a true story. The hurricanes hit and nobody said this is ridiculous. Our customers were being impacted by the hurricanes immediately. Why are we mitigating risk of a traditional rollout, when we have the power of the Cloud? This is exactly the scenario that our customers need this capability for. I think people will now realize that these kind of activities and integration are much easier when we can use cloud technology, and especially microservices. We can break problems into much smaller problems. Those get much easier to wrap your head around and develop. This is the technology that we need to be adopting now in order to keep pace with our customer expectations. What I loved about my job at Cementos Pacasmayo, is that it challenged me every single day. Technology moves so agile that I have to keep up with it. And what I love is that the company is also on that same road, driving towards it's digital transformations. Our customers are now demanding quicker time to market and also a more broad portfolio of products. We need to keep up and deliver the best service and the best products that they can get, in the least amount of time. We implemented SAP S/4HANA on IBM Cloud since it brings us a cost-effective infrastructure and also, it's quite scalable. Cementos Pacasmayo is now turning from a product-driven company, to a service-driven company. The accounting area now has real insights in real time on our financial statements that they didn't used to have and also on our supply chain area, especially on the procurement. They now have the dashboard that helps them take decisions right on time. [Music] My name is Mukesh Sharma, and I'm the Senior Manager of IT at Welch Food. Welch started as an organization 150 years ago, and pretty fast became the national. We are owned by the farmers. It's a co-op. And we value the sweat equity our farmers put into it. So, that's what drives us every day, that our farmers are working out on the farms. And we were working here to make sure the organization gets the best benefit of each dollar spent back to them. Welch's IT is the heart of the business - all the manufacturing systems, any manufacturing data which comes in through different processing ERP systems, you name it. We started this journey with the private cloud. Then at the same time, we have started the process of asking ourselves on any new application request, can it be run in the public cloud? The approach we have taken is to slowly and steadily start to move these non mission-critical systems, which can be done better by somebody else, let them run outside. Let them spend time on it, while we spend time on our core values. Liquid Power is a company that sells products that when you inject them into crude oil or gasoline pipelines, it changes the flow characteristics to such a point that you can either one, increase the flow of those fluids, or secondly, use less energy to actually push the fluids down the pipeline. We needed to become a standalone company under Berkshire Hathaway, without any experience in running our own infrastructure, or SAP. We had to come up with the solution that was best for our business to operate as a standalone company. There were many decisions we had to make. Do we go cloud? Do we stay on-premise? How do we create our own infrastructure and back office? And what's the best decision for us not only today and now, but in the future going forward? So, I talked to some CIOs and IT professionals. If you had a blank sheet of paper, what would you do? Cloud or go on-premise? Without a doubt, all of them say cloud. This whole process and the migration to the cloud, is giving what we believe is a competitive advantage. And what I really love about having IBM with SAP on the cloud, is it's scalable. We can do it much faster, and change in a much faster time period than we could otherwise.

**Career Opportunities and Job Roles in Cloud Computing**

Cloud computing is a key part of an enterprise’s digital transformation strategy.

As more and more companies are moving critical business processes and applications to a mix

of cloud infrastructures, qualified cloud computing professionals are in high demand.

According to Gartner’s report on the Cloud Services Industry, from a market size of 182.4

Billion USD in 2018 to a projected market size of 331.2 Billion USD in 2022, the market

size and growth of the cloud services industry is at nearly three times the growth of overall

IT services.

That is the scale at which the cloud market is growing.

Employer demand is outpacing the number of qualified candidates available.

Gartner TalentNeuron’s database of more than one billion unique job listings, scores

the hiring scale for jobs requiring cloud computing skills at 78, which means employers

are finding it "difficult" to get the right applicants for open positions in cloud technologies.

There are many specialization areas within this domain.

Here’s a look at some of the common roles available currently:

Cloud Developers, or Cloud Software Engineers, work through all phases of the software development

lifecycle: writing, testing, and maintaining the code.

They work with the front- and back-end of applications, as well as platforms and systems

that their applications run on.

Cloud Developers need to have a mix of technical skills, business knowledge, and experience

with at least one of the major cloud providers.

Technical skill for a Cloud Developer would typically include:

Knowledge of data structures, distributed systems, operating systems, and algorithms.

Experience with databases.

Proficiency in commonly used web application development languages, such as Python, JavaScript,

Java, HTML, and CSS.

Cloud Integration Specialists are responsible for integrating new cloud services, applications

and infrastructure, into the organization’s portfolio of internal systems and existing

cloud services.

These specialists assess the implications and trade-offs between different solutions

as they relate to the integration between external and internal systems, optimize integration

and user-experience, and ensure that performance standards adhere to service level agreements

set with the enterprise.

Cloud Data Engineers are responsible for designing, developing, and deploying scalable data pipelines

and data services.

They look at integrating new data management technologies and software engineering tools

into existing infrastructure.

Their responsibilities include: Understanding existing systems to recommend

automated integration of disparate data sets.

Collaborating with data scientists and researchers to develop predictive models and proofs of

concept.

Promoting best practices that enable teams to accelerate their consumption and understanding

of data.

Improving overall efficiency by introducing new engineering processes and tools.

Cloud Security Engineers provide expertise around the systems and processes needed to

protect the confidentiality, integrity, and availability of an organization’s systems

and application data.

They determine security requirements,

Plan, implement, and test security systems,

perform threat simulations to detect possible risks.

Recommend innovative technologies that will enhance the security of cloud-based environments.

Cloud Security Engineers need to have deep knowledge of cloud platforms and services,

software design patterns, and DevOps tools and methodologies.

Cloud DevOps Engineers collaborate with development and operations teams to create reliable and

rapid release pipelines for software and updates.

This may typically involve:

Creating custom automation tools.

Building and maintaining configuration and deployment frameworks.

Tracking design bugs and automating the debugging process for developers.

Maintaining and deploying web-based applications.

Monitoring security systems.

Measuring performance against expected business outcomes.

Containerization expertise is increasingly a must-have for DevOps Engineers.

Cloud Solutions Architects work to translate business requirements into application architecture

and design.

Some of the technical skills required for a Cloud Architect role include:

Deep knowledge of cloud platforms and services.

Deep understanding of software design patterns.

Knowledge of DevOps tools and methodologies.

Good understanding of networking.

A high-level understanding of key security concepts.

Solution Architects work closely with Cloud Developers, Networking Specialists, Security

Engineers, Integration Specialists, and DevOps Engineers to architect and design solutions.

There are several resources available for learning cloud technologies, in a variety

of delivery methods, including instructor-led courses, self-paced online courses, online

videos, books, and also technology focused community forums.

Many cloud providers have dedicated learning portals with extensive resources available

on the complete range of cloud technologies and services they provide.

They offer: learning paths, which make resources available

as per specific career roles.

Hands-on learning labs, with interactive learning resources that can be filtered by role, level,

or product.

Free trials on their suite of products and services.