Diploma in

IT, Networking and Cloud

Elective Module 1

Cloud Application Developer

Theory Manual

Learning Outcome

- Able to understand the Cloud, Architecture, App Development & Deployment in cloud environment.

- Understanding IBM Cloud, AWS cloud global infrastructure, scalability, elasticity, fault tolerance, reliability & durability aspects.
- Introduction to core services- compute, storage, network, database, security & management
- Getting started with IBM Cloud foundry, benefits, features, use cases, get introduced to Storage, PaaS Model services in IBM Cloud, Hosting web application son PaaS
- Understanding web application deployment strategies and planning. Preparing application for cloud deployment over PaaS model instance in IBM Cloud Foundry
- IBM Cloud CLI
- Cloud Best practices

Table of Contents

Introduction to cloud, benefits, types, service delivery models

Introduction to Cloud Computing

Cloud Computing refers to manipulating, configuring, and accessing the hardware and software resources remotely. It offers online data storage, infrastructure, and application.

Cloud Computing provides an alternative to the on-premises datacenter. With an on-premises datacenter, we have to manage everything, such as purchasing and installing hardware, virtualization, installing the operating system, and any other required applications, setting up the network, configuring the firewall, and setting up storage for data. After doing all the set-up, we become responsible for maintaining it through its entire lifecycle.

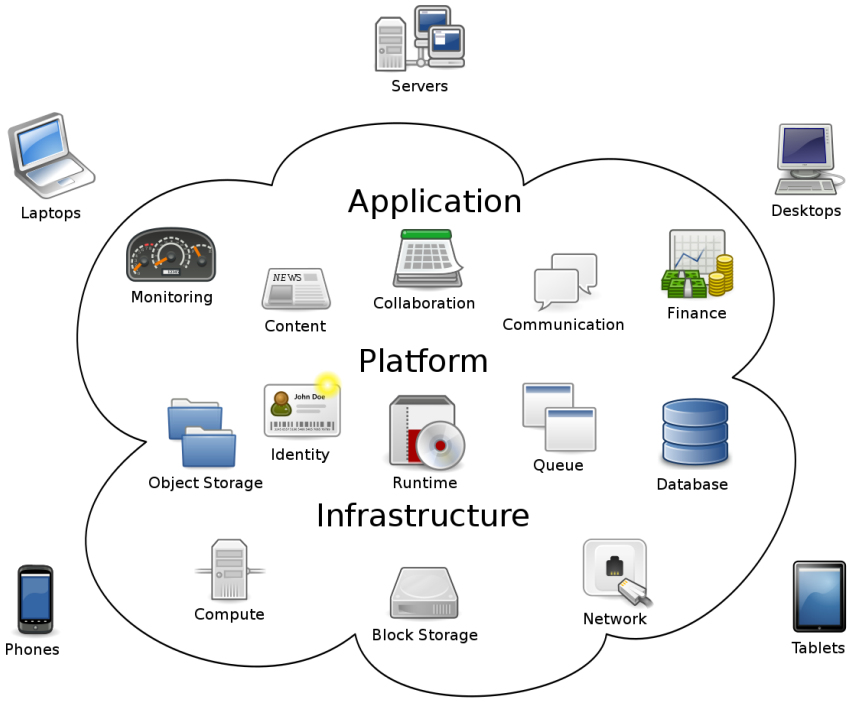


Image: Introduction to cloud computing
Reference: https://upload.wikimedia.org/wikipedia/commons/thumb/b/b5/Cloud_computing.svg/1200px-Cloud_computing.svg.png

But if we choose Cloud Computing, a cloud vendor is responsible for the hardware purchase and maintenance. They also provide a wide variety of software and platform as a service. We can take any required services on rent. The cloud computing services will be charged based on usage.

The cloud environment provides an easily accessible online portal that makes handy for the user to manage the compute, storage, network, and application resources. Some cloud service providers are in the following figure.

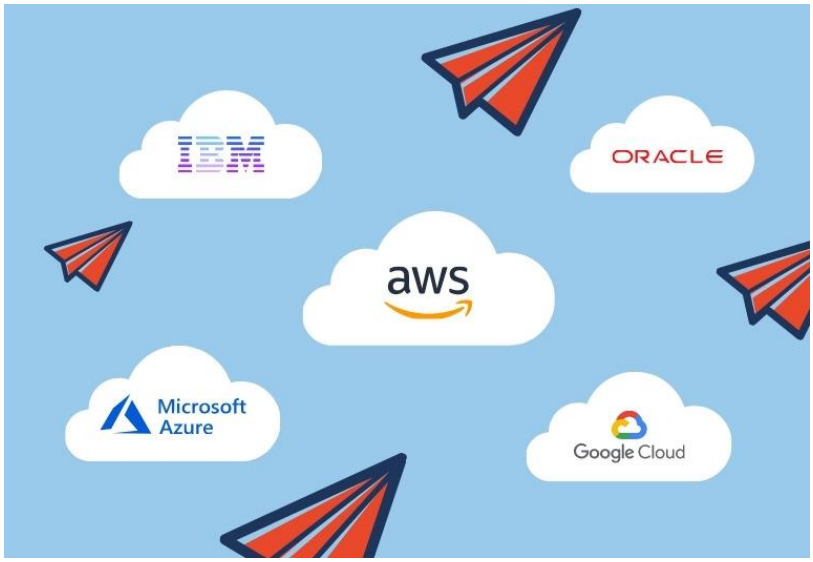


Image: Some cloud service providers
Reference: https://mettlesoft.com.au/wp-content/uploads/2020/10/cloud-computing-blog-post.jpg

Benefits of Cloud

Cost:

It reduces the huge capital costs of buying hardware and software.

Speed:

Resources can be accessed in minutes, typically within a few clicks.

Scalability:

We can increase or decrease the requirement of resources according to the business requirements.

Productivity:

While using cloud computing, we put less operational effort. We do not need to apply patching, as well as no need to maintain hardware and software. So, in this way, the IT team can be more productive and focus on achieving business goals.

Reliability:

Backup and recovery of data are less expensive and very fast for business continuity.
Security: Many cloud vendors offer a broad set of policies, technologies, and controls that strengthen our data security.

Image: Benefits of Cloud
Reference: https://thinkitsolutions.com/wp-content/uploads/2019/02/Benefits-of-Cloud-Computing.png

Types of Cloud

There are 3 main types of cloud computing: private clouds, public clouds and hybrid clouds. There are also 3 main types of cloud computing services: Infrastructure-as-a-Service (IaaS), Platforms-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

Choosing a cloud type or cloud service is a unique decision. No 2 clouds are the same (even if they're the same type), and no 2 cloud services are used to solve the same problem. But by

understanding the similarities, you can be more informed about how the caveats of each cloud computing type and cloud service might impact your business.

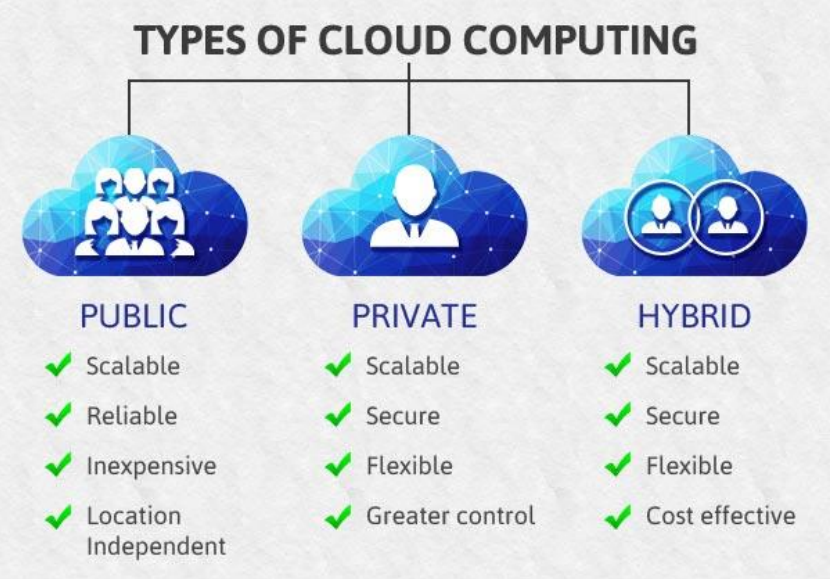


Image: Types of Cloud Computing
Reference: https://convergenceservices.in/images/typesofcloudcomputing.jpg

Public Cloud:

The cloud resources that are owned and operated by a third-party cloud service provider are termed as public clouds. It delivers computing resources such as servers, software, and storage over the internet. Public clouds are cloud environments typically created from IT infrastructure not owned by the end user. Some of the largest public cloud providers include Alibaba Cloud, Amazon Web Services (AWS), Google Cloud, IBM Cloud, and Microsoft Azure.

Traditional public clouds always ran off-premises, but today's public cloud providers have started offering cloud services on clients' on-premise data centers. This has made location and ownership distinctions obsolete.

All clouds become public clouds when the environments are partitioned and redistributed to multiple tenants. Fee structures aren't necessary characteristics of public clouds anymore, since some cloud providers (like the Massachusettes Open Cloud) allow tenants to use their clouds for free. The bare-metal IT infrastructure used by public cloud providers can also be abstracted and sold as IaaS, or it can be developed into a cloud platform sold as PaaS.

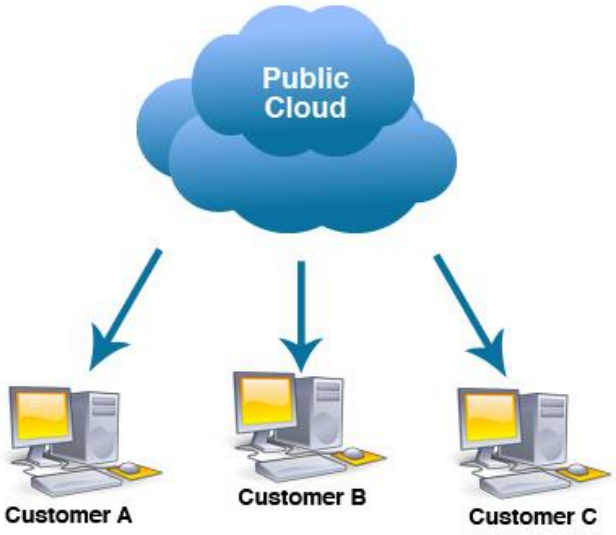


Image: Public Cloud
Reference: https://www.znetlive.com/blog/wp-content/uploads/2015/12/public-cloud.jpg

Private Cloud:

The cloud computing resources that are exclusively used inside a single business or organization are termed as a private cloud. A private cloud may physically be located on the company's on-site datacentre or hosted by a third-party service provider.

Private clouds are loosely defined as cloud environments solely dedicated to a single end user or group, where the environment usually runs behind that user or group's firewall. All clouds become private clouds when the underlying IT infrastructure is dedicated to a single customer with completely isolated access.

But private clouds no longer have to be sourced from on-prem IT infrastructure. Organizations are now building private clouds on rented, vendor-owned data centers located off-premises, which makes any location and ownership rules obsolete. This has also led to a number of private cloud subtypes, including:

Managed private clouds: Customers create and use a private cloud that's deployed, configured, and managed by a third-party vendor. Managed private clouds are a cloud delivery option that helps enterprises with understaffed or underskilled IT teams provide better private cloud services and infrastructure.

Dedicated clouds: A cloud within another cloud. You can have a dedicated cloud on a public cloud (e.g. Red Hat OpenShift® Dedicated) or on a private cloud. For example, an accounting department could have its own dedicated cloud within the organization's private cloud.

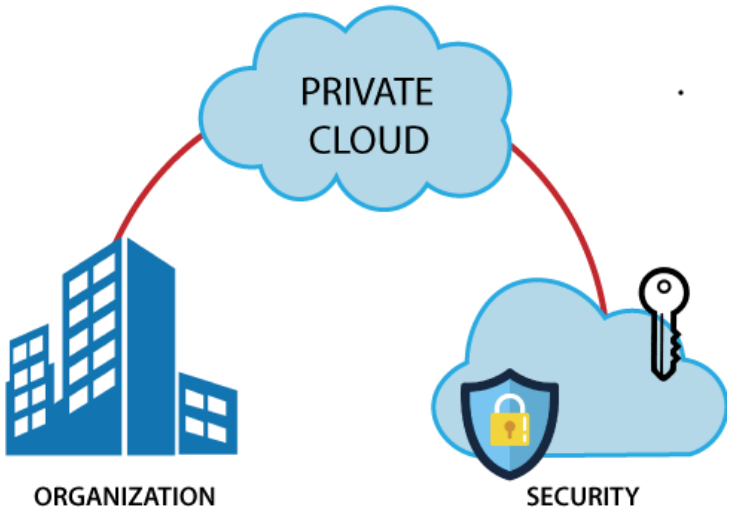


Image: Private Cloud
Reference: https://static.javatpoint.com/cloudpages/images/privatecloud.png

Hybrid Cloud:

It is the combination of public and private clouds, which is bounded together by technology that allows data applications to be shared between them. Hybrid cloud provides flexibility and more deployment options to the business.

A hybrid cloud is a seemingly single IT environment created from multiple environments connected through local area networks (LANs), wide area networks (WANs), virtual private networks (VPNs), and/or APIs.

The characteristics of hybrid clouds are complex and the requirements can differ, depending on whom you ask. For example, a hybrid cloud may need to include:

- At least 1 private cloud and at least 1 public cloud
- 2 or more private clouds
- 2 or more public clouds
- A bare-metal or virtual environment connected to at least 1 public cloud or private cloud

But every IT system becomes a hybrid cloud when apps can move in and out of multiple separate—yet connected—environments. At least a few of those environments need to be sourced from consolidated IT resources that can scale on demand. And all those environments need to be managed as a single environment using an integrated management and orchestration platform.

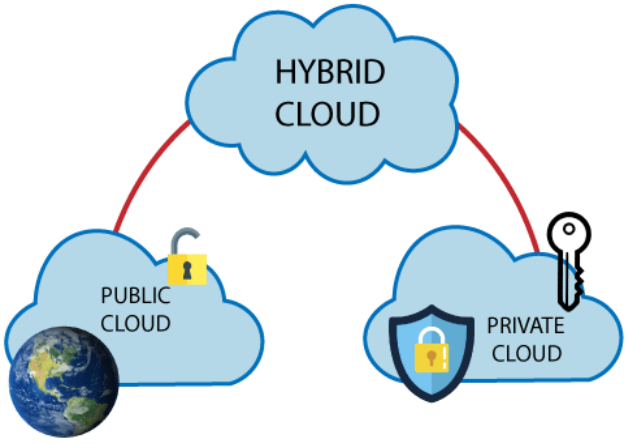


Image: Private Cloud
Reference: https://static.javatpoint.com/cloudpages/images/hybridcloud.png

Service Delivery Model

Cloud services are infrastructure, platforms, or software that are hosted by third-party providers and made available to users through the internet. There are 3 main types of as-a-Service solutions: IaaS, PaaS, and SaaS. Each facilitates the flow of user data from front-end clients through the internet, to the cloud service provider's systems, and back—but vary by what's provided.

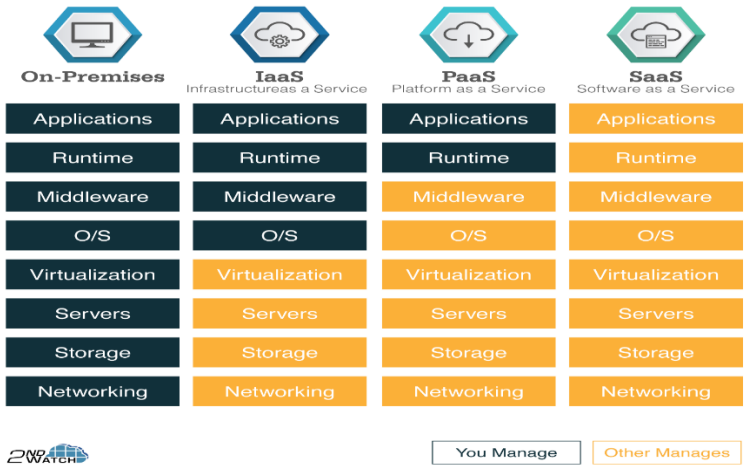


Image: Service Delivery Model
Reference: https://www.2ndwatch.com/wp-content/uploads/2021/08/2W_CloudComputingServiceModels_Infographic_2021-P1.png

Infrastructure as a Service (IaaS):

In IaaS, we can rent IT infrastructures like servers and virtual machines (VMs), storage, networks, operating systems from a cloud service vendor. We can create VM running Windows or Linux and install anything we want on it. Using IaaS, we don't need to care about the hardware or virtualization software, but other than that, we do have to manage everything else. Using IaaS, we get maximum flexibility, but still, we need to put more effort into maintenance. The simplest example of IaaS cloud computing is ordinary web-hosting. This is where you pay a monthly fee or by megabyte/gigabyte to have a company host your files from their servers. IaaS is an extremely flexible option, as it permits the user to customise the infrastructure of the computing environment. From web-hosting to big data analytics, IaaS covers the whole spectrum.

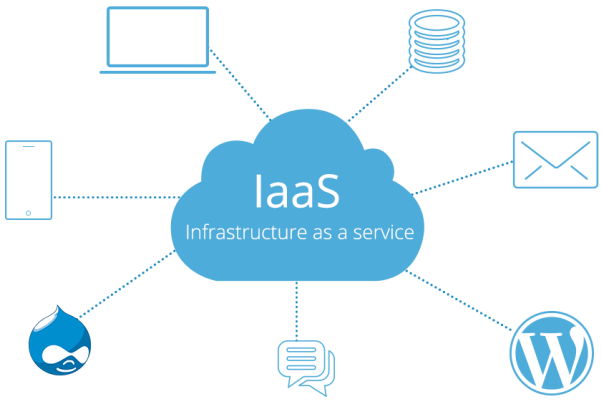


Image: IaaS
Reference: https://cdnblog.filecloud.com/blog/wp-content/uploads/2020/03/iaas-intro-01.png

Platform as a Service (PaaS):

This service provides an on-demand environment for developing, testing, delivering, and managing software applications. The developer is responsible for the application, and the PaaS vendor provides the ability to deploy and run it. Using PaaS, the flexibility gets reduce, but the management of the environment is taken care of by the cloud vendors. PaaS cuts down on the complexity of setting up and properly maintaining an infrastructure, while also allowing for supported collaboration between teams. An example of this is if you develop your own commerce site, but basically have the entire process running on a separate server. Like with SaaS, you're only exposed to the interface you interact with.

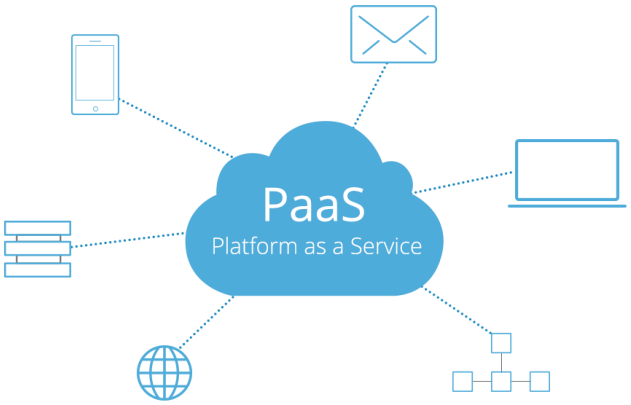


Image: Paas
Reference: https://www.milesweb.in/images/paas/paas-intro.png

Software as a Service (SaaS):

It provides a centrally hosted and managed software services to the end-users. It delivers software over the internet, on-demand, and typically on a subscription basis. E.g., Microsoft One Drive, Dropbox, WordPress, Office 365, and Amazon Kindle. SaaS is used to minimize the operational cost to the maximum extent. Examples of SaaS applications include any web-based mail services. The different services supplied by Google such as Google Docs and Google Sheets are also examples of SaaS. Adobe Creative Cloud services is also another example of SaaS in action. With this kind of model, the user is only exposed to the interface that they choose to interact with.

Image: Saas
Reference: https://www.milesweb.in/images/saas/saas-intro-1.png

IBM Cloud

What is IBM Cloud?

IBM Cloud® offers the most open and secure public cloud for business, a next-generation hybrid multicloud platform, advanced data and AI capabilities, and deep enterprise expertise across 20 industries.

IBM Cloud provides solutions that enable higher levels of compliance, security, and management, with proven architecture patterns and methods for rapid delivery for running mission-critical workloads.

IBM Cloud is a suite of cloud computing services from IBM that offers both platform as a service (PaaS) and infrastructure as a service (IaaS).

Image 1: IBM Cloud
Reference: https://en.wikipedia.org/wiki/IBM_Cloud

What is the IBM Cloud platform?

The IBM Cloud® platform combines platform as a service (PaaS) with infrastructure as a service (IaaS) to provide an integrated experience. The platform scales and supports both small development teams and organizations, and large enterprise businesses. Globally deployed across data centers around the world, the solution you build on IBM Cloud® spins up fast and performs reliably in a tested and supported environment you can trust!

IBM Cloud provides solutions that enable higher levels of compliance, security, and management, with proven architecture patterns and methods for rapid delivery for running mission-critical workloads. Available in data centers worldwide, across 19 countries with multizone regions in

North and South America, Europe, Asia, and Australia, you are enabled to deploy locally with global scalability.

IBM Cloud offers the most open and secure public cloud for business with a next-generation hybrid cloud platform, advanced data and AI capabilities, and deep enterprise expertise across 20 industries.

IBM Cloud Computing Models

- IaaS, or infrastructure as a service, is on-demand access to cloud-hosted physical and virtual servers, storage and networking - the backend IT infrastructure for running applications and workloads in the cloud.
- PaaS, or platform as a service, is on-demand access to a complete, ready-to-use, cloud-hosted platform for developing, running, maintaining and managing applications.
- SaaS, or software as a service, is on-demand access to ready-to-use, cloud-hosted application software.

IBM Cloud deployment models

IBM offers three deployment models for its cloud platform:

- **Public**: A public cloud that provides access to virtual servers in a multi-tenant environment. An enterprise can choose to deploy its applications in one or multiple geographical regions.
- **Dedicated**: A single-tenant private cloud that IBM hosts in one of its data centers. An enterprise can connect to the environment using a direct network connection or VPN, and IBM manages the platform.
- **IBM Cloud Private**: A version of the IBM platform that an organization deploys as a private cloud in its own data center behind a firewall.

What's built into the platform?

As the following diagram illustrates, the IBM Cloud platform is composed of multiple components that work together to provide a consistent and dependable cloud experience.

- A robust console that serves as the front end for creating, viewing, managing your cloud resources

- An identity and access management component that securely authenticates users for both platform services and controls access to resources consistently across IBM Cloud
- A catalog that consists of hundreds of supported products
- A search and tagging mechanism for filtering and identifying your resources
- An account and billing management system that provides exact usage for pricing plans and secure credit card fraud protection

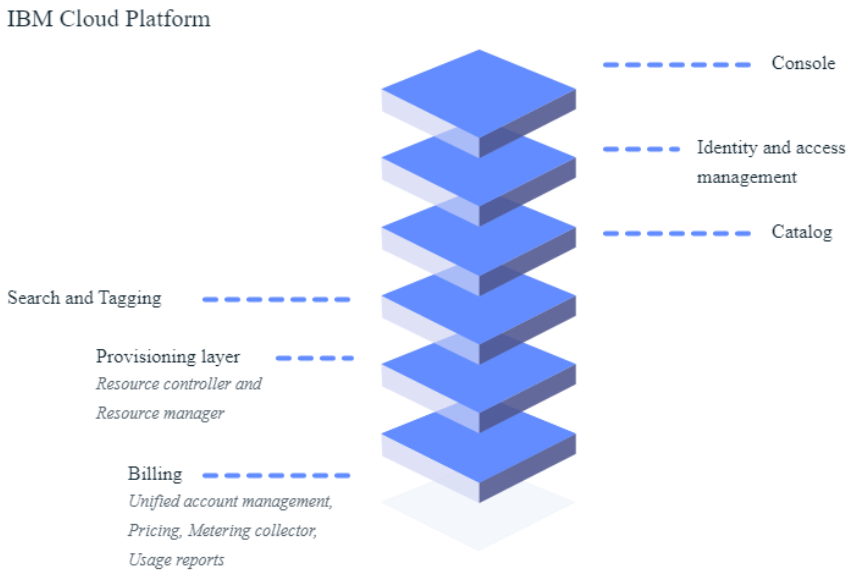


Image 2: Components of the IBM Cloud platform
Reference: https://cloud.ibm.com/docs/overview?topic=overview-whatis-platform

Why should I use IBM Cloud?

The IBM Cloud is also the only program that offers a true cross-platform experience to users by making all of its internal platforms easily accessible. As companies gain more confidence in the cloud's capabilities, they can automate more of their day-to-day functions and truly put their data to work.

IBM Cloud features

There are a number of IBM cloud services that are a part of the IBM cloud. These services are grouped into 16 categories:

- **AI/machine learning**: A collection of Watson-based AI resources and tools for building your own AI models.
- **Analytics**: Offers data science tools such as Apache Spark, Apache Hadoop and IBM Watson Machine Learning, as well as analytics services for streaming data.
- **Automation**: Automation resources enable business workflows to be automated using IBM Cloud Pak. Turbonomic is also available as an automation resource and can be used for application resource management and cost optimization.
- **Blockchain**: Provides IBM's Blockchain Platform, a SaaS offering to develop apps, enforce governance and monitor a blockchain network.
- **Compute**: Offers various compute resources, including bare-metal servers, VMs and serverless computing on which enterprises can host their workloads.
- **Containers**: IBM offers its own cloud Kubernetes service, as well as access to the container registry, Red Hat OpenShift and Istio (a server mesh for microservices).
- **Databases**: Provides a variety of SQL and NoSQL databases, as well as data querying, warehousing and migration tools.
- **Developer tools**: Includes a CLI, as well as a set of tools for continuous delivery, continuous release and application pipelines.
- **IBM Cloud Paks**: IBM Cloud Paks are applications that are certified for use on Red Hat Open Shift. Cloud Paks exist for business automation, data, integration, network automation, security and Watson.
- **Integration**: Offers services to integrate cloud and on-premises systems or various applications, such as API Connect, App Connect and IBM Secure Gateway.
- **Internet of things (IoT)**: Includes the IBM IoT Platform, which provides services that connect and manage IoT devices, and analyzes the data they produce.
- **Logging and monitoring**: Provide tools to log, manage and monitor cloud deployments, including Cloud Activity Tracker, Cloud Log Analysis and Cloud Monitoring.
- **Networking**: Provides cloud networking services, such as a load balancer, a content delivery network, VPN tunnels and firewalls.
- **Quantum**: Provides the ability to run workloads on quantum systems through IBM Quantum composer, the IBM Quantum Lab and the Qiskit SDK.

- **Security**: Includes services for activity tracking, identity and access management and authentication.
- **Storage**: IBM's cloud storage offerings include object, block and file storage for cloud data.

IBM Cloud Products

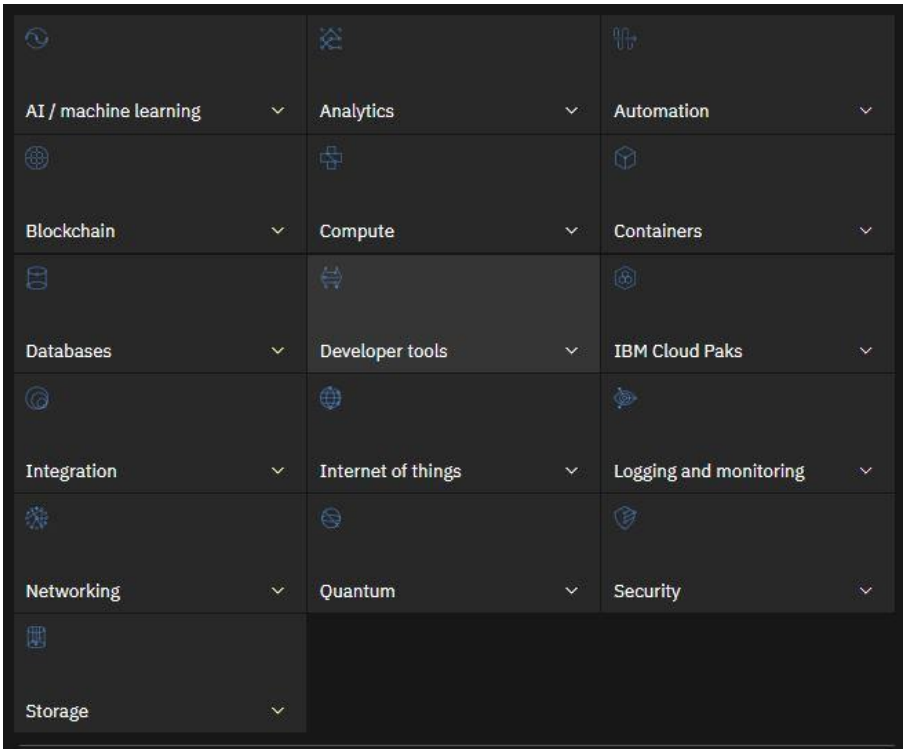


Image 3: IBM Cloud Products
Reference: https://www.ibm.com/cloud/products

AI / machine learning- Use Watson's AI or build your own machine

- IBM Watson Assistant-Virtual agents customizable to any domain
- IBM Watson Discovery-Search and analytics engine that adapts to custom domains
- IBM Watson Knowledge Catalog-SaaS for AI data management

- IBM Watson Knowledge Studio-Visual interface to teach Watson domain-specific knowledge
- IBM Watson Language Translator-API for translation with domain-specific models
- IBM Watson Natural Language Classifier-Visual tool and API for text classification
- IBM Watson Natural Language Understanding-API for text analysis and metadata extraction
- IBM Watson Speech to Text-API for real-time speech recognition and transcription
- IBM Watson Studio-IDE to build, run and manage AI models
- IBM Watson Text to Speech-API for real-time text to speech conversion

Analytics-Aggregate and analyze large datasets

- IBM Analytics Engine-PaaS to build analytics applications on Apache Spark and Hadoop
- IBM Cloud SQL Query-Query tool for IBM Cloud Object Storage
- IBM Db2 Warehouse on Cloud-Managed data warehouse on IBM Cloud or AWS
- IBM InfoSphere® Information Server on Cloud-Data integration suite for ETL, governance and anlaysis
- IBM Master Data Management on Cloud-Managed master data storage
- IBM Streaming Analytics-Dashboard for real-time analysis of data streams

Automation-Automate workflows from IT operations to business processes

- IBM Cloud Pak for Business Automation-Operations management software with AI insights
- IBM Cloud Pak for Integration-Tools to connect all of your apps, data and events
- IBM Cloud Pak for Network Automation-Management software for telco network operations
- IBM Cloud Pak for Watson AIOps-DevOps management tool with AI analysis and recommendations
- Turbonomic-Software to automate application resource management and optimize costs

Blockchain-Build applications on an immutable blockchain ledger

- IBM Blockchain Platform-Blockchain SDK based on Hyperledger Fabric

Compute-Run workloads on cloud infrastructure

- Cloud Foundry-PaaS for Java, Node, PHP, Python, Ruby, Swift and Go apps.
- Dizzion Managed DaaS on IBM Cloud-Virtual desktops, in the cloud and managed
- IBM Cloud Bare Metal Servers-Dedicated hardware for maximum performance
- IBM Cloud Code Engine-Serverless container hosting platform as a service
- IBM Cloud for VMware Solutions-Hosting for virtualized workloads
- IBM Cloud Functions-Serverless function runtime based on Apache OpenWhisk
- IBM Cloud Virtual Server for VPC-Virtual servers with enhanced network security
- IBM Cloud Virtual Servers for Classic Infrastructure-Virtual servers with the most flexible configuration options
- IBM Hyper Protect Virtual Servers-Virtual servers for workloads with sensitive data
- IBM Power Systems Virtual Servers-Virtual servers with IBM POWER processors
- IBM WebSphere Application Server on Cloud-Runtime and SDK for Java applications

Containers-Deploy applications consistently across environments

- Container registry-SaaS for container storage and management
- IBM Cloud Kubernetes Service-Container hosting with self-healing and horizontal scaling
- Istio-A service mesh for microservices in Kubernetes clusters
- Red Hat® OpenShift® on IBM Cloud-Container hosting on managed Red Hat Linux servers

Databases-Store, query and analyze structured data

- IBM Cloud Databases for Elasticsearch-Managed JSON document store for full-text search
- IBM Cloud Databases for EnterpriseDB-Managed SQL database
- IBM Cloud Databases for etcd-Managed distributed key-value store
- IBM Cloud Databases for MongoDB-Managed NoSQL JSON document store
- IBM Cloud Databases for PostgreSQL-Managed SQL database
- IBM Cloud Databases for Redis-Managed in-memory key value store
- IBM Cloudant®-Managed PCI-compliant JSON document store on CouchDB
- IBM Db2 Warehouse on Cloud-Managed data warehouse on IBM Cloud or AWS
- IBM Db2® on Cloud-Managed SQL database
- IBM Hyper Protect DBaaS-Managed PostgreSQL and MongoDB for sensitive data

- IBM Informix® on Cloud-Managed DB for time series, spatial, NoSQL and SQL data

Developer tools-Manage infrastructure, environments and deployments

- IBM Cloud App Configuration-SDK, dashboard and API for application feature management
- IBM Cloud CLI-A CLI to manage your IBM Cloud
- IBM Cloud Continuous Delivery-UI and CLI based DevOps workflows based on Tekton Pipelines
- IBM Cloud Messages for RabbitMQ-Managed message broker
- IBM Cloud Schematics-Managed service to provision resources with terraform templates
- IBM Cloud SQL Query-Query tool for IBM Cloud Object Storage
- Tekton-Kubernetes-native CI and CD pipelines

IBM Cloud Paks-Integrated and certified applications on Red Hat OpenShift

- IBM Cloud Pak for Business Automation-Operations management software with AI insights
- IBM Cloud Pak for Data-Tools for data analysis, organization and management
- IBM Cloud Pak for Integration-Tools to connect all of your apps, data and events
- IBM Cloud Pak for Network Automation-Management software for telco network operations
- IBM Cloud Pak for Security-Security auditing, reporting, analysis and governance
- IBM Cloud Pak for Watson AIOps-DevOps management tool with AI analysis and recommendations

Integration-Integrate your apps through APIs, messaging and networking tools

- IBM API Connect®-A toolkit to rapidly create, secure and manage APIs
- IBM App Connect-Codeless connectors for your data, apps and APIs
- IBM Aspera® on Cloud-UI and CLI to rapidly move large files
- IBM Cloud for Skytap Solutions-Virtualization for x86, AIX, System i and Power applications
- IBM Cloud Pak for Integration-Tools to connect all of your apps, data and events
- IBM Event Streams-PaaS stream processing based on Apache Kafka
- IBM Lift-Toolkit to migrate on-premises data to the cloud

- IBM MQ on Cloud-Managed message broker
- Secure gateway-Secure tunnels between cloud and external environments

Internet of things-Connect and manage edge devices

- IBM Edge Application Manager-An autonomous management platform for edge computing
- IBM Watson IoT® Platform-SaaS for device management, monitoring and data storage

Logging and monitoring-Log and analyze activity in your infrastructure

- IBM Cloud Activity Tracker-Dashboard to monitor and analyze events in IBM Cloud
- IBM Cloud Log Analysis-Analysis dashboard for system and application logs
- IBM Cloud Monitoring-Managed dashboard for Kubernetes devops
- IBM Cloud Pak for Watson AIOps-DevOps management tool with AI analysis and recommendations
- Networking-Run and manage public, private and virtual networks
- Domain name services-Public and private DNS and domain registration
- IBM Cloud Content Delivery Network-Content caching and delivery on the Akamai network
- IBM Cloud Direct Link-Physical or virtual private connections to IBM Cloud
- IBM Cloud Internet Services-Load balancing and CDN solution
- Load balancer-Traffic flow management for cloud applications
- Network appliances-Physical and virtual routers, firewalls, VPNs and more
- Network security-Hardware and software firewalls, network security group management

Quantum-Run code on cutting-edge quantum systems and simulators

- Qiskit-Python SDK for open-source quantum development
- Qiskit Runtime-Low-latency execution environment for quantum programs

Security-Secure your cloud resources and simplify regulatory compliance

- IBM Cloud App ID-Compliant authentication, authorization and user data SaaS
- IBM Cloud Certificate Manager-Certificate management for cloud resources
- IBM Cloud Data Shield-Runtime encryption for data-in-use protection on Kubernetes clusters

- IBM Cloud Hardware Security Module-Tamper-resistant hardware to store and process cryptographic keys
- IBM Cloud Secrets Manager-Single-tenant, dedicated instance to manage your secrets
- IBM Cloud Security Advisor-Dashboard for security management, analysis and remediation
- IBM Cloud Security and Compliance Center-SaaS to define and audit the compliance posture of your cloud
- IBM Hyper Protect Crypto Services-Cloud hardware to store and process cryptographic keys
- IBM Key Protect-Encryption key provisioning and storage for IBM Cloud apps
- Network security-Hardware and software firewalls, network security group management
- SSL certificates-Certificates for cloud resources

Storage-Store, migrate and back up your data

- IBM Cloud Backup-Fully encrypted backup and recovery across multiple datacenters
- IBM Cloud Block Storage-The lowest latency, highest redundancy data storage option
- IBM Cloud File Storage-The simplest solution for data organized into files and folders
- IBM Cloud Mass Data Migration-Secure hardware to physically ship petabytes of data to IBM Cloud
- IBM Cloud Object Storage-Unstructured cloud data storage with an API frontend

IBM Cloud Global Infrastructure

Infrastructure as a service

Infrastructure as a service (IaaS) is a type of cloud computing that lets you allocate your compute, network, storage and security resources on demand. The IBM approach to IaaS lets you scale and shrink resources as needed around the world in more than 60 data centers.

Get access to the full stack of compute, down to the bare metal. Get more control. Customize hardware to your exact specifications to meet the precise demands of your workload.

Image 4: IBM Cloud Global Infrastructure
Reference: https://www.ibm.com/in-en/cloud/infrastructure

Data Centers, Region and Availability Zones

A data center is a facility that centralizes an organization's shared IT operations and equipment for the purposes of storing, processing, and disseminating data and applications.

Deploy workloads in over 60 data centers across 6 regions and 19 availability zones across globally, located in the United States (Dallas and Washington, DC), Germany, UK, Japan, and Australia. These regions provide a full cloud service stack that enables highly available, redundant, and geographically dispersed customer solution

What is a data center?

A data center is the physical facility that makes enterprise computing possible, and it houses the following:

- Enterprise computer systems.
- The networking equipment and associated hardware needed to ensure the computer systems' ongoing connectivity to the Internet or other business networks.

- Power supplies and subsystems, electrical switches, backup generators, and environmental controls (such as air conditioning and server cooling devices) that protect the data center hardware and keep it up and running.

A data center is central to an enterprise's IT operations. It's a repository for the majority of business-critical systems, where most business data is stored, processed, and disseminated to users.

Maintaining the security and reliability of data centers is essential to protecting an enterprise's operational continuity—it's ability to conduct business without interruption.

What is in a data center?

The IT equipment within a data center consists of three main elements necessary for a computing environment to function:

- Compute: The memory and processing power needed to run applications that are usually supplied by enterprise-grade servers.
- Storage: Data centers include primary and backup storage devices. They may be hard disk or even tape drives, but best-in-class facilities typically feature all-flash arrays.
- Networking: They contain a broad array of networking equipment, ranging from routers and switches to controllers and firewalls.

In addition to the IT equipment it contains, every data center houses that equipment's support infrastructure, including the following:

- Environmental controls: Sensors monitor the airflow, humidity, and temperature in the facility at all times, with systems in place to guarantee that temperature and humidity remain within hardware manufacturers' specified ranges.
- Server racks: Most data center equipment is housed in specially designed racks or in purpose-built cabinets or shelving.
- Power supplies: Most data centers employ battery-based backup power systems able to compensate for short-term power outages and larger generators that can supply power in case longer commercial power grid outages occur.
- Cabling and cable management systems: An enterprise data center may contain hundreds of miles of fiber optic cable. Systems and equipment to keep that cabling orderly and accessible are a must.

Regions

IBM Cloud® Functions is available in the US South, US East, Germany, United Kingdom, Tokyo, and Sydney IBM Cloud regions

What is an availability zone?

When you create an instance, after you select the IBM Cloudant tile, you must select a region. These locations are called availability zones. An availability zone is an IBM Cloud® Public location that hosts your data. All Lite and Standard plans automatically deploy into a multi-zone region. Dedicated Hardware plan instances can be deployed in most IBM data center locations.

What is the difference between a single-zone and a multi-zone region?

A **multi-zone** region includes three availability zones that can be used by an instance that is deployed to that region. The multi-zone regions available with IBM Cloudant include the following regions:

- Dallas
- Frankfurt
- London
- Osaka
- Sydney
- Tokyo
- Washington DC

A **single-zone** region offers only one availability zone for that region. The single-zone regions available with IBM Cloudant include the following regions:

- Seoul
- Chennai

Infrastructure (Compute, Networking, and Storage)

- Bare metal servers-Get the raw IaaS power you need for your processor-intensive workloads.
- Virtual private cloud (VPC)-Define and control your virtual networks in IBM public cloud.

- Virtual servers-Choose virtual servers when you need more resources to meet sudden demand.
- Object storage-Flexible, cost-effective and scalable cloud storage for unstructured data. Get 25 GB per month, cost free.
- Kubernetes Service-Build cloud-native applications or more easily modernize existing apps.
- Compute-Access high-performance cloud servers in over 60 data centers across the globe.
- Storage-Scale capacity without interruption. Flexibly deploy across the globe to achieve higher performance for applications.
- Networking-Help enhance your traffic, satisfy your users and protect your sanity.

Benefits

- Accelerate performance-Deploy an IaaS bare metal server, a virtual server or both in minutes.
- Integrate with your systems-Take advantage of our services being integrated to optimize performance.
- Employ enterprise-grade hardware-Select from more than bare metal and virtual, including SAP-certified infrastructure.
- Extend your on-premises data centers-Create, deploy and manage servers on their own or in a hybrid cloud.

IBM Cloud Scalability

Cloud scalability in cloud computing refers to the ability to increase or decrease IT resources as needed to meet changing demand. Scalability is one of the hallmarks of the cloud and the primary driver of its exploding popularity with businesses.

System scalability for your hybrid cloud

Extend your hybrid cloud with industry-leading IT infrastructure that provides the performance and reliability needed to modernize your IT operations. As your business needs expand and grow, you can take advantage of Capacity on Demand to gain the benefits and control of on-premises solutions.

Benefits

- Ensure business continuity-A dynamic, pay-for-use consumption model with leading business continuity and agility lets you maintain complete control of your hybrid cloud resources.
- Secure and control data on-prem or in flight-Deploy your workloads on-premises or in the cloud, all while maintaining control of your data and reducing the risks associated with a cyber security breach.
- Avoid unplanned downtime-Embrace a hybrid cloud strategy that delivers key business functions and ensures that unplanned downtime won't affect your customers' experience.
- Confidently move applications-Digitally transform your cloud infrastructure with next-level security and stability, giving you agile deployment, deeper insights and ultimate uptime.

Three forms for scalability

Manual scaling

Manual scalability starts with a manual forecast of the expected workload on the cluster or farm of resources, then manually adding resources to add capacity. This is done using mostly physical servers, which are manually installed and configured. Ordering, installing and configuring physical resources take a lot of time, so the forecasting needs to be done weeks if not months in advance. Another downside to manual scalability is that removing resources mostly does not result in cost savings as the physical server has been paid for already.

Semi-automated scaling

Semi-automated scalability takes advantage of the concept of virtual servers, which are provisioned (installed) using predefined images. Either a manual forecast or automated warning of system monitoring tooling will trigger operations to expand or reduce the cluster or farm of resources.

By using predefined, tested and approved images, every new virtual server will be exactly the same as all the others (except for some minor configuration) which gives you repetitive results. This also reduced the manual labor on the systems significantly, and it is a well-known fact that around 70 to 80 percent of all errors are caused by manual actions on systems.

Using virtual servers also has a huge benefit, this does allow getting cost savings once a virtual server is de-provisioned (removed). Freed resources can be directly utilized for other purposes.

Elastic scaling (fully-automated scaling)

Elasticity, or fully-automated scalability, takes advantage of the same concepts that semi-automated scalability does, but removes any manual labor needed to increase or reduce capacity. Everything is controlled by triggers of the system monitoring tooling, which gives you this "rubber band" effect. If more capacity is needed now, then it's added now and there within minutes. No need for the additional capacity anymore? Based on the system monitoring tooling the capacity is reduced instantly.

Auto scale

Auto scale for IBM Cloud® Virtual Servers provides you with the ability to automate the manual scaling process that's associated with adding or removing instances to support your business applications. This automation sets up new instances automatically as more resources are needed and then those instances are shut down and removed when the extra load subsides. Auto scale uses groups to contain the policies that change how your environment expands or shrinks. These policies use actions to add or remove virtual server based on your business and application needs.

Auto scale features

Auto scaling enables the following features:

- Seamless and automatic scaling up of instances when more resources are required due to demand
- Seamless and automatic scaling down of instances by removing unnecessary resources when demand goes down (saving you money)
- Flexible scaling triggers: CPU percentage, outgoing public and private bandwidth, and incoming public and private bandwidth
- Near-real-time status updates for scaling activity in groups
- Optional integration of virtual LAN (VLAN) and local load balancers

Two common business solutions exist to which you can use auto scaling:

Schedule-based scaling:

Schedule-based scaling can be used to set up policies for time-based usage spikes, like over weekends or holidays. Schedule-based scaling can be used when a company is expecting traffic to spike, for example, a social networking site that requires more resources based on a schedule.

Resource-based scaling:

Resource-based scheduling can be used to set up policies for irregular spikes, based on resource usage. Irregular spikes in traffic can occur when you have a push to get a product to market or an e-commerce site is having a sale and resources are needed to sustain response times.

IBM Cloud Elasticity

Cloud Elasticity is the property of a cloud to grow or shrink capacity for CPU, memory, and storage resources to adapt to the changing demands of an organization.

Elasticity is basically a 'rename' of scalability, which has been a known non-functional requirement in IT architectures for many years already. Scalability is the ability to add or remove capacity, mostly processing, memory, or both, to or from an IT environment when this is needed. This it typically done in two ways:

- **Horizontal scalability**: Adding or removing nodes, servers or instances to or from a pool like a cluster or farm.
- **Vertical scalability**: Adding or removing resources to an existing node, server or instance to increase the capacity of the node, server or instance.

Most implementations of scalability are implemented using the horizontal method, as this easiest to implement especially in the current web-based world we live in. A well-known example is adding a load balancer in front of a farm of web servers that distributes the requests. Vertical scaling is less dynamic most of the time because this requires reboots of systems, sometimes adding physical components to servers.

IBM Cloud Fault Tolerance

Fault tolerance is a system's ability to continue functioning during a partial failure. Creating a resilient system places requirements on all of the services in it. The dynamic natures of cloud environments demand that services are written to expect and respond gracefully to the unexpected.

Fault tolerance is a process that enables an operating system to respond to a failure in hardware or software. This fault-tolerance definition refers to the system's ability to continue operating despite failures or malfunctions.

Fault Tolerance in Security

A system is fault-tolerant if it can continue to perform despite parts failing. Fault tolerance helps to make your remote-boot infrastructure more robust.

In the case of OS deployment servers, the whole system is fault-tolerant if the OS deployment servers back up each other. When a server fails, other servers handle the requests from the down server.

Implementing fault tolerance at the Tivoli® Provisioning Manager for Images level does not mean that your whole network infrastructure is fault-tolerant. You can implement fault-tolerances at all levels:

- At the physical level, by having redundant power sources (if all OS deployment servers are out of power at the same time, fault-tolerance at the product level is useless)
- At the network level, by having backup network links, and backup active elements (the backup server must be able to reach remote-boot targets)
- At the network operating system level, by having multiple network domains, or by running OS deployment servers outside of your domain architecture (OS deployment servers should not be all linked to the same NT PDC, or the same NFS server)
- At the DHCP level, by having multiple DHCP servers on the same subnet
- At the Tivoli Provisioning Manager for Images level, by implementing the fault-tolerance instructions.
- At the operating system level. If Tivoli Provisioning Manager for Images is able to survive to a severe problem, but then the operating system cannot find its network server, fault tolerance is useless

The following sections present information about how to implement fault tolerance at the DHCP and Tivoli Provisioning Manager for Images levels. Other levels are beyond the scope of this document.

Fault tolerance at the DHCP level

The DHCP protocol allows the implementation of fault tolerance and load-balancing very easily. If you connect two DHCP servers to the same IP subnet, and both servers are configured to serve IP addresses on this subnet, the protocol handles all conflicts between the two servers. A system is fault-tolerant if it can continue to perform despite parts failing. Load balancing specifies the maximum number of DHCP/BINL requests to a OS deployment server in one minute.

Fault tolerance at the Tivoli Provisioning Manager for Images level

Fault tolerance helps to make your remote-boot infrastructure more robust. A system is fault-tolerant if it can continue to perform despite parts failing. Fault tolerance at the product level is implemented with two configuration parameters: Backup and BootReplyDelay.

IBM Cloud Reliability

Reliability is composed of a number of terms or factors including resiliency, robustness, and availability. It's important to have a clear definition of those factors.

Resiliency

The ability of a system or workload to recover from adverse conditions ranging from, but not limited to, infrastructure, network, or loss of compute resource

Robustness

The ability of a system to handle various conditions either individually or in combination.

Availability

The percentage of time that a system or a workload is available to be used or be consumed. It is defined as a service level objective (SLO) or more formally as a service level agreement (SLA).

Availability = (system availability time/total time) × 100

How resiliency, robustness, and availability contribute toward increased reliability?

A reliability architecture requires the consideration of the availability of each component within a system. The availability of each component depends on the resiliency or robustness of the component. You can impact resiliency and robustness of a component and ultimately, the system, in several ways:

Chaos testing

Deliberate and destructive preproduction testing of components to find breakpoints either by volume or infrastructure failure can reduce the likelihood of failure in production.

Automation

Can be used to reduce human error in mundane repeatable tasks. For example, DevOps pipelines can reduce build and deployment configuration errors that can impact availability.

Horizontal scaling

Spreading load on a component over a number of instances can reduce the number of times that a component is impacted by volume and increase robustness. For microservices, it also allows better maintenance management and reduces potential failures that can occur during maintenance windows.

Failure monitoring

Monitoring and alerting can be used to engage automation to resolve commonly observed issues that a system is experiencing and increase resiliency.

System complexity

The number of distinct components in a system also impacts the overall reliability of a system. Increasing availability comes at a price and needs to be weighed against the nonfunctional requirements of the system.

Other important aspects of increasing reliability are around good Site Reliability Engineering (SRE) practices and multicloud or hybrid cloud considerations that can impact availability through latency issues. Another important consideration is the security of a component or system. A compromised component can affect its neighbors and reduce the overall reliability of a system.

IBM Cloud Durability

Durability, on the other hand, refers to long-term data protection, i.e. the stored data does not suffer from bit rot, degradation or other corruption. Rather than focusing on hardware redundancy, it is concerned with data redundancy so that data is never lost or compromised.

Durability of Block Storage

Think of durability as a measurement of how healthy and resilient your data is. Durability in Block Storage means that your data is stored consistent and intact without any signs of data decay, influence of drive failures, or any other form of corruption. 99.999999999% (11 nines) durability means that if you store 10 million files, then you expect to lose one file every 10000 years.

When people hear the word durability, most of them think of hardware failures of storage, compute, and network components that could cause data loss. In Block Storage, your data is protected against drive failures and numerous type of disk errors that otherwise might negatively impact data durability and data integrity. The data is stored redundantly across multiple physical disks in an Availability Zone to prevent data loss due to failure of any single component.

Other than physical failure, a common source of data loss is accidental deletion or modifications of files by users. Block Storage is only accessible to authorized hosts within your network. You control who can access it. Another measure to protect against accidental deletion and modification of files by users is a snapshot. If a user accidentally modifies or deletes crucial data from a volume, the data can be easily and quickly restored from a snapshot copy.

The 11 nines durability target applies to a single Availability Zone. To protect against natural or man-made disasters that could destroy an entire Availability Zone, consider storing your most important data in multiple locations.

AWS

What is AWS?

Amazon Web Services (AWS) is an excellent cloud service that provides you the services like Storage, compute, databases, security, etc. and you can access those services from any corner of the globe.

Amazon Web Services (AWS) is very much secure, reliable. AWS is one of the most demanded cloud services because of the best quality of services and benefits. Along with this, as support is quite important, it provides a nice support system.

Image 5: AWS
Reference: https://aws.amazon.com/what-is-aws/

AWS Cloud Computing Models

There are three cloud computing models available on AWS.

Infrastructure as a Service (IaaS):

It is the basic building block of cloud IT. It generally provides access to data storage space, networking features, and computer hardware (virtual or dedicated hardware). It is highly flexible and gives management controls over the IT resources to the developer. For example, VPC, EC2, EBS.

Platform as a Service (PaaS):

This is a type of service where AWS manages the underlying infrastructure (usually operating system and hardware). This helps the developer to be more efficient as they do not have to worry about undifferentiated heavy lifting required for running the applications such as capacity planning, software maintenance, resource procurement, patching, etc., and focus more on deployment and management of the applications. For example, RDS, EMR, ElasticSearch

Software as a Service (SaaS):

It is a complete product that usually runs on a browser. It primarily refers to end-user applications. It is run and managed by the service provider. The end-user only has to worry about the application of the software suitable to its needs. For example, Saleforce.com, Web-based email, Office 365

AWS Deployment Models

Cloud

A cloud-based application is fully deployed in the cloud and all parts of the application run in the cloud. Applications in the cloud have either been created in the cloud or have been migrated from an existing infrastructure to take advantage of the benefits of cloud computing. Cloud-based applications can be built on low-level infrastructure pieces or can use higher level services that provide abstraction from the management, architecting, and scaling requirements of core infrastructure.

Hybrid

A hybrid deployment is a way to connect infrastructure and applications between cloud-based resources and existing resources that are not located in the cloud. The most common method of hybrid deployment is between the cloud and existing on-premises infrastructure to extend, and grow, an organization's infrastructure into the cloud while connecting cloud resources to internal system. For more information on how AWS can help you with your hybrid deployment, please visit our hybrid page.

On-premises

Deploying resources on-premises, using virtualization and resource management tools, is sometimes called “private cloud”. On-premises deployment does not provide many of the benefits of cloud computing but is sometimes sought for its ability to provide dedicated resources. In most

cases this deployment model is the same as legacy IT infrastructure while using application management and virtualization technologies to try and increase resource utilization.

Why AWS services?

Amazon, the preeminent cloud vendor, broke new ground by establishing the first cloud computing service, Amazon EC2, in 2008. AWS offers more solutions and features than any other provider and has free tiers with access to the AWS Console, where users can centrally control their ministrations.

Designed around ease-of-use for various skill sets, AWS is tailored for those unaccustomed to software development utilities. Web applications can be deployed in minutes with AWS facilities, without provisioning servers or writing additional code.

Amazon hosts global data centers with a vast network ensuring reduced latency worldwide. AWS' replication capacity allows you to duplicate services regionally, helping you recover quickly and avoid costly downtime.

AWS Key Services

Amazon Web Services (AWS) is one of the leading cloud service providers in the market now that provides the best quality cloud solutions across countries with more than 200 cloud services.

Amazon Web Services offers a wide range of different business purpose global cloud-based products. The products include storage, databases, analytics, networking, mobile, development tools, enterprise applications, with a pay-as-you-go pricing model.

Image 6: AWS Services

Reference: https://3nvaqm3pc5a23uy435vgbqe1-wpengine.netdna-ssl.com/wp-content/uploads/2019/03/aws-native-services-1024x681.png

AWS Compute Services

Here, are Cloud Compute Services offered by Amazon:

- EC2(Elastic Compute Cloud)- EC2 is a virtual machine in the cloud on which you have OS level control. You can run this cloud server whenever you want.
- LightSail- This cloud computing tool automatically deploys and manages the computer, storage, and networking capabilities required to run your applications.
- Elastic Beanstalk- The tool offers automated deployment and provisioning of resources like a highly scalable production website.
- EKS (Elastic Container Service for Kubernetes)- The tool allows you to Kubernetes on Amazon cloud environment without installation.
- AWS Lambda- This AWS service allows you to run functions in the cloud. The tool is a big cost saver for you as you to pay only when your functions execute.

Migration

Migration services used to transfer data physically between your datacenter and AWS.

- DMS (Database Migration Service)- DMS service can be used to migrate on-site databases to AWS. It helps you to migrate from one type of database to another -for example, Oracle to MySQL.
- SMS (Server Migration Service)-SMS migration services allows you to migrate on-site servers to AWS easily and quickly.
- Snowball-Snowball is a small application which allows you to transfer terabytes of data inside and outside of AWS environment.

Storage

- Amazon Glacier- It is an extremely low-cost storage service. It offers secure and fast storage for data archiving and backup.
- Amazon Elastic Block Store (EBS)- It provides block-level storage to use with Amazon EC2 instances. Amazon Elastic Block Store volumes are network-attached and remain independent from the life of an instance.

- AWS Storage Gateway- This AWS service is connecting on-premises software applications with cloud-based storage. It offers secure integration between the company's on-premises and AWS's storage infrastructure.

Security Services

- IAM (Identity and Access Management)-IAM is a secure cloud security service which helps you to manage users, assign policies, form groups to manage multiple users.
- Inspector-It is an agent that you can install on your virtual machines, which reports any security vulnerabilities.
- Certificate Manager-The service offers free SSL certificates for your domains that are managed by Route53.
- WAF (Web Application Firewall)-WAF security service offers application-level protection and allows you to block SQL injection and helps you to block cross-site scripting attacks.
- Cloud Directory-This service allows you to create flexible, cloud-native directories for managing hierarchies of data along multiple dimensions.
- KMS (Key Management Service)-It is a managed service. This security service helps you to create and control the encryption keys which allows you to encrypt your data.
- Organizations-You can create groups of AWS accounts using this service to manages security and automation settings.
- Shield- Shield is managed DDoS (Distributed Denial of Service protection service). It offers safeguards against web applications running on AWS.
- Macie-It offers a data visibility security service which helps classify and protect your sensitive critical content.
- GuardDuty- It offers threat detection to protect your AWS accounts and workloads.

Database Services

- Amazon RDS- This Database AWS service is easy to set up, operate, and scale a relational database in the cloud.
- Amazon DynamoDB- It is a fast, fully managed NoSQL database service. It is a simple service which allow cost-effective storage and retrieval of data. It also allows you to serve any level of request traffic.
- Amazon ElastiCache- It is a web service which makes it easy to deploy, operate, and scale an in-memory cache in the cloud.
- Neptune- It is a fast, reliable and scalable graph database service.

- Amazon RedShift- It is Amazon's data warehousing solution which you can use to perform complex OLAP queries.

Analytics

- Athena-This analytics service allows perm SQL queries on your S3 bucket to find files.
- CloudSearch-You should use this AWS service to create a fully managed search engine for your website.
- ElasticSearch-It is similar to CloudSearch. However, it offers more features like application monitoring.
- Kinesis-This AWS analytics service helps you to stream and analyzing real-time data at massive scale.
- QuickSight-It is a business analytics tool. It helps you to create visualizations in a dashboard for data in Amazon Web Services. For example, S3, DynamoDB, etc.
- EMR (Elastic Map Reduce)-This AWS analytics service mainly used for big data processing like Spark, Splunk, Hadoop, etc.
- Data Pipeline-Allows you to move data from one place to another. For example from DynamoDB to S3.

Management Services

- CloudWatch-Cloud watch helps you to monitor AWS environments like EC2, RDS instances, and CPU utilization. It also triggers alarms depends on various metrics.
- CloudFormation-It is a way of turning infrastructure into the cloud. You can use templates for providing a whole production environment in minutes.
- CloudTrail-It offers an easy method of auditing AWS resources. It helps you to log all changes.
- OpsWorks-The service allows you to automated Chef/Puppet deployments on AWS environment.
- Config-This AWS service monitors your environment. The tool sends alerts about changes when you break certain defined configurations.
- Service Catalog-This service helps large enterprises to authorize which services user will be used and which won't.
- AWS Auto Scaling-The service allows you to automatically scale your resources up and down based on given CloudWatch metrics.

- Systems Manager-This AWS service allows you to group your resources. It allows you to identify issues and act on them.
- Managed Services-It offers management of your AWS infrastructure which allows you to focus on your applications.

Internet of Things

- IoT Core-It is a managed cloud AWS service. The service allows connected devices?like cars, light bulbs, sensor grids, to securely interact with cloud applications and other devices.
- IoT Device Management-It allows you to manage your IoT devices at any scale.
- IoT Analytics-This AWS IOT service is helpful to perform analysis on data collected by your IoT devices.
- Amazon FreeRTOS-This real-time operating system for microcontrollers helps you to connect IoT devices in the local server or into the cloud.

Application Services

- Step Functions-It is a way of visualizing what's going inside your application and what different microservices it is using.
- SWF (Simple Workflow Service)-The service helps you to coordinate both automated tasks and human-led tasks.
- SNS (Simple Notification Service)-You can use this service to send you notifications in the form of email and SMS based on given AWS services.
- SQS (Simple Queue Service)-Use this AWS service to decouple your applications. It is a pull-based service.
- Elastic Transcoder-This AWS service tool helps you to changes a video's format and resolution to support various devices like tablets, smartphones, and laptops of different resolutions.

Deployment and Management

- AWS CloudTrail: The services records AWS API calls and send backlog files to you.
- Amazon CloudWatch: The tools monitor AWS resources like Amazon EC2 and Amazon RDS DB Instances. It also allows you to monitor custom metrics created by user's applications and services.

- AWS CloudHSM: This AWS service helps you meet corporate, regulatory, and contractual, compliance requirements for maintaining data security by using the Hardware Security Module(HSM) appliances inside the AWS environment.

Developer Tools

- CodeStar-Codestar is a cloud-based service for creating, managing, and working with various software development projects on AWS.
- CodeCommit-It is AWS's version control service which allows you to store your code and other assets privately in the cloud.
- CodeBuild-This Amazon developer service help you to automates the process of building and compiling your code.
- CodeDeploy-It is a way of deploying your code in EC2 instances automatically.
- CodePipeline-It helps you create a deployment pipeline like testing, building, testing, authentication, deployment on development and production environments.
- Cloud9-It is an Integrated Development Environment for writing, running, and debugging code in the cloud.

Mobile Services

- Mobile Hub-Allows you to add, configure and design features for mobile apps.
- Cognito-Allows users to signup using his or her social identity.
- Device Farm- Device farm helps you to improve the quality of apps by quickly testing hundreds of mobile devices.
- AWS AppSync-It is a fully managed GraphQL service that offers real-time data synchronization and offline programming features.

Business Productivity

- Alexa for Business-It empowers your organization with voice, using Alexa. It will help you to Allows you to build custom voice skills for your organization.
- Chime-Can be used for online meeting and video conferencing.
- WorkDocs- Helps to store documents in the cloud
- WorkMail- Allows you to send and receive business emails.

Desktop & App Streaming

- WorkSpaces-Workspace is a VDI (Virtual Desktop Infrastructure). It allows you to use remote desktops in the cloud.
- AppStream- A way of streaming desktop applications to your users in the web browser. For example, using MS Word in Google Chrome.

Artificial Intelligence

- Lex-Lex tool helps you to build chatbots quickly.
- Polly-It is AWS's text-to-speech service allows you to create audio versions of your notes.
- Rekognition - It is AWS's face recognition service. This AWS service helps you to recognize faces and object in images and videos.
- SageMaker-Sagemaker allows you to build, train, and deploy machine learning models at any scale.
- Transcribe- It is AWS's speech-to-text service that offers high-quality and affordable transcriptions.
- Translate- It is a very similar tool to Google Translate which allows you to translate text in one language to another.

AR & VR (Augmented Reality & Virtual Reality)

Sumerian-Sumerian is a set of tool for offering high-quality virtual reality (VR) experiences on the web. The service allows you to create interactive 3D scenes and publish it as a website for users to access.

Customer Engagement

- Amazon Connect- Amazon Connect allows you to create your customer care center in the cloud.
- Pinpoint- Pinpoint helps you to understand your users and engage with them.
- SES (Simple Email Service)- Helps you to send bulk emails to your customers at a relatively cost-effective price.

Game Development

- GameLift- It is a service which is managed by AWS. You can use this service to host dedicated game servers. It allows you to scale seamlessly without taking your game offline.

AWS Global Infrastructure

- AWS is a cloud computing platform which is globally available.
- Global infrastructure is a region around the world in which AWS is based. Global infrastructure is a bunch of high-level IT services which is shown below:
- AWS is available in 19 regions, and 57 availability zones in December 2018 and 5 more regions 15 more availability zones for 2019.

AWS Global Infrastructure Map

The AWS Cloud spans 84 Availability Zones within 26 geographic regions around the world, with announced plans for 24 more Availability Zones and 8 more AWS Regions in Australia, Canada, India, Israel, New Zealand, Spain, Switzerland, and United Arab Emirates (UAE).

26 Launched Regions Each with multiple Availability Zones (AZ's)	**84 Availability Zones**	**17 Local Zones** **24 Wavelength Zones** For ultralow latency applications	**8 Announced Regions** **32 Announced Local Zones**
2x More Regions With multiple AZ's than the next largest cloud provider	**245 Countries and Territories Served**	**108 Direct Connect Locations**	**310+ Points of Presence** 300+ Edge Locations and 13 Regional Edge Caches

Image 7: AWS Global Infrastructure
Reference:https://aws.amazon.com/about-aws/global-infrastructure/

The AWS Global Cloud Infrastructure is the most secure, extensive, and reliable cloud platform, offering over 200 fully featured services from data centers globally. Whether you need to deploy your application workloads across the globe in a single click, or you want to build and deploy specific applications closer to your end-users with single-digit millisecond latency, AWS provides you the cloud infrastructure where and when you need it.

With millions of active customers and tens of thousands of partners globally, AWS has the largest and most dynamic ecosystem. Customers across virtually every industry and of every size, including start-ups, enterprises, and public sector organizations, are running every imaginable use case on AWS.

The following are the components that make up the AWS infrastructure:

- Availability Zones
- Region
- Edge locations
- Regional Edge Caches

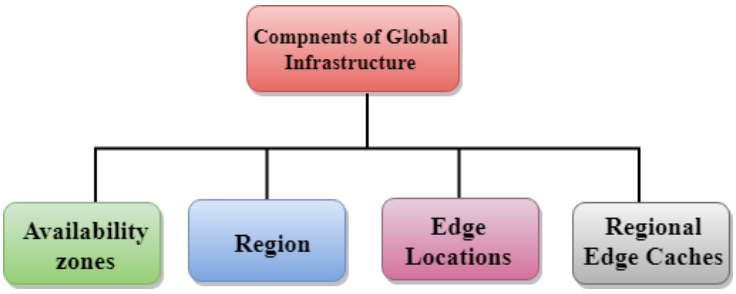


Image 8: Components of AWS Infrastructure
Reference:https://www.javatpoint.com/aws global-infrastructure

Availability zone as a Data Center

- An availability zone is a facility that can be somewhere in a country or in a city. Inside this facility, i.e., Data Centre, we can have multiple servers, switches, load balancing, firewalls. The things which interact with the cloud sits inside the data centers.

- An availability zone can be a several data centers, but if they are close together, they are counted as 1 availability zone.

Region

- A region is a geographical area. Each region consists of 2 more availability zones.
- A region is a collection of data centers which are completely isolated from other regions.
- A region consists of more than two availability zones connected to each other through links.
- Availability zones are connected through redundant and isolated metro fibers.

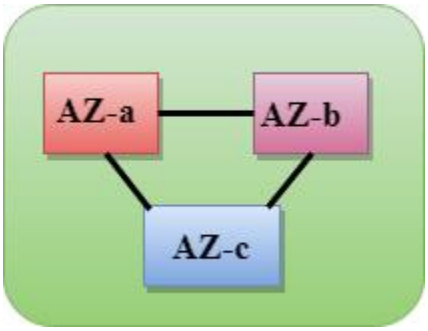


Image 9: Region
Reference:https://www.javatpoint.com/aws global-infrastructure

Edge Locations

- Edge locations are the endpoints for AWS used for caching content.
- Edge locations consist of CloudFront, Amazon's Content Delivery Network (CDN).
- Edge locations are more than regions. Currently, there are over 150 edge locations.
- Edge location is not a region but a small location that AWS have. It is used for caching the content.
- Edge locations are mainly located in most of the major cities to distribute the content to end users with reduced latency.
- For example, some user accesses your website from Singapore; then this request would be redirected to the edge location closest to Singapore where cached data can be read.

Regional Edge Cache

- AWS announced a new type of edge location in November 2016, known as a Regional Edge Cache.
- Regional Edge cache lies between CloudFront Origin servers and the edge locations.
- A regional edge cache has a large cache than an individual edge location.
- Data is removed from the cache at the edge location while the data is retained at the Regional Edge Caches.
- When the user requests the data, then data is no longer available at the edge location. Therefore, the edge location retrieves the cached data from the Regional edge cache instead of the Origin servers that have high latency.

Benefits

Security

Security at AWS starts with our core infrastructure. Custom-built for the cloud and designed to meet the most stringent security requirements in the world, our infrastructure is monitored 24/7 to help ensure the confidentiality, integrity, and availability of your data. All data flowing across the AWS global network that interconnects our datacenters and Regions is automatically encrypted at the physical layer before it leaves our secured facilities. You can build on the most secure global infrastructure, knowing you always control your data, including the ability to encrypt it, move it, and manage retention at any time.

Availability

AWS delivers the highest network availability of any cloud provider. Each region is fully isolated and comprised of multiple AZs, which are fully isolated partitions of our infrastructure. To better isolate any issues and achieve high availability, you can partition applications across multiple AZs in the same region. In addition, AWS control planes and the AWS management console are distributed across regions, and include regional API endpoints, which are designed to operate securely for at least 24 hours if isolated from the global control plane functions without requiring customers to access the region or its API endpoints via external networks during any isolation.

Performance

The AWS Global Infrastructure is built for performance. AWS Regions offer low latency, low packet loss, and high overall network quality. This is achieved with a fully redundant 100 GbE fiber network backbone, often providing many terabits of capacity between Regions. AWS Local

Zones and AWS Wavelength, with our telco providers, provide performance for applications that require single-digit millisecond latencies by delivering AWS infrastructure and services closer to end-users and 5G connected devices. Whatever your application needs, you can quickly spin up resources as you need them, deploying hundreds or even thousands of servers in minutes.

Global Footprint

AWS has the largest global infrastructure footprint of any provider, and this footprint is constantly increasing at a significant rate. When deploying your applications and workloads to the cloud, you have the flexibility in selecting a technology infrastructure that is closest to your primary target of users. You can run your workloads on the cloud that delivers the best support for the broadest set of applications, even those with the highest throughput and lowest latency requirements. And If your data lives off this planet, you can use AWS Ground Station, which provides satellite antennas in close proximity to AWS infrastructure Regions.

Scalability

The AWS Global Infrastructure enables companies to be extremely flexible and take advantage of the conceptually infinite scalability of the cloud. Customers used to over provision to ensure they had enough capacity to handle their business operations at the peak level of activity. Now, they can provision the amount of resources that they actually need, knowing they can instantly scale up or down along with the needs of their business, which also reduces cost and improves the customer's ability to meet their user's demands. Companies can quickly spin up resources as they need them, deploying hundreds or even thousands of servers in minutes.

Flexibility

The AWS Global Infrastructure gives you the flexibility of choosing how and where you want to run your workloads, and when you do you are using the same network, control plane, APIs, and AWS services. If you would like to run your applications globally you can choose from any of the AWS Regions and AZs. If you need to run your applications with single-digit millisecond latencies to mobile devices and end-users you can choose AWS Local Zones or AWS Wavelength. Or if you would like to run your applications on-premises you can choose AWS Outposts.

AWS Scalability

What is scalability?

Cloud scalability refers to the ability to increase or decrease IT resources (virtual machines, databases, networks) as needed to meet changing needs. Scalability is one of the main advantages of the cloud and the main driving force for its popularity in businesses.

Public cloud providers such as AWS (Amazon Web Services) already have all the infrastructure in place; in the past, when scaling had to be done using on-premises infrastructure, the process could take weeks or months and require capital investment.

Systems have four general areas that scalability can apply to:

- CPU
- Disk I/O
- Memory
- Network I/O

Scalability on AWS

AWS as a platform has scalability built-in. They offer many services and features that can help set up your application to scale up or down depending on the resource requirements.

Below you can find a list of services that are commonly used for horizontal or vertical scaling or both.

Horizontal scaling in AWS

To scale horizontally (scaling in or out), you add more resources like virtual machines to your system to spread out the workload across them. Horizontal scaling is especially important for companies that need high availability services with a requirement for minimal downtime.

A simple example of horizontal scaling in AWS Cloud is adding/removing Amazon EC2 instances from your application architecture behind Elastic Load Balancer. A simple example architecture is provided below.

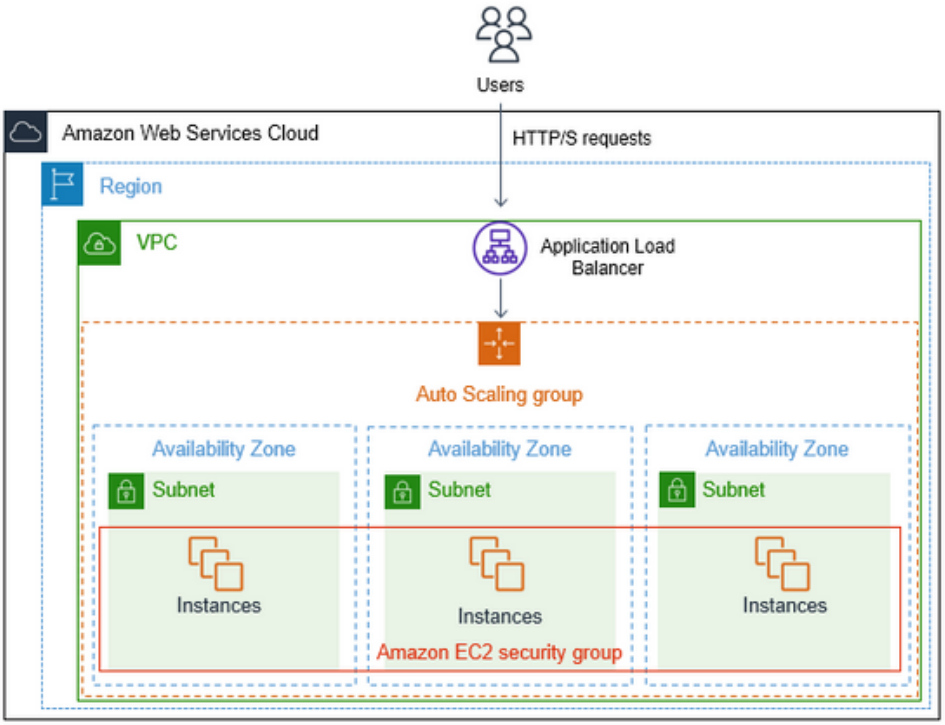


Image 10: Horizontal Scaling
Reference: https://www.stormit.cloud/post/

And here is a list of AWS services and features that can help you with horizontal scaling:

1. Regions & Availability Zones

Regions and Availability Zones enable you to scale applications horizontally across data centers and geographic regions to ensure resiliency and proximity to users.

An Availability Zone consists of one or more data centers in a geographic area, which are physically separated from each other in terms of power, network, and security. The best practice is to distribute the workload across multiple Availability Zones to reduce the risk of hardware or facility failure.

A Region is a geographic area that contains two or more Availability Zones. Scaling the application across multiple regions helps ensure the best experience for your users.

2. Elastic Load Balancer

This service is used to automatically distribute incoming application traffic across multiple targets (EC2 instances, Lambda functions, etc.)

This service is divided into the two of the most used types:

Application Load Balancer (ALB) – best suited for load balancing of HTTP and HTTPS traffic. They operate at Layer 7 and are application-aware. They do have some default limits set, some of which can be raised on request. The default limits include 1000 Targets per ALB, 50 listeners per ALB, 50 ALBs per region and 3,000 targets per region.

Network Load Balancer (NLB) – Network Load Balancers operate at Layer 4, the transport layer. They can load balance tens of millions of requests per second at very low latency. As for ALB, you can have 50 NLBs per region and 3000 target groups per region by default.

3. Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling enables fleets of EC2 instances to scale based on application traffic or demand.

It's defined as an Auto Scaling group by the launch template on the Elastic Load Balancer. The launch template defines the minimum and the maximum number of EC2 instances in the group, as well as the indicators that trigger the launch of new instances. These triggers can be based on instance health checks, CPU load, incoming or outgoing network traffic, or the number of load balancer requests per target.

4. AWS Auto Scaling

AWS Auto Scaling is a service in the AWS Cloud environment which enables you to configure scaling for selected AWS services that are part of your application in minutes. You can be sure that you'll always have enough resources/instances to handle your application load, no matter how greatly or suddenly traffic may spike.

5. AWS Elastic Beanstalk

Elastic Beanstalk enables you to create simple web applications that scale automatically without worrying about any underlying infrastructure such as Elastic Load Balancers, EC2 instances, and

databases. It supports web applications written in Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.

6. Amazon Elastic Container Service (ECS)

Amazon ECS is a fully managed container orchestration service. Containerized applications lend themselves very well to horizontal scaling. By default, ECS can manage 10,000 clusters per region, with 1,000 services per cluster.

Vertical scaling in AWS

Through vertical scaling (scaling up or down), you can increase or decrease the capacity of existing services/instances by upgrading the memory (RAM), storage, or processing power (CPU). Usually, this means that the expansion has an upper limit based on the capacity of the server or machine being expanded.

Every AWS service mentioned below supports vertical scaling, but you are also able to horizontally scale these services using Availability Zones and Regions.

1. Amazon EC2 Instance

An EC2 instance is a virtual server in the AWS Cloud. For any service in the AWS environment, the EC2 instances are behind it as virtual machines. AWS has a huge range of EC2 instances for different workload types.

The maximum for vertical scaling of CPU is the Compute Optimized EC2 instance (c5d.metal) that has 96 vCPUs while the largest Memory Optimized instance (u-24tb1.metal) has 24TB of memory.

2. AWS Lambda

Lambda is a computing service that lets you run code without provisioning or managing servers. Just upload your code and start a Lambda function. Lambda then scales automatically. Every time an event notification is received for your function, AWS Lambda quickly locates free capacity within its compute fleet and runs your code.

3. EBS Volumes

EBS Volumes are the hard disk drive volumes that can be attached to EC2 instances. A single General Purpose (GP2) EBS volume can scale to 16TB and 10,000 IOPS. A Provisioned IOPS EBS volume can scale to 16TB and 64,000 IOPS.

4. EFS Volumes

EFS or Elastic File System is a shared storage volume that can be mounted via NFS to an operating system, enabling multiple instances to see the same disk volume. With EFS Storage you only pay for what you consume, but an EFS volume is virtually infinitely scalable.

5. Amazon S3

S3 or Simple Storage Service is an AWS object storage solution for documents, videos, audio files, etc. S3 has almost unlimited scalability, and you only pay for the content stored in it. The size of a single object can be up to 5TB, and there is no limit to the number of objects that can be stored in the bucket.

6. Amazon DynamoDB

DynamoDB is a fast, highly reliable, and cost-effective NoSQL database service. DynamoDB delivers automatic scaling of throughput and storage based on your previously set capacity by monitoring the performance usage of your application.

7. Amazon RDS

Amazon Relational Database Service (Amazon RDS) makes it easy to operate and scale a relational database in the AWS Cloud. With RDS Storage Auto Scaling, you simply set your desired maximum storage limit, and this feature takes care of the rest.

AWS Elasticity

In AWS, the process of getting the resources dynamically when you actually require them and then release the resources when you are done and do not need them is known as elasticity. In another way, growing or shrinking the resources dynamically when needed is known as Elasticity.

Increasing or decreasing the number of resources automatically based on the need is known as Elasticity in AWS.

Elasticity is one of the best properties that increase productivity.

The ability to acquire resources as you need them and release resources when you no longer need them. In the cloud, you want to do this automatically.

Most people, when thinking of cloud computing, think of the ease with which they can procure resources when needed. This is only one aspect to elasticity. The other aspect is to contract when they no longer need resources. Scale out and scale in. Scale up and scale down. Some services do this as part of their service: Amazon S3, Amazon SQS, Amazon SNS, Amazon SES, Amazon Aurora, etc. Some require vertical scaling, like Amazon RDS. Others integrate with AWS Auto Scaling, like Amazon EC2, Amazon ECS, AWS Fargate, Amazon EKS, and Amazon DynamoDB. Amazon Aurora Serverless and Amazon Athena also qualify as elastic.

Implement elasticity

- Identify the workloads that have variable load.
- Identify the workload load range. That is, is there enough variability to warrant adding or removing resources?
- Identify the application limitations (sessions, long initialization, licensing, etc.) that may limit elasticity.
- Identify if the increase in demand can be met by automatic scaling, or if it needs to be in place before (for events, launches, etc.).
- Identify applications that can use Amazon Athena or Amazon Aurora Serverless
- Implement elasticity using AWS Auto Scaling or Application Auto Scaling for the aspects of your service that are not elastic by design.
- Test elasticity both up and down, ensuring it will meet requirements for load variance.
- Iterate on implementation and testing until you can meet requirements. You may want to investigate golden Amazon Machine Images, docker containers, etc. to speed launch.

Types of Elasticity AWS

Elasticity is of two types in AWS and those are as below.

- Volume-based elasticity
- Time-based elasticity

Volume-based elasticity

Volume-based elasticity is about matching the scale and adding the resources based on the demand. You can use Amazon EC2 Spot Fleets that can help you to start the instances or terminating the instances when not needed. Not only that, but you can also use the Amazon EMR clusters to programmatically scale out and scale in the nodes in a cluster.

Time-based elasticity

Time-based elasticity is all about releasing the resources when you no longer need them. It is always a good idea to automate the time-based elasticity process.

You can use the AWS instance scheduler to create an automatic schedule for starting and stopping for the EC2 instance. You can also use the Amazon EC2 APIs to terminate any of the instances programmatically. Not only that, but you can also use the AWS Lambda functions to shut down the instances when they are no longer needed.

Amazon CloudWatch is one of the excellent tools that can help to automate the process here.

AWS Fault Tolerance

Fault Tolerance simply means a system's ability to continue operating uninterrupted despite the failure of one or more of its components. This is true whether it is a computer system, a cloud cluster, a network, or something else.

By using different AWS Fault Tolerance services, you can build a fault-tolerant system that will be robust against any failure and alerts when there are problems. AWS Fault Tolerance services allow you to set up a fault-tolerant system with little human supervision and upfront financial commitment.

What is the AWS Fault Tolerance Architecture?

When running a machine, faults are inevitable. Faults can occur due to network outage, system crash, running out of memory, malware, etc.

AWS Fault Tolerance architecture provides:

- A vast amount of IT infrastructure.
- Computing Instances, and

- Storage that you can use to create fault-tolerant systems.

AWS systems are self-reliant to failures and can automatically recover from the failures.A single service is not fault-tolerant; you have to use various services to make the application fault-tolerant.

AWS Fault Tolerance Components

AWS provides several components or services that can create fault-tolerant systems. Some of these AWS Fault Tolerance components are:

- Auto Scaling
- Elastic Load Balancing
- Elastic IPs
- Reserved Instances
- Elastic Block Store
- Relational Database Service
- Simple Storage Service
- Simple Queue Service
- Route 53

1. Auto Scaling

Auto-Scaling is the concept of automatically scaling up the machines (compute resources) as demanded by load, thereby safeguarding the machines from failures. Autoscaling is a powerful option that can be easily applied to your applications.

Auto-Scaling allows you to set rules that will automatically scale up or down your compute resources. The rules can be:

- Launch server instances when the CPU threshold increases beyond a certain point. The AWS CloudWatch component can obtain the CPU metrics.
- When the number of servers is above (or below) a certain number, then launch (or terminate) the servers.

Auto Scaling generally follows the rule of N+1 redundancy. N+1 redundancy rule is a popular strategy for making instances always available. N+1 dictates that there should be N+1 resources available when N resources are sufficient to handle the anticipated load.

Auto Scaling will automatically detect the failure of instances and launch replacement instances.

2. Elastic Load Balancing

Elastic Load Balancer is another AWS product that distributes several servers' incoming traffic (EC2 instance).

The Elastic Load balancer uses a hostname on which the incoming traffic arrives, and then it redistributes that traffic to the pool of Amazon instances.

Elastic Load Balancing can detect unhealthy instances within its pool of Amazon EC2 instances and automatically reroutes traffic to healthy instances.

Autoscaling and Elastic Load Balancing is a great combination to create a fault-tolerant system as ELB reroutes traffic to healthy clusters. In contrast, Auto-Scaling ensures that there are always healthy clusters available.

3. Elastic IPs

Elastic IP Addresses are the variable public IPs and can be mapped to any EC2 instances within the particular EC2 region.

These Elastic Addresses are associated with an AWS account and are not specific to instances. Hence, EIPs make a significant contribution to designing fault-tolerant applications.

In a short period, an elastic IP address can be removed from a failing instance and mapped to a replacement instance.

4. Reserved Instances

Reserve Instances are reserved for future failover to ensure that an instance is always available in case of a shortage of resources on the AWS side.

AWS has massive hardware resources available, but these resources are finite. The best way to create a fault-tolerant system is to reserve such instances beforehand to avoid last-minute unavailability.

With Reserved Instances, you reserve computing capacity in the Amazon Web Services cloud. Doing this can bring lower prices. More significantly, it will increase your chances of receiving the computing capacity you require in the context of fault tolerance.

5. Elastic Block Store

Amazon Elastic Block Store (EBS) is the block storage volume used with Amazon EC2 instances. EBS persists the data outside the compute instances and persists the data independently from the life of the compute instances.

Amazon EBS volumes are hard drives that may be added to a running Amazon EC2 instance. Amazon EBS and Amazon EC2 machines are used in conjunction with one another when building fault-tolerant systems.

Amazon EBS stores the data outside the EC2 instances. Hence, any failure to EC2 instances can not impact the data. The EBS can be attached to any other running instances. EBS creates the backup of the data by using the technique called Snapshot. These snapshots can be stored in Amazon S3, another Simple Storage Service that is highly available and fault-tolerant.

6. Relational Database Service

RDS (Amazon Relational Database Service) is another AWS service that offers the framework for running relational databases in the cloud. Amazon RDS offers several features to enhance the reliability of the database in building fault-tolerant systems.

Amazon RDS creates a backup of your database and transaction log time-to-time to provide data recovery in case of failure. The backups can help to recover any data loss suffered from any failures. These database backups will be stored by Amazon RDS unless deleted.

7. Simple Storage Service

Amazon S3, or Amazon Simple Storage Service, is a simple online service that delivers exceptionally durable, fault-tolerant data storage. Amazon S3 stores the data on multiple regions and multiple devices so that in case of failure of any data center, the data is still accessible. Amazon Web Services is responsible for maintaining availability and fault tolerance within all the applications.

Amazon S3 has a versioning feature that allows you to track and retain any previous versions of data/objects stored and protects against any unintentional modifications done to the data. Amazon S3 is an essential part of creating a fault-tolerant system within AWS.

8. Simple Queue Service

SQS (Amazon Simple Queue Service) is a fault-tolerant and distributed messaging system that serves as the foundation for any fault-tolerant application. It is mainly used to send messages in case of failures and any abrupt things happening on applications. Amazon SQS stores the messages in Queue and retains them for up to four days unless read/deleted by the application.

9. Route 53

Amazon Route 53 is a highly available and scalable DNS web service from the stack of Amazon Web Services. It is designed to provide a reliable and cost-effective way to route end users to Internet applications by resolving the Domain name with the numeric IP address that allows computers to interact with each other.

You can configure DNS health checks using Amazon Route 53, then use Route 53 Application Recovery Controller to continually monitor and govern your applications' capacity to recover from failures.

AWS Reliability

The Reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it's expected to. This includes the ability to operate and test the workload through its total lifecycle

Design Principles

There are five design principles for reliability in the cloud:

- **Automatically recover from failure**: By monitoring a workload for key performance indicators (KPIs), you can trigger automation when a threshold is breached. These KPIs should be a measure of business value, not of the technical aspects of the operation of the service. This allows for automatic notification and tracking of failures, and for automated recovery processes that work around or repair the failure. With more sophisticated automation, it's possible to anticipate and remediate failures before they occur.

Test recovery procedures: In an on-premises environment, testing is often conducted to prove that the workload works in a particular scenario. Testing is not typically used to validate recovery strategies. In the cloud, you can test how your workload fails, and you can validate your recovery procedures. You can use automation to simulate different failures or to recreate scenarios that led to failures before. This approach exposes failure pathways that you can test and fix before a real failure scenario occurs, thus reducing risk.

Scale horizontally to increase aggregate workload availability: Replace one large resource with multiple small resources to reduce the impact of a single failure on the overall workload. Distribute requests across multiple, smaller resources to ensure that they don't share a common point of failure.

Stop guessing capacity: A common cause of failure in on-premises workloads is resource saturation, when the demands placed on a workload exceed the capacity of that workload (this is often the objective of denial-of-service attacks). In the cloud, you can monitor demand and workload utilization, and automate the addition or removal of resources to maintain the optimal level to satisfy demand without over- or under-provisioning. There are still limits, but some quotas can be controlled and others can be managed (see Manage Service Quotas and Constraints).

Manage change in automation: Changes to your infrastructure should be made using automation. The changes that need to be managed include changes to the automation, which then can be tracked and reviewed.

There are four best practice areas for reliability in the cloud:

- Foundations
- Workload Architecture
- Change Management
- Failure Management

To achieve reliability, you must start with the foundations — an environment where service quotas and network topology accommodate the workload. The workload architecture of the distributed system must be designed to prevent and mitigate failures. The workload must handle changes in demand or requirements, and it must be designed to detect failure and automatically heal itself.

Foundations

Foundational requirements are those whose scope extends beyond a single workload or project. Before architecting any system, foundational requirements that influence reliability should be in place. For example, you must have sufficient network bandwidth to your data center. With AWS, most of these foundational requirements are already incorporated or can be addressed as needed. The cloud is designed to be nearly limitless, so it's the responsibility of AWS to satisfy the requirement for sufficient networking and compute capacity, leaving you free to change resource size and allocations on demand.

Workload Architecture

A reliable workload starts with upfront design decisions for both software and infrastructure. Your architecture choices will impact your workload behavior across all five Well-Architected pillars. For reliability, there are specific patterns you must follow. With AWS, workload developers have their choice of languages and technologies to use. AWS SDKs take the complexity out of coding by providing language-specific APIs for AWS services.

Change Management

Changes to your workload or its environment must be anticipated and accommodated to achieve reliable operation of the workload. Changes include those imposed on your workload, such as spikes in demand, as well as those from within, such as feature deployments and security patches. Using AWS, you can monitor the behavior of a workload and automate the response to KPIs. For example, your workload can add additional servers as a workload gains more users. You can control who has permission to make workload changes and audit the history of these changes.

Failure Management

In any system of reasonable complexity, it is expected that failures will occur. Reliability requires that your workload be aware of failures as they occur and take action to avoid impact on availability. Workloads must be able to both withstand failures and automatically repair issues. With AWS, you can take advantage of automation to react to monitoring data. For example, when a particular metric crosses a threshold, you can trigger an automated action to remedy the problem. Also, rather than trying to diagnose and fix a failed resource that is part of your production environment, you can replace it with a new one and carry out the analysis on the failed resource out of band. Since the cloud enables you to stand up temporary versions of a whole system at low cost, you can use automated testing to verify full recovery processes.

AWS Durability

A system that is durable is able to perform its responsibilities over time, even when unexpected events may occur. For example, a durable storage system will reliably store data without data loss

AWS Storage Services: AWS offers a wide range of storage services that can be provisioned depending on your project requirements and use case. AWS storage services have different provisions for highly confidential data, frequently accessed data, and the not so frequently accessed data. You can choose from various storage types namely, object storage, file storage, block storage services, backups, and data migration options. All of which fall under the AWS Storage Services list.

AWS Simple Storage Service (S3): From the aforementioned list, S3, is the object storage service provided by AWS. It is probably the most commonly used, go-to storage service for AWS users given the features like extremely high availability, security, and simple connection to other AWS Services. AWS S3 can be used by people with all kinds of use cases like mobile/web applications, big data, machine learning and many more.

Image 11: Amazon S3
Reference: https://miro.medium.com/max/1280/1*B9CIOrxdROHvtdmouQA1_A.png

AWS S3 Terminology:

- **Bucket**: Data, in S3, is stored in containers called buckets.
  Each bucket will have its own set of policies and configuration. This enables users to have more control over their data. Bucket Names must be unique. Can be thought of as a parent folder of data. There is a limit of 100 buckets per AWS accounts. But it can be increased if requested from AWS support.
- **Bucket Owner**: The person or organization that owns a particular bucket is its bucket owner.
- **Import/Export Station**: A machine that uploads or downloads data to/from S3.

- **Key**: Key, in S3, is a unique identifier for an object in a bucket.

For example, in a bucket 'ABC' your GFG.java file is stored at javaPrograms/GFG.java then 'javaPrograms/GFG.java' is your object key for GFG.java.

It is important to note that 'bucketName+key' is unique for all objects.

This also means that there can be only one object for a key in a bucket. If you upload 2 files with the same key. The file uploaded latest will overwrite the previously contained file.

- **Versioning**: Versioning means to always keep a record of previously uploaded files in S3. Points to note:

Versioning is not enabled by default. Once enabled, it is enabled for all objects in a bucket.

Versioning keeps all the copies of your file, so, it adds cost for storing multiple copies of your data. For example, 10 copies of a file of size 1GB will have you charged for using 10GBs for S3 space. Versioning is helpful to prevent unintended overwrites and deletions.

Note that objects with the same key can be stored in a bucket if versioning is enabled (since they have a unique version ID).

- **null Object**: Version ID for objects in a bucket where versioning is suspended is null. Such objects may be referred to as null objects.

For buckets with versioning enabled, each version of a file has a specific version ID.

- **Object**: Fundamental entity type stored in AWS S3.
- **Access Control Lists (ACL)**: A document for verifying the access to S3 buckets from outside your AWS account. Each bucket has its own ACL.
- **Bucket Policies**: A document for verifying the access to S3 buckets from within your AWS account, this controls which services and users have what kind of access to your S3 bucket. Each bucket has its own Bucket Policies.
- **Lifecycle Rules**: This is a cost-saving practice that can move your files to AWS Glacier (The AWS Data Archive Service) or to some other S3 storage class for cheaper storage of old data or completely delete the data after the specified time.

Features of AWS S3:

- **Durability**: AWS claims Amazon S3 to have a 99.999999999% of durability (11 9's). This means the possibility of losing your data stored on S3 is one in a billion.
- **Availability**: AWS ensures that the up-time of AWS S3 is 99.99% for standard access. Note that availability is related to being able to access data and durability is related to losing data altogether.

Server-Side-Encryption (SSE): AWS S3 supports three types of SSE models:

SSE-S3: AWS S3 manages encryption keys.

SSE-C: The customer manages encryption keys.

SSE-KMS: The AWS Key Management Service (KMS) manages the encryption keys.

- **File Size support**: AWS S3 can hold files of size ranging from 0 bytes to 5 terabytes. A 5TB limit on file size should not be a blocker for most of the applications in the world.
- **Infinite storage space**: Theoretically AWS S3 is supposed to have infinite storage space. This makes S3 infinitely scalable for all kinds of use cases.
- **Pay as you use**: The users are charged according to the S3 storage they hold.
- **AWS-S3** is region-specific.

S3 storage classes:

AWS S3 provides multiple storage types that offer different performance and features and different cost structure.

- **Standard**: Suitable for frequently accessed data, that needs to be highly available and durable.
- **Standard Infrequent Access (Standard IA)**: This is a cheaper data-storage class and as the name suggests, this class is best suited for storing infrequently accessed data like log files or data archives. Note that there may be a per GB data retrieval fee associated with Standard IA class.
- **Intelligent Tiering**: This service class classifies your files automatically into frequently accessed and infrequently accessed and stores the infrequently accessed data in infrequent access storage to save costs. This is useful for unpredictable data access to an S3 bucket.

- **One Zone Infrequent Access (One Zone IA)**: All the files on your S3 have their copies stored in a minimum of 3 Availability Zones. One Zone IA stores this data in a single availability zone. It is only recommended to use this storage class for infrequently accessed, non-essential data. There may be a per GB cost for data retrieval.
- **Reduced Redundancy Storage (RRS)**: All the other S3 classes ensure the durability of 99.999999999%. RRS only ensures a 99.99% durability. AWS no longer recommends RRS due to its less durability. However, it can be used to store non-essential data.

What are the Core cloud computing concepts?

Cloud Computing technology is the future of online data across the world, should you cover up with core cloud concepts.

Advances in computing facilities began in the 1960s with the introduction of mainframes. Every computing has one or more problems, so cloud computing technology has been introduced, keeping this in mind. The roots of cloud computing are linked to older technologies. Some old technologies have similarities with it, such as distributed computing, autonomic computing, hardware virtualization, and Internet technologies. Cloud computing can be described with two models, one is the service model, and the other is the deployment model.

Along with providing many important services, the main role of cloud management is to provide resources. While there are many benefits to cloud computing, there are also some challenges in adapting to public clouds due to the reliance on infrastructure, which has been made possible by the collaboration of many businesses. In this article, we present core concepts of cloud computing, highlighting its number of core ideas, service models, their benefits, and security issues related to this technology. Continue reading with us, here provides detailed knowledge about cloud computing and to understand core cloud concepts in this field.

What is compute?

In cloud computing, the term “compute” describes concepts and objects related to software computation. It is a generic term used to reference processing power, memory, networking, storage, and other resources required for the computational success of any program.

For example, applications that run machine learning algorithms or 3D graphics rendering functions require many gigs of RAM and multiple CPUs to execute successfully. In this case, the CPUs, RAM, and Graphic Processing Units required will be called compute resources, and the applications would be compute-intensive applications.

What are compute resources?

Compute resources are measurable quantities of compute power that can be requested, allocated, and consumed for computing activities. Some examples of compute resources include:

CPU

The central processing unit (CPU) is the brain of any computer. CPU is measured in units called millicores. Application developers can specify how many allocated CPUs are required for running their application and to process data.

Memory

Memory is measured in bytes. Applications can make memory requests that are needed to run efficiently.

If applications are running on a single physical device, they have limited access to the compute resources of that device. But if applications run on the cloud, they can simultaneously access more processing resources from many physical devices. Let's take a closer look at this.

Compute Services

Your server is the base of your infrastructure. Depending on your needs, you have various options, or you can mix it up if that's what your environment requires. Check out the following table for a summary of your compute options.

Virtual servers

Option	Description
Virtual Servers for VPC	Scalable virtual servers for VPC that are purchased with cores and memory allocations.
Hyper Protect Virtual Servers	Hyper Protect Virtual Servers is an IBM Cloud™ service that provides highly secure virtual servers that can run Linux applications and containerized workloads.
Power Systems Virtual Server	A Power Systems Virtual Server integrates your AIX and IBM i capabilities into the IBM Cloud™ experience.

Table 1 - Compute options - Virtual Servers

References: https://cloud.ibm.com/docs/cloud-infrastructure?topic=cloud-infrastructure-compute

Networking services

What is cloud networking?

Cloud networking is a type of IT infrastructure in which some or all of an organization's network capabilities and resources are hosted in a public or private cloud platform, managed in-house or by a service provider, and available on demand.

Companies can either use on-premises cloud networking resources to build a private cloud network or use cloud-based networking resources in the public cloud, or a hybrid cloud combination of both. These network resources can include virtual routers, firewalls, and bandwidth and network management software, with other tools and functions available as required.

Why cloud networking?

Businesses today turn to the cloud to drive agility, deliver differentiation, accelerate time-to-market, and increase scale. The cloud model has become the standard approach to build and deliver applications for the modern enterprise.

Cloud networking has also played a critical role in the way organizations address their growing infrastructure needs, regional expansions, and redundancy plans. Many organizations are adopting a multi-data center strategy and leveraging multiple clouds from multiple cloud service providers (CSPs).

Benefits of cloud networking

Most organizations have become a patchwork of on-premises technologies, public cloud services, legacy applications and systems, and emerging technologies — a complex situation that contributes to a weak security posture and results in inadequate governance, visibility, and manageability across fragmented networks.

What is a virtual private cloud (VPC)?

A VPC is a public cloud offering that lets an enterprise establish its own private cloud-like computing environment on shared public cloud infrastructure. A VPC gives an enterprise the ability to define and control a virtual network that is logically isolated from all other public cloud tenants, creating a private, secure place on the public cloud.

Imagine that a cloud provider's infrastructure is a residential apartment building with multiple families living inside. Being a public cloud tenant is akin to sharing an apartment with a few roommates. In contrast, having a VPC is like having your own private condominium—no one else has the key, and no one can enter the space without your permission.

Features

VPCs are a "best of both worlds" approach to cloud computing. They give customers many of the advantages of private clouds, while leveraging public cloud resources and savings. The following are some key features of the VPC model:

- **Agility:** Control the size of your virtual network and deploy cloud resources whenever your business needs them. You can scale these resources dynamically and in real-time.
- **Availability:** Redundant resources and highly fault-tolerant availability zone architectures mean your applications and workloads are highly available.
- **Security:** Because the VPC is a logically isolated network, your data and applications won't share space or mix with those of the cloud provider's other customers. You have full control over how resources and workloads are accessed, and by whom.
- **Affordability:** VPC customers can take advantage of the public cloud's cost-effectiveness, such as saving on hardware costs, labor times, and other resources.

Benefits

Each VPC's main features readily translate into a benefit to help your business achieve agility, increased innovation, and faster growth.

- Flexible business growth: Because cloud infrastructure resources—including virtual servers, storage, and networking—can be deployed dynamically, VPC customers can easily adapt to changes in business needs.
- Satisfied customers: In today's "always-on" digital business environments, customers expect uptime ratios of nearly 100%. The high availability of VPC environments enables reliable online experiences that build customer loyalty and increase trust in your brand.
- Reduced risk across the entire data lifecycle: VPCs enjoy high levels of security at the instance or subnet level, or both. This gives you peace of mind and further increases the trust of your customers.

- More resources to channel toward business innovation: With reduced costs and fewer demands on your internal IT team, you can focus your efforts on achieving key business goals and exercising core competencies.

Architecture

In a VPC, you can deploy cloud resources into your own isolated virtual network. These cloud resources—also known as logical instances—fall into three categories.

- **Compute:** Virtual server instances (VSIs, also known as virtual servers) are presented to the user as virtual CPUs (vCPUs) with a predetermined amount of computing power, memory, etc.
- **Storage:** VPC customers are typically allocated a certain block storage quota per account, with the ability to purchase more. It is akin to purchasing additional hard drive space. Recommendations for storage are based on the nature of your workload.
- **Networking:** You can deploy virtual versions of various networking functions into your virtual private cloud account to enable or restrict access to its resources. These include *public gateways*, which are deployed so that all or some areas of your VPC environment can be made available on the public-facing Internet; *load balancers*, which distribute traffic across multiple VSIs to optimize availability and performance; and *routers*, which direct traffic and enable communication between network segments. Direct or dedicated links enable rapid and secure communications between your on-premises enterprise IT environment or your private cloud and your VPC resources on public cloud.

Three-tier architecture in a VPC

The majority of today's applications are designed with a three-tier architecture comprised of the following interconnected tiers:

- The web or presentation tier, which takes requests from web browsers and presents information created by, or stored within, the other layers to end users.
- The application tier, which houses the business logic and is where most processing takes place.
- The database tier, comprised of database servers that store the data processed in the application tier.

To create a three-tier application architecture on a VPC, you assign each tier its own subnet, which will give it its own IP address range. Each layer is automatically assigned its own unique ACL (Access control lists).

VPC infrastructure

Within the infrastructure layer, you can build a virtual private cloud, which is a virtual network that is tied to your IBM Cloud account.

Option	Description
Getting started with IBM Cloud® Virtual Private Cloud	Provides cloud security and the ability to dynamically scale your virtual server instances.
Load balancer for VPC	Distributes traffic among multiple server instances within the same region of your VPC.
VPN for VPC	Securely connect your VPC to another private network. You can use VPN to set up an IPsec site-to-site tunnel between your VPC and your on-premises private network or another VPC.
DNS Services	Provides private DNS to Virtual Private Cloud (VPC) users.
Transit Gateway	Interconnect IBM Cloud Classic and Virtual Private Cloud (VPC) infrastructures worldwide, keeping traffic within the IBM Cloud network.
Direct Link 2.0	Seamlessly connect your on-premises resources to your cloud resources.

Table 2- Networking options – VPC

References - https://cloud.ibm.com/docs/cloud-infrastructure?topic=cloud-infrastructure-network

Cloud Storage Services

Cloud storage service is built to provide applications, services and organizations with access to offsite storage capacity that can be provisioned instantly, is flexible in scaling automatically at run time and is globally accessible.

Cloud storage service is provided, hosted and managed by the storage service provider (SSP), and it works on the combination of storage servers, which are designed on storage virtualization

architecture. This technique allows a single storage server to create multiple logical and virtual drives with scalable capacity and tight coupling. End users and applications access the logical storage by the online management interface or integrating vendor APIs with the application and are only billed for the storage capacity metered.

Cloud storage services are delivered in public and hybrid storage models.

VPC infrastructure

Option	Description
Block Storage for VPC	Persistent, high-performance data storage for virtual server instances in the IBM Cloud Virtual Private Cloud (VPC). The VPC infrastructure provides rapid scaling across multiple regions and zones, and extra performance and security.
IBM Cloud Object Storage	Distributed, multi-tenant Cloud object storage for data encrypted and dispersed across multiple geographic locations, accessed over HTTP by using a REST API. This service uses the distributed storage technologies that are provided by the IBM Cloud Object Storage System (formerly Cleversafe).

Table 3 -Storage options – VPC

References - https://cloud.ibm.com/docs/cloud-infrastructure?topic=cloud-infrastructure-storage

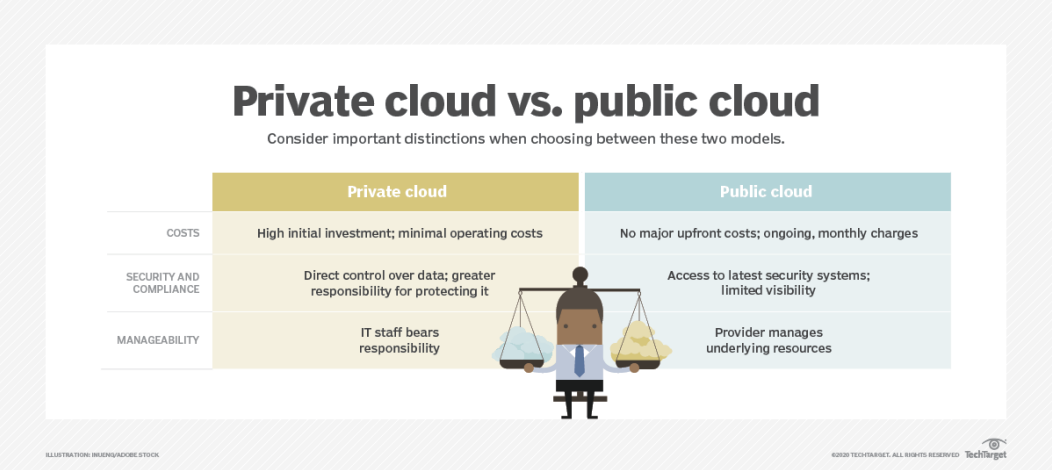
What is public cloud?

Public cloud is the classic cloud-computing model, where users can access a large pool of computing power over the internet (whether that is IaaS, PaaS, or SaaS). One of the significant benefits here is the ability to rapidly scale a service. The cloud-computing suppliers have vast amounts of computing power, which they share out between a large number of customers – the 'multi-tenant' architecture. Their huge scale means they have enough spare capacity that they can easily cope if any particular customer needs more resources, which is why it is often used for less-sensitive applications that demand a varying amount of resources.

What is private cloud?

Private cloud allows organizations to benefit from some of the advantages of public cloud – but without the concerns about relinquishing control over data and services, because it is tucked away

behind the corporate firewall. Companies can control exactly where their data is being held and can build the infrastructure in a way they want – largely for IaaS or PaaS projects – to give developers access to a pool of computing power that scales on-demand without putting security at risk. However, that additional security comes at a cost, as few companies will have the scale of AWS, Microsoft or Google, which means they will not be able to create the same economies of scale. Still, for companies that require additional security, private cloud might be a useful stepping stone, helping them to understand cloud services or rebuild internal applications for the cloud, before shifting them into the public cloud.



Reference: https://www.techtarget.com/searchnetworking/definition/cloud-networking

What is hybrid cloud?

Hybrid cloud is perhaps where everyone is in reality: a bit of this, a bit of that. Some data in the public cloud, some projects in private cloud, multiple vendors and different levels of cloud usage.

What cloud-computing services are available?

Cloud-computing services cover a vast range of options now, from the basics of storage, networking and processing power, through to natural language processing and artificial intelligence as well as standard office applications. Pretty much any service that doesn't require you to be physically close to the computer hardware that you are using can now be delivered via the cloud – even quantum computing.

Service

A Cloud Services manages that which type of service you access according to the client's requirement.

Cloud computing offers the following three type of services:

i. **Software as a Service (SaaS) –** It is also known as **cloud application services**. Mostly, SaaS applications run directly through the web browser means we do not require to download and install these applications. Some important example of SaaS is given below

   Example: Google Apps, Salesforce Dropbox, Slack, Hubspot, Cisco WebEx.

ii. **Platform as a Service (PaaS) –** It is also known as **cloud platform services**. It is quite similar to SaaS, but the difference is that PaaS provides a platform for software creation, but using SaaS, we can access software over the internet without the need of any platform.

   Example: Windows Azure, Force.com, Magento Commerce Cloud, OpenShift.

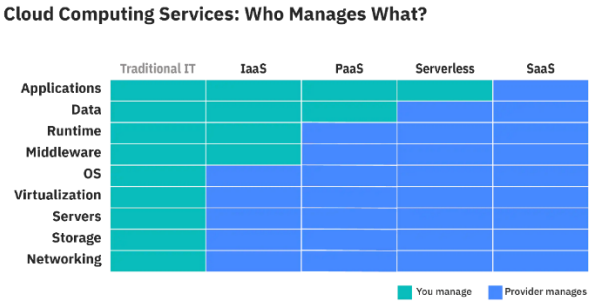
iii. **Infrastructure as a Service (IaaS) –** It is also known as **cloud infrastructure services**. It is responsible for managing applications data, middleware, and runtime environments.

   Example: Amazon Web Services (AWS) EC2, Google Compute Engine (GCE), Cisco Metapod.

iv. **Serverless computing -** Serverless computing (also called simply serverless) is a cloud computing model that offloads all the backend infrastructure management tasks–provisioning, scaling, scheduling, patching—to the cloud provider, freeing developers to focus all their time and effort on the code and business logic specific to their applications.

   What's more, serverless runs application code on a per-request basis only and scales the supporting infrastructure up and down automatically in response to the number of requests. With serverless, customers pay only for the resources being used when the application is running—they never pay for idle capacity.

FaaS, or Function-as-a-Service, is often confused with serverless computing when, in fact, it's a subset of serverless. FaaS allows developers to execute portions of application code (called functions) in response to specific events. Everything besides the code—physical hardware, virtual machine operating system, and web server software management—is provisioned automatically by the cloud service provider in real-time as the code executes and is spun back down once the execution completes. Billing starts when execution starts and stops when execution stops.



References - https://www.ibm.com/in-en/cloud/learn/cloud-computing

Software Needed to Support Cloud Computing

With the infrastructure in place, a prospective cloud provider must now settle on the software that will be the backbone of its cloud computing capabilities. If they are acting as an Infrastructure-as-a-Service (IaaS) provider, they will need to include more options than if they are acting as a Platform-as-a-Service (PaaS) provider. PaaS commonly provides a limited set of features that are focused on solving a narrower set of problems. Either way, there are core cloud computing technologies that companies must inevitably include to offer a credible product and service set, such as:

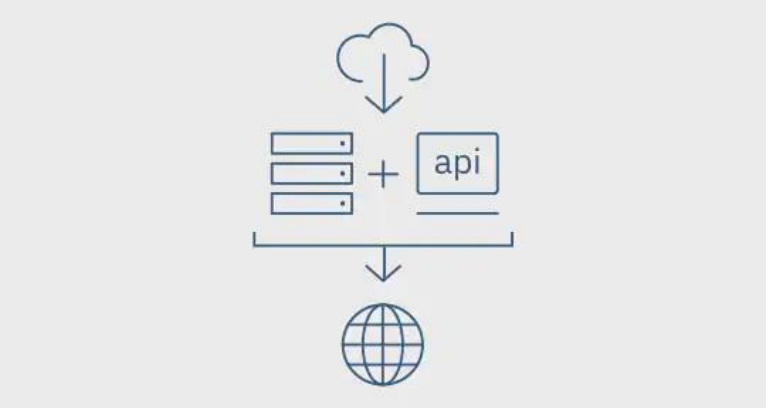
What is a cloud database?

A cloud database is a database service built and accessed through a cloud platform. It serves many of the same functions as a traditional database with the added flexibility of cloud computing. Users install software on a cloud infrastructure to implement the database.

Key features:

- A database service built and accessed through a cloud platform
- Enables enterprise users to host databases without buying dedicated hardware
- Can be managed by the user or offered as a service and managed by a provider
- Can support relational databases (including MySQL and PostgreSQL) and NoSQL databases (including MongoDB and Apache CouchDB)
- Accessed through a web interface or vendor-provided API

Why cloud databases?



Ease of access - Users can access cloud databases from virtually anywhere, using a vendor's API or web interface.

Scalability - Cloud databases can expand their storage capacities on run-time to accommodate changing needs. Organizations only pay for what they use.

Disaster recovery - In the event of a natural disaster, equipment failure or power outage, data is kept secure through backups on remote servers.

Considerations for cloud databases

- **Control options** - Users can opt for a virtual machine image managed like a traditional database or a provider's database as a service (DBaaS).
- **Database technology** - SQL databases are difficult to scale but very common. NoSQL databases scale more easily but do not work with some applications.
- **Security** - Most cloud database providers encrypt data and provide other security measures; organizations should research their options.
- **Maintenance** - When using a virtual machine image, one should ensure that IT staffers can maintain the underlying infrastructure.

For example, financial organizations are embracing the hybrid concept by using the database as a central repository for all their disparate data sources, and then delivering this financial data in JSON format. This data is then distributed to the database as a service and replicated to geographic regions across the world.

If a customer in Singapore has to wait more than 4 seconds for their mobile application data to be retrieved from a database in New Jersey, that customer is not likely to use that application again. Database-as-a-service can replicate and distribute immediately and offer near real-time access to data worldwide.

Cloud databases can collect, deliver, replicate, and push to the edge all your data using the new hybrid cloud concept. Users no longer have to deploy the dependent middleware to deliver database requests anywhere in the world. They can connect applications directly to their database.

VPC infrastructure

Option	Description
Block Storage for VPC	Persistent, high-performance data storage for virtual server instances in the IBM Cloud Virtual Private Cloud (VPC). The VPC infrastructure provides rapid scaling across multiple regions and zones, and extra performance and security.
IBM Cloud Object Storage	Distributed, multi-tenant Cloud object storage for data encrypted and dispersed across multiple geographic locations, accessed over HTTP by using a REST API. This service uses the distributed storage technologies that are provided by the IBM Cloud Object Storage System (formerly Cleversafe).

Table 4 - Storage options – VPC

Reference- https://cloud.ibm.com/docs/cloud-infrastructure?topic=cloud-infrastructure-storage

Components in Cloud Computing

Cloud computing components correspond to platforms such as front end, back end, and cloud-dependent delivery and the utilized network. So, a framework of cloud computing is broadly categorized as three specifically clients, distributed servers and datacentre.

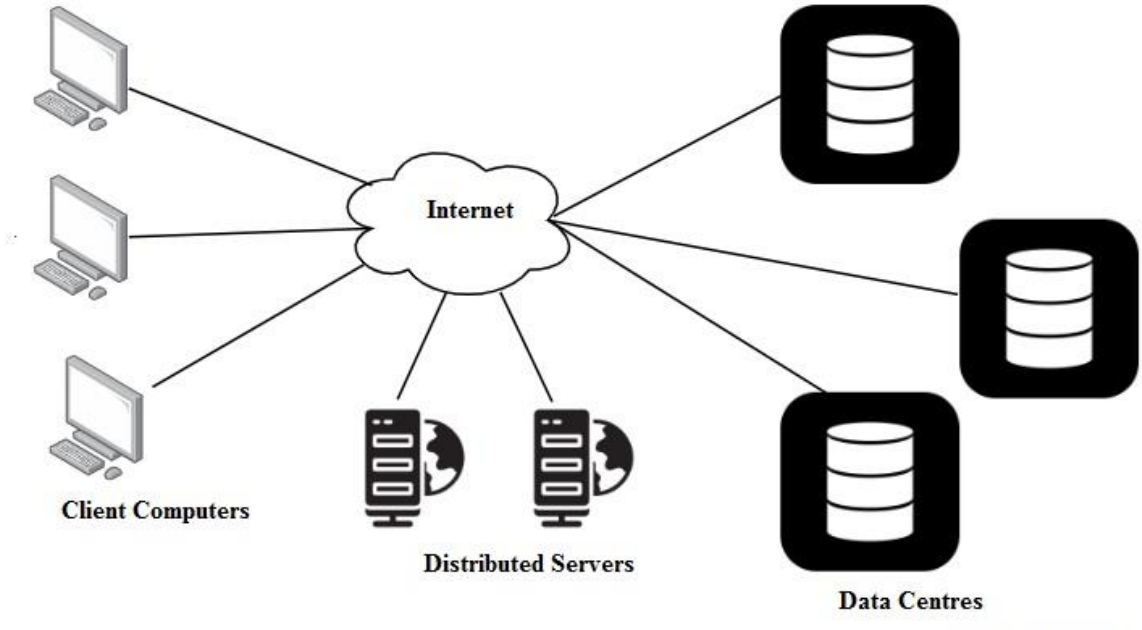


Image - **Components**

References: https://www.elprocus.com/wp-content/uploads/components-of-cloud-computing.jpg

For the operation of this computing, the following three components have a big hand and the responsibilities of these components can be elucidated clearly as below:

Clients

Clients in cloud computing are in general to the operation of Local Area Networks (LAN's). They are just the desktops where they have their place on desks. These might be also in the form of laptops, mobiles, tablets to enhance mobility. Clients hold the responsibility of interaction which pushes for the management of data on cloud servers.

Datacentre

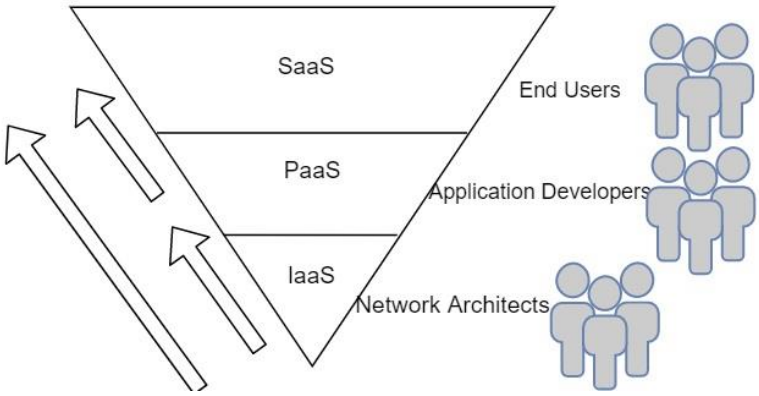
It is an array of servers that houses the subscribed application. Progressing the IT industry has brought the concept of virtualizing servers, where the software might be installed through the

utilization of various instances of virtual servers. This approach streamlines the process of managing dozens of virtual servers on multiple physical servers.

Distributed Servers

These are considered as a server where that is housed in the other location. So, the physical servers might not be housed in a similar location. Even the distributed server and the physical server appear to be in different locations, they perform as they are so close to each other.

While the other component is Cloud Applications, where it is defined as cloud computing in the form of software architecture. So, cloud applications serve as a service which operates both the hardware and software architecture.



Reference: https://www.elprocus.com/wp-content/uploads/cloud-computing-components.jpg

Data Security and Privacy

During the last 20 years, software systems had conquered an essential critical role in society, because of which now we are totally dependent upon the computerized systems. Besides, the use of computerized systems needs for a high level of security is the major obstacle in the world of computing. As a new technology, cloud computing is in use these days because of its features.

As data has moved into the cloud, security concerns are growing because it is strange to put your data on someone else's hard disk and use someone else's CPU. Therefore, security issues arise. The user must rely on the cloud computing provider's promise for data security and privacy. The main dimensions of security are confidentiality, integrity, privacy, and availability. Security is the

prevention of unauthorized access, prevention of altering the data, prevention of withholding the recourses, prevention of disclosure of information. That's because, the data is stored in the clouds and can be accessed anywhere in the world. So, the security and privacy of stored data can be compromised.

Cloud computing technology is controlled and managed by third parties. That's why it raises questions about the confidentiality of our business data.

These questions could be; whether data has been stored appropriately, whether data is secure, whether data will be kept secret throughout, whether data will remain the same. So, for this purpose, rights to access data should be defined clearly, Plus some data security techniques should be applied so that everyone could access data of his/her choice or which is relevant to his/her job.

What about cloud-computing security?

Many companies remain concerned about the security of cloud services, although breaches of security are rare. How secure you consider cloud computing to be will largely depend on how secure your existing systems are. In-house systems managed by a team with many other things to worry about are likely to be leakier than systems monitored by a cloud provider's engineers dedicated to protecting that infrastructure.

However, concerns do remain about security, especially for companies moving their data between many cloud services, which has led to growth in cloud security tools, which monitor data moving to and from the cloud and between cloud platforms. These tools can identify fraudulent use of data in the cloud, unauthorised downloads, and malware. There is a financial and performance impact, however: these tools can reduce the return on investment of the cloud by 5% to 10%, and impact performance by 5% to 15%. The country of origin of cloud services is also worrying some organisations.

Learning Outcome

In this section, we will read about:

- Getting started with IBM Cloud foundry, Benefits, features, use cases
- Get introduced to Storage
- PaaS Model services in IBM Cloud
- Hosting web application on PaaS

Getting started with IBM Cloud foundry

Cloud Foundry ensures that the build and deploy aspects of coding remain carefully coordinated with any attached services — resulting in quick, consistent and reliable iterating of applications.

As an industry-standard platform as a service (PaaS), Cloud Foundry ensures the fastest, easiest and most reliable deployment of cloud-native applications. IBM offers the Cloud Foundry PaaS in several hosting models, allowing you to customize your PaaS experience and balance a range of considerations — including price, deployment speed and security.

Benefits of Cloud Foundry

- **Choose your own language** - IBM Cloud® Foundry includes runtimes for Java, Node.js, PHP, Python, Ruby, Swift and Go; plus, Cloud Foundry community build packs are also available. Combined with DevOps services, the app runtimes enable a delivery pipeline that automates much of the iterative development process.
- **Fault tolerant** - Runtimes facilitate developing apps as stateless processes that quickly: start and stop, replicate if an instance fails, and duplicate if sustained or increased performance requires.
- **Extend apps with services** - Runtimes link IBM Cloud services to apps as endpoints, giving any instance of an app embedded knowledge of how to manage relevant calls and data. In fact, runtimes manage all linked resources this way: SDKs, APIs (whether made available as cloud services or exposed from within a traditional enterprise as custom services), and also apps themselves when used as resources by other apps.
- **Access control** - Fine grain assignment/dispensing of compute capacity to development teams.
- **Automatic placement** - Apps are automatically placed across multiple data-center PODs for maximum reliability.
- **Automatic health management** - Crashing apps will restart automatically.
- **Automatic routing** - Internet reachable routes are automatically created for your apps.
- **High availability** - Supports full high availability for extremely high app availability.

- **Automatic deployment scaling** - The Auto-Scaling for IBM Cloud service enables you to automatically increase or decrease the compute capacity of your app, to rapidly adjust to dynamic loading needs.

Features

- **Access control** - Your role determines the actions that you are authorized to undertake.
- **Automatic health management** - Crashing applications will restart automatically.
- **Automatic placement** - Applications are automatically placed across multiple data centers.
- **Automatic routing** - Internet-reachable routes are automatically created for your applications.

Use Cases

- **Deploy a secure web application across multiple regions**

  Apps or parts of your apps will have outages - it is a fact. It can be a problem in your code, a planned maintenance impacting the resources used by your app, a hardware failure bringing down a zone, a location, a data center where your app is hosted. Any of these will happen and you have to be prepared. With IBM Cloud, you can deploy your application to multiple locations to increase your application resilience. And with your application now running in multiple locations, you can also redirect user traffic to the nearest location to reduce latency.

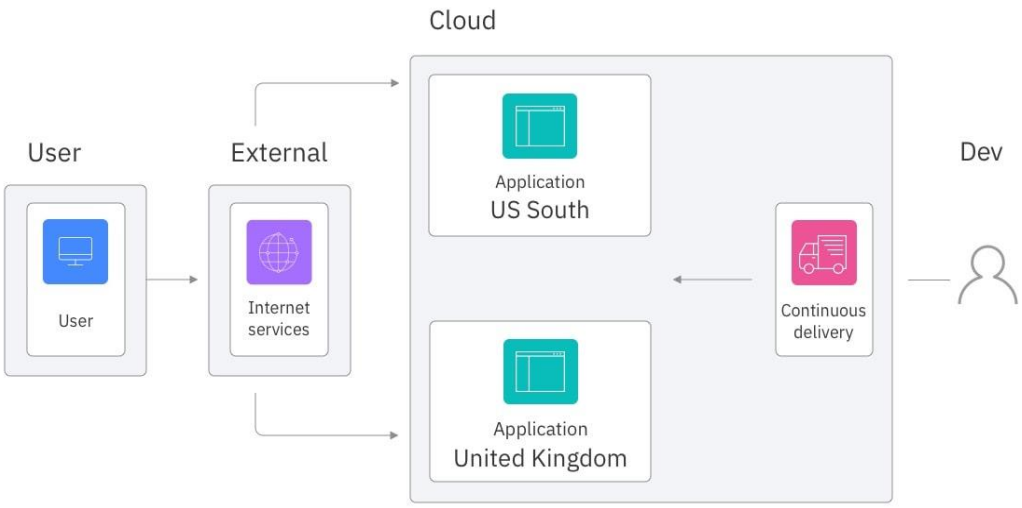


Image 1- Deploy across regions

Reference: https://1.cms.s81c.com/sites/default/files/2021-12-23/cloud-foundry-use-cases-deploy-across-regions.jpg

- **Modern web application using MEAN stack**

  It is composed of a **M**ongoDB, **E**xpress web framework, **A**ngular front end framework and a **N**ode.js runtime. The user accesses the application using a web browser. The Node.js app running in Code Engine accesses the Databases for MongoDB database to fetch data.

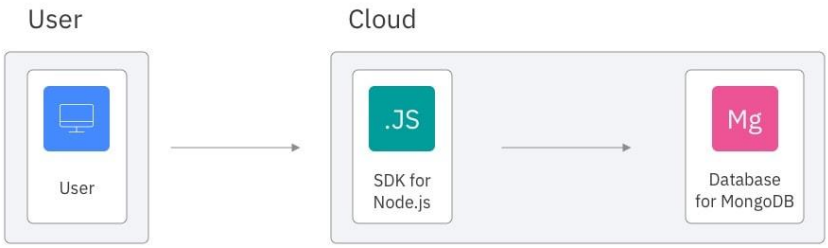


Image 2- Web application using MEAN stack

Reference: https://1.cms.s81c.com/sites/default/files/2021-12-22/cloud-foundry-use-cases-using-mean-stack.jpg

- **Analyze logs and monitor health of a Cloud Foundry application**

  IBM Log Analysis service can be used to configure and access logs of a Kubernetes application that is deployed on IBM Cloud. User connects to the application and generates log entries. The application runs in a Kubernetes cluster from an image stored in the Container Registry. The user will configure IBM Log Analysis service agent to access application and cluster-level logs. The user will configure IBM Cloud Monitoring service agent to monitor the health and performance of the IBM Cloud Kubernetes Service cluster and also the app deployed to the cluster.

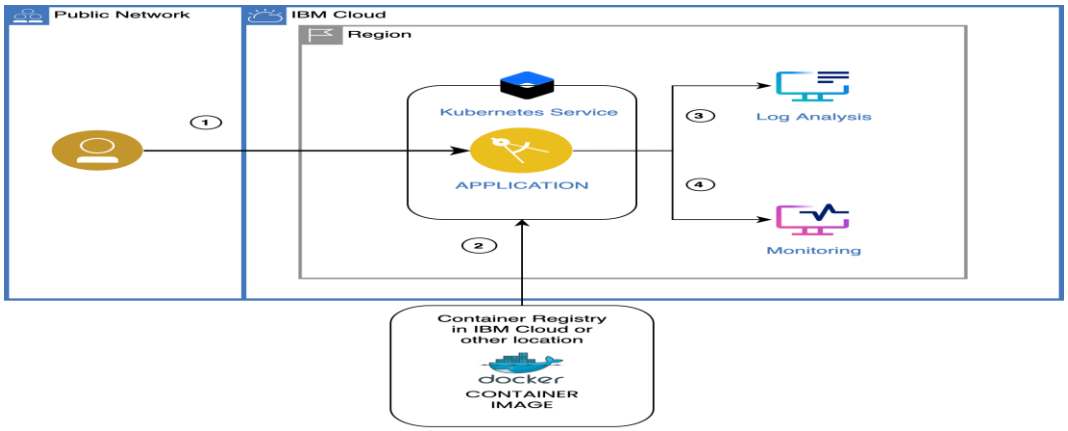


Image 2- IBM Log Analysis and Monitoring

Reference: https://cloud.ibm.com/docs-content/v1/content/c2630dd3261834db4c7cb144522ae09438afcf0e/solution-tutorials/images/solution12/Architecture.png

Get introduced to Storage

What is cloud storage?

Storage growth continues at a significant rate, driven by new workloads like analytics, video and mobile applications. While storage demand is increasing, most IT organizations are under continued pressure to lower the cost of their IT infrastructure through the use of shared cloud computing resources. It's vital for software designers and solution architects to match the specific requirements of their workloads to the appropriate storage solution or, in many enterprise cases, a mix.

Cloud storage allows you to save data and files in an off-site location that you access either through the public internet or a dedicated private network connection. Data that you transfer off-site for storage becomes the responsibility of a third-party cloud provider. The provider hosts, secures, manages, and maintains the servers and associated infrastructure and ensures you have access to the data whenever you need it.

Cloud storage delivers a cost-effective, scalable alternative to storing files on on-premise hard drives or storage networks. Computer hard drives can only store a finite amount of data. When users run out of storage, they need to transfer files to an external storage device. Traditionally, organizations built and maintained storage area networks (SANs) to archive data and files. SANs are expensive to maintain, however, because as stored data grows, companies have to invest in adding servers and infrastructure to accommodate increased demand.

One of the biggest advantages of cloud storage is flexibility. A company that has your data or data you want will be able to manage, analyze, add to and transfer it all from a single dashboard something impossible to do today on storage hardware that sits alone in a data center.

The other major benefit of storage software is that it can access and analyze any kind of data wherever it lives, no matter the hardware, platform or format. So, from mobile devices linked to your bank to servers full of unstructured social media information, data can be understood via the cloud.

How does it work?

Like on-premise storage networks, cloud storage uses servers to save data; however, the data is sent to servers at an off-site location. Most of the servers you use are virtual machines hosted on a physical server. As your storage needs increase, the provider creates new virtual servers to meet demand.

Typically, you connect to the storage cloud either through the internet or a dedicated private connection, using a web portal, website, or a mobile app. The server with which you connect forwards your data to a pool of servers located in one or more data centers, depending on the size of the cloud provider's operation. As part of the service, providers typically store the same data on multiple machines for redundancy. This way, if a server is taken down for maintenance or suffers an outage, you can still access your data.

Cloud storage is available in private, public and hybrid clouds.

Public storage clouds: In this model, you connect over the internet to a storage cloud that's maintained by a cloud provider and used by other companies. Providers typically make services accessible from just about any device, including smart phones and desktops and let you scale up and down as needed.

Private cloud storage: Private cloud storage setups typically replicate the cloud model, but they reside within your network, leveraging a physical server to create instances of virtual servers to increase capacity. You can choose to take full control of an on-premise private cloud or engage a cloud storage provider to build a dedicated private cloud that you can access with a private connection. Organizations that might prefer private cloud storage include banks or retail companies due to the private nature of the data they process and store.

Hybrid cloud storage: This model combines elements of private and public clouds, giving organizations a choice of which data to store in which cloud. For instance, highly regulated data subject to strict archiving and replication requirements is usually more suited to a private cloud environment, whereas less sensitive data (such as email that doesn't contain business secrets) can be stored in the public cloud. Some organizations use hybrid clouds to supplement their internal storage networks with public cloud storage.

Types of cloud storage

There are three main types of cloud storage: block, file, and object. Each comes with its set of advantages:

Block Storage: Traditionally employed in SANs, block storage is also common in cloud storage environments. In this storage model, data is organized into large volumes called “blocks." Each block represents a separate hard drive. Cloud storage providers use blocks to split large amounts of data among multiple storage nodes. Block storage resources provide better performance over a network thanks to low IO latency (the time it takes to complete a connection between the system and client) and are especially suited to large databases and applications. Used in the cloud, block storage scales easily to support the growth of your organization’s databases and applications. Block storage would be useful if your website captures large amounts of visitor data that needs to be stored.

File Storage: The file storage method saves data in the hierarchical file and folder structure with which most of us are familiar. The data retains its format, whether residing in the storage system or in the client where it originates, and the hierarchy makes it easier and more intuitive to find and retrieve files when needed. File storage is commonly used for development platforms, home directories, and repositories for video, audio, and other files.

Object storage: Object storage differs from file and block storage in that it manages data as objects. Each object includes the data in a file, its associated metadata, and an identifier. Objects store data in the format it arrives in and makes it possible to customize metadata in ways that make the data easier to access and analyze. Instead of being organized in files or folder hierarchies, objects are kept in repositories that deliver virtually unlimited scalability. Since there is no filing hierarchy and the metadata is customizable, object storage allows you to optimize storage resources in a cost-effective way.

Pros and cons

The pros of cloud storage include the following:

Off-site management: Your cloud provider assumes responsibility for maintaining and protecting the stored data. This frees your staff from tasks associated with storage, such as procurement, installation, administration, and maintenance. As such, your staff can focus on other priorities.

Quick implementation: Using a cloud service accelerates the process of setting up and adding to your storage capabilities. With cloud storage, you can provision the service and start using it within hours or days, depending on how much capacity is involved.

Cost-effective: As mentioned, you pay for the capacity you use. This allows your organization to treat cloud storage costs as an ongoing operating expense instead of a capital expense with the associated upfront investments and tax implications.

Scalability: Growth constraints are one of the most severe limitations of on-premise storage. With cloud storage, you can scale up as much as you need. Capacity is virtually unlimited.

Business continuity: Storing data offsite supports business continuity in the event that a natural disaster or terrorist attack cuts access to your premises.

Cloud storage cons include the following:

Security: Security concerns are common with cloud-based services. Cloud storage providers try to secure their infrastructure with up-to-date technologies and practices, but occasional breaches have occurred, creating discomfort with users.

Administrative control: Being able to view your data, access it, and move it at will is another common concern with cloud resources. Offloading maintenance and management to a third party offers advantages but also can limit your control over your data.

Latency: Delays in data transmission to and from the cloud can occur as a result of traffic congestion, especially when you use shared public internet connections. However, companies can minimize latency by increasing connection bandwidth.

Regulatory compliance: Certain industries, such as healthcare and finance, have to comply with strict data privacy and archival regulations, which may prevent companies from using cloud storage for certain types of files, such as medical and investment records. If you can, choose a cloud storage provider that supports compliance with any industry regulations impacting your business.

IBM Cloud Storage

IBM Cloud Storage offers a comprehensive suite of cloud storage services, including out-of-the-box solutions, components to create your own storage solution, and standalone and secondary storage. We can take the advantage of IBM's automated data backup and recovery system, which is managed through the IBM Cloud Backup WebCC browser utility. The system allows you to securely back up data in one or more IBM cloud data centers around the world.

Benefits of IBM Cloud solutions include:

- Global reach
- Scalability
- Flexibility
- Simplicity

PaaS Model services in IBM Cloud

PaaS, or Platform-as-a-Service, provides a complete, flexible and cost-effective cloud platform for developing, running and managing applications. PaaS, or Platform-as-a-Service, is a cloud computing model that provides customers a complete cloud platform—hardware, software, and infrastructure—for developing, running, and managing applications without the cost, complexity, and inflexibility that often comes with building and maintaining that platform on-premises. The PaaS provider hosts everything—servers, networks, storage, operating system software, databases, development tools—at their data center. Typically customers can pay a fixed fee to provide a specified amount of resources for a specified number of users, or they can choose 'pay-as-you-go' pricing to pay only for the resources they use. Either option enables PaaS customers to build, test, deploy run, update and scale applications more quickly and inexpensively they could if they had to build out and manage their own on-premises platform.

Every leading cloud service provider—including Amazon Web Services (AWS), Google Cloud, IBM Cloud and Microsoft Azure—has its own PaaS offering. Popular PaaS solutions are also available as open source projects (e.g. Apache Stratos, Cloud Foundry) or from software ventors (e.g. Red Hat OpenShift and Salesforce Heroku). IBM provides rich and scalable PaaS solutions for developing cloud native applications from scratch, or modernizing existing applications to benefit from the flexibility and scalability of the cloud.

IBM Red Hat OpenShift on IBM Cloud is a fully managed OpenShift service that uses the enterprise scale and security of IBM Cloud to automate updates, scaling and provisioning, and to handle unexpected surges in traffic. Your teams can jump-start development and app modernization with a range of tools and features, and deploy highly available fully-managed clusters with a single click.

IBM Cloud Pak for Applications helps you modernize existing applications, embed additional security, and develop new apps that unleash digital initiatives. It offers cloud-native development solutions that can quickly deliver value, along with flexible licensing that can be tailored to your specific needs.

Hosting web application on PaaS

Creating web app with the IBM Cloud console

After signing up, start to build your first app by using the IBM Cloud catalog and console. In IBM Cloud apps are associated with IBM Cloud organizations and spaces. An organization is owned and used by multiple collaborators. Initially, you get a default organization that is named after your user name and you are the only collaborator. You also get a space within this organization. The space is an environment to run your apps; for example, you can have a dev space as a development environment, a test space as a test environment, and a production space as a production environment. Each of the environments belongs to a region. With IBM Cloud, you can deploy your apps to a specific geographical region for lower network latency, data privacy, and better availability.

Assume that you are in the US and most of your app users are also in the US. For lower network latency you decide to build and run your app close to your user base. After logging in to IBM Cloud, click the user account preferences link and select the US South region. Then, you can take the following steps to create an app:

- Click Catalog in the IBM Cloud toolbar.
- Click Cloud Foundry Apps and choose the Cloud Foundry tile.
- Click Public Applications to select Region and Runtime.
    - Type a unique name for your app and click Create. The app name must be unique in the whole IBM Cloud environment.

Once created, a Getting started page is added to the left navigation pane. Follow the instructions in that page to download the starter code of your app, modify, and deploy it. The app is created with one instance and 512 MB memory quota by default. Click Overview to specify your app instances and memory quota.

Building app by using the Cloud Foundry CLI

First, download and develop your app code.

- Click Getting started in your app dashboard.
- Click the Download the sample code link to download your app code.
- Extract the downloaded file to a directory.
- Develop the code with your locally integrated development environment.

Install the ibmcloud cf command line interface (CLI).

- Download the ibmcloud cf command line tool installation program for your operating system.
- Follow the tool wizard to complete the installation.
- Use the ibmcloud cf -v command to verify the version of the ibmcloud cf command line interface.

After you install the ibmcloud cf command line interface, you must specify which IBM Cloud region you want to work with by using the ibmcloud cf api command. For example, the API endpoint for the US South region is api.us-south.cf.cloud.ibm.com.

ibmcloud cf api api.us-south.cf.cloud.ibm.com

Additional API endpoints for other regions can be found at https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-endpoints. Next, log in to IBM Cloud by using the ibmcloud cf login command. After you are logged in to IBM Cloud, you are ready to deploy your app back to IBM Cloud. From your app directory, enter the following command:

ibmcloud cf push <your_appname>

Understanding web application deployment strategies and planning

Developing an application deployment strategy

An application deployment strategy is a way to change or update an application in a production environment in a controlled manner. This can be important for many reasons, including the following:

Avoiding application downtime

Enabling new functionality to be tested in production without impacting customers

Limiting the impact of potential issues in production to a subset of users

Enabling rapid rollback to the previous version, if issues are found

There are many possible deployment strategies. In general, they depend on running multiple instances of the application and managing how the various instances are updated. These are three recommended strategies:

Rolling: Deploy a new release with no downtime by incrementally updating all the instances in a sequential manner.

Blue-Green: Deploy a new release to an environment identical to — and isolated from — current production for quicker rollbacks and ease of testing.

Canary: Deploy a new release incrementally to the production environment, gradually replacing the older version of the application with the newer one, gating increments with a test to reducing risks.

App deployment and IBM Cloud Continuous Delivery

IBM Cloud Continuous Delivery supports the rolling, blue-green and canary deployment strategies. You can deploy containerized or non-containerized workloads using these strategies to a Kubernetes cluster or virtual machines (VM), respectively.

The deployment strategies can be configured in a few clicks by using a guided setup to create a toolchain based on a template included with the Continuous Delivery service, which also provides Tekton Pipelines, Git Repos and Issue Tracking, Code Risk Analyzer, DevOps Insights and the Eclipse Orion Web IDE in the Cloud.

The toolchain templates also take advantage of other IBM Cloud services, such as IBM Cloud Secrets Manager, IBM Key Protect for IBM Cloud, IBM Cloud Object Storage and IBM Cloud Container Registry. Users can customize the toolchain to use external tools that enterprises have standardized upon, such as Git providers and artifact stores. Users can also add triggers, policies under pipelines and configuring quality metrics as per your requirements.

IBM Cloud Continuous Delivery

Secure web application across multiple regions

Objectives

- Deploy a Cloud Foundry application to multiple locations with Continuous Delivery.
- Map a custom domain to the application.
- Configure global load balancing to your multi-location application.
- Bind an SSL certificate to your application.
- Monitor application performance.

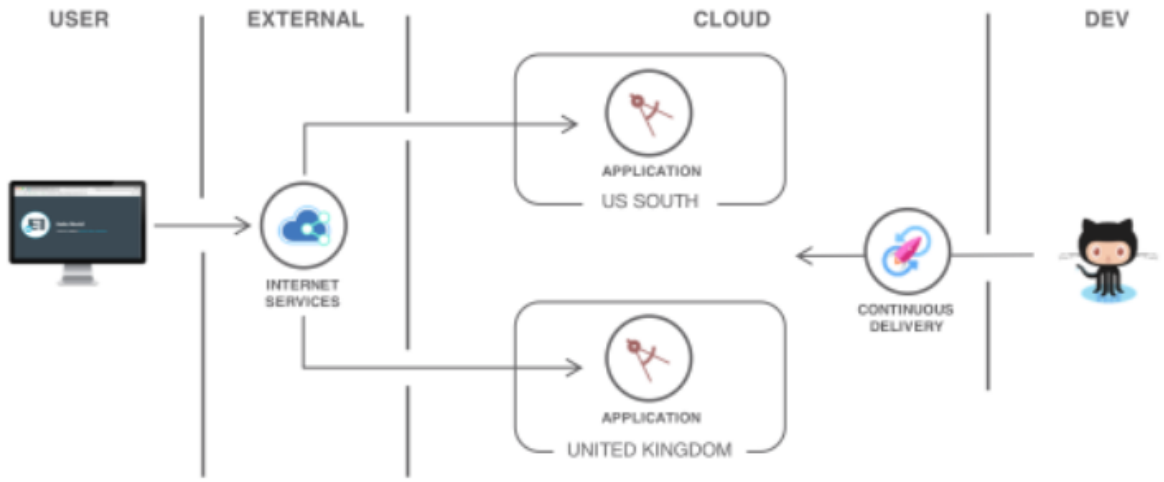


Image : Secure web application across multiple regions
Reference: https://cloud.ibm.com/docs/solution-tutorials?topic=solution-tutorials-multi-region-webapp

Before you begin

Requirements:

- A custom domain and IBM Cloud Internet Services for load balancing the application across multiple regions. The custom domain is needed so you can configure the DNS for this domain to point to IBM Cloud Internet Services name servers. If you do not own a domain, you can buy one from a registrar.
- git to clone the source code repository,
- Node.js and related tools to run code locally.

Step 1: Create a Node.js application

Start by creating a SDK for Node.js starter application that runs in a Cloud Foundry environment.

1. In the Catalog, under the Compute category, select Cloud Foundry, Select Public Applications and choose the SDK for Node.js runtime.
2. Verify SDK for Node.js is checked.
3. Select a region. Dallas, us-south, is expected for the text below, but any region will work if you make the appropriate substitutions.
4. Enter a unique name for your application, for example: myusername-nodeapp. Notice how the host name is filled identically.
5. Take note of the selected IBM Domain. The default for the Dallas region is us-south.cf.appdomain.cloud.
6. Select an Organization and Space. The space will be named after the region: myaccount-org (Organization) / us-south (Space). A second space will be needed in a future step.
7. Click Create.
8. After the application starts, click the Visit URL link on the Overview page to see your application in a new tab.

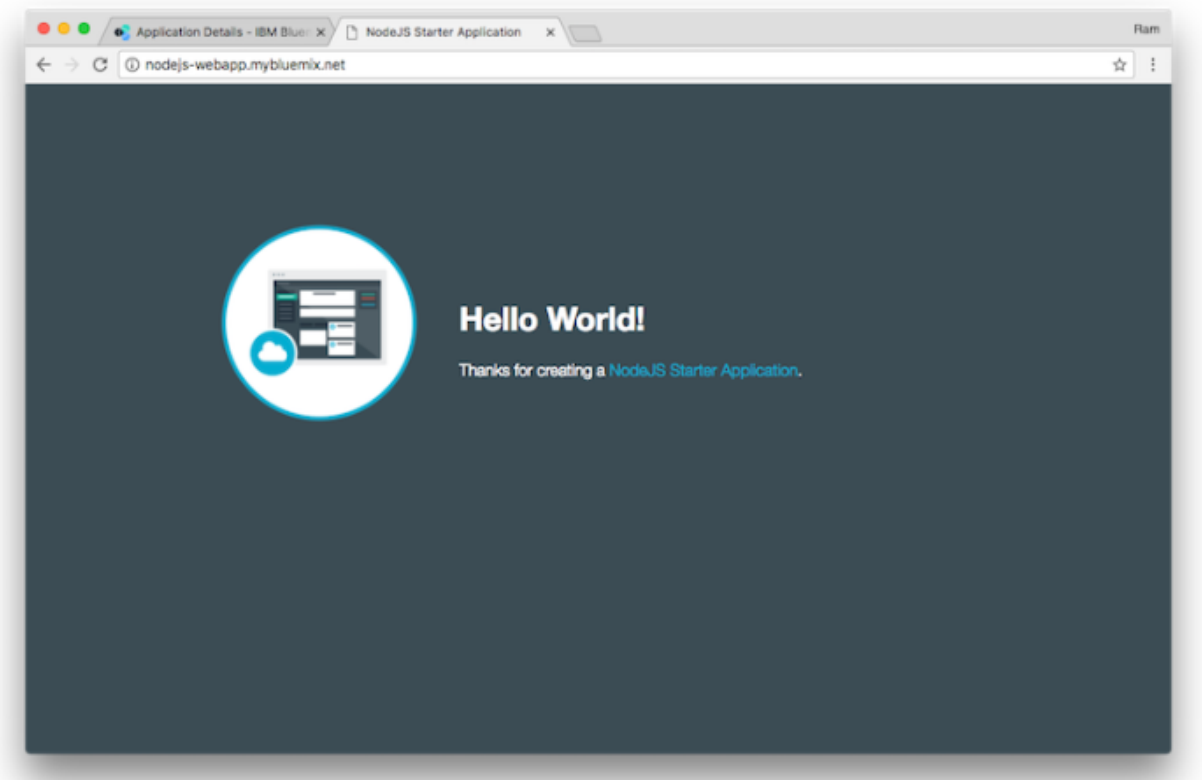


Image : Create a basic nodejs application
Reference: https://cloud.ibm.com/docs/solution-tutorials?topic=solution-tutorials-multi-region-webapp

Step 2: Set up source control and Continuous Delivery

In this step, you set up a git source control repository to store your code and then create a pipeline, which deploys any code changes automatically.

1. On the left pane of your application you just created, select Overview and scroll to find Continuous Delivery. Click Enable Continuous Delivery.
2. Keep the default options in the Git Repos and Issue Tracking panel.
3. Click the Delivery Pipeline panel and choose New to create a API key with full access and click OK or follow the instructions to create or use a specific API key.
4. Click Create. You should now have a default toolchain created.
5. Select the Git tile under Code. You're then directed to your git repository page.

6. If you haven't set up SSH keys yet, you should see a notification bar at the top with instructions. Follow the steps by opening the add an SSH key link in a new tab or if you want to use HTTPS instead of SSH, follow the steps by clicking create a personal access token. Remember to save the key or token for future reference.
7. Select SSH or HTTPS and copy the git URL. Clone the source to your local machine.

```
$ git clone <your_repo_url>
cd <name_of_your_app>
```

Note: If you're prompted for a user name, provide your git user name. For the password, use an existing SSH key or personal access token or the one you created in the previous step.

8. Open the cloned repository in an IDE of your choice and navigate to public/index.html. Now, let's update the code. Try changing "Hello World" to something else.
9. Run the application locally by running the commands npm install and npm start, then visit localhost:<port_number> in your browser. <port_number> is displayed on the console. You can click on the link.
10. Push the change to your repository with three simple steps: Add, commit, and push.

```
$ git add public/index.html
git commit -m "my first changes"
git push origin master
```

11. Go to the toolchain you created earlier and click the Delivery Pipeline tile.
12. Confirm that you see the BUILD and DEPLOY stages.

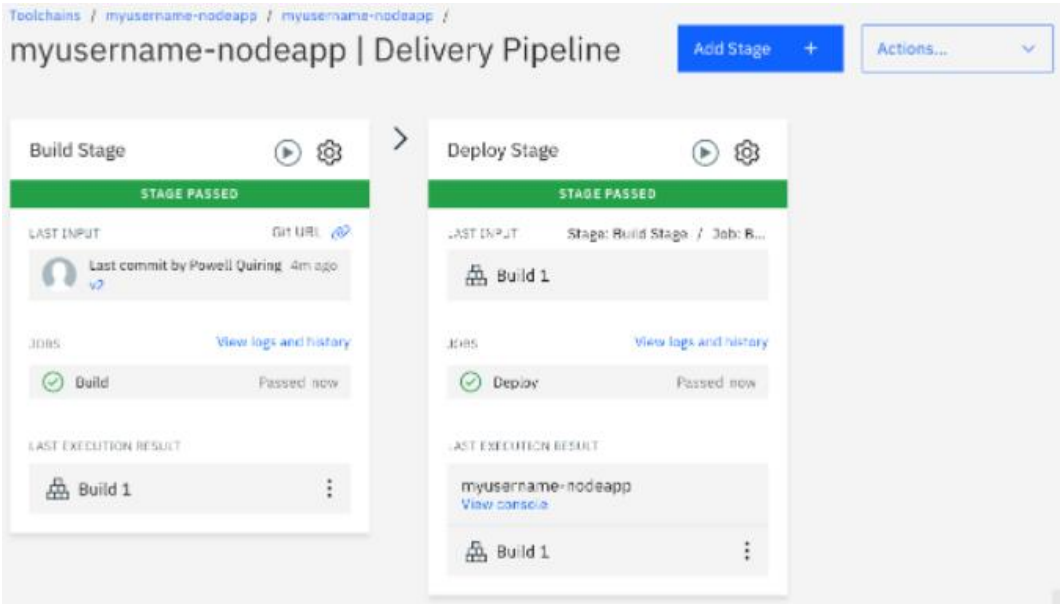


Image : Application Delivery Pipeline
Reference: https://cloud.ibm.com/docs/solution-tutorials?topic=solution-tutorials-multi-region-webapp

13. Wait for the DEPLOY stage to complete.
14. Click the View console to open the application under Last execution result to visit the application. In the Cloud Foundry application click on Visit app URL to see the app and changes.

Continue making further changes to your application and periodically commit your changes to your git repository. If you don't see your application updating, check the logs of the DEPLOY and BUILD stages of your pipeline.

Preparing application for cloud deployment over PaaS model instance in IBM Cloud Foundry

What is Cloud Foundry?

Cloud Foundry is the premier industry standard Platform-as-a-Service (PaaS) that ensures the fastest, easiest, and most reliable deployment of cloud-native apps. Cloud Foundry ensures that the build and deploy aspects of coding remain carefully coordinated with any attached services; resulting in quick, consistent, and reliable iterating of apps.

Benefits of Cloud Foundry

- Choose your own language - IBM Cloud® Foundry includes runtimes for Java™, Node.js, PHP, Python, Ruby, Swift, and Go. Cloud Foundry community build packs are also available. Combined with DevOps services, the app runtimes enable a delivery pipeline that automates much of the iterative development process.
- Fault tolerant - Runtimes facilitate developing apps as stateless processes that quickly: start and stop, replicate if an instance fails, and duplicate if sustained or increased performance requires.
- Extend apps with services - Runtimes link IBM Cloud services to apps as endpoints, giving any instance of an app embedded knowledge of how to manage relevant calls and data. In fact, runtimes manage all linked resources this way: SDKs, APIs (whether made available as cloud services or exposed from within a traditional enterprise as custom services), and also apps themselves when used as resources by other apps.
- Access control - Fine grain assignment/dispensing of compute capacity to development teams.
- Automatic placement - Apps are automatically placed across multiple data-center PODs for maximum reliability.
- Automatic health management - Crashing apps restart automatically.
- Automatic routing - Internet reachable routes are automatically created for your apps.
- High availability - Supports full high availability for high app availability.
- Automatic deployment scaling - The Auto-Scaling for IBM Cloud service automatically increases or decreases the compute capacity of your app, to rapidly adjust to dynamic loading needs.

Getting started with Cloud Foundry Public

Developing your app

There are three ways to develop your app:

1. With the Continuous Delivery service
2. In the IBM Cloud® console
3. At the Cloud Foundry command line

Developing and deploying your apps by using toolchains and the Continuous Delivery service

Add a toolchain that includes the IBM Cloud® Continuous Delivery service to your app. Then, use the toolchain to develop and deploy your app.

Creating your web app with the IBM Cloud console

After signing up, start to build your first app by using the IBM Cloud catalog and console.

In IBM Cloud apps are associated with IBM Cloud organizations and spaces. An organization is owned and used by multiple collaborators. Initially, you get a default organization that is named after your user name and you are the only collaborator. You also get a space within this organization. The space is an environment to run your apps; for example, you can have a dev space as a development environment, a test space as a test environment, and a production space as a production environment. Each of the environments belong to a region. With IBM Cloud, you can deploy your apps to a specific geographical region for lower network latency, data privacy, and better availability.

For this scenario, you want to develop a web app by using Node.js. Assume that you are in the US and most of your app users are also in the US. For lower network latency you decide to build and run your app close to your user base. After logging in to IBM Cloud, click the user account preferences link and select the US South region. Then, you can take the following steps to create an app:

1. Click Catalog in the IBM Cloud toolbar.
2. Click Cloud Foundry Apps and choose the Cloud Foundry tile.
3. Click Public Applications to select Region and Runtime.
4. Type a unique name for your app and click Create. The app name must be unique in the whole IBM Cloud environment.

Once created, a Getting started page is added to the left navigation pane. Follow the instructions in that page to download the starter code of your app, modify, and deploy it.

The app is created with one instance and 512 MB memory quota by default. You can increase the memory, or add more instances to get high availability of your app, for example, three instances with 1 GB memory per instance. Click Overview to specify your app instances and memory quota. For example, type 3 for instances and 1 GB for memory quota, and click Save. You can also see the files, logs, and environment variables to troubleshoot your problems.

Binding a service by using IBM Cloud console

After creating your app, connect to a database with your app. You can store and retrieve the app data by using a database query language. In this scenario, you decide to use the IBM Cloudant service that is provided by IBM Cloud.

To use IBM Cloud with your Cloud Foundry app, create a service instance and bind your app to the service instance:

1. In the IBM Cloud catalog, select the IBM Cloudant service. Add a unique name for your IBM Cloudant service and click Create. In the Cloudant Manage panel, launch the service by clicking Launch.
2. Click Connections. Then, click Create connection.
3. Click Connect next to your app.
4. The Restage App window is displayed. Click Restage.

Now your app is bound to the IBM Cloudant service. The <VCAP_SERVICES> environment variable contains the data that is required for the app to communicate with the service instance. Because IBM Cloud hosts several apps on the same virtual machine, multiple apps cannot use the same HTTP port number to receive incoming requests. To avoid conflicts, each app is given a unique port number. This port number is available in the <VCAP_APP_PORT> variable.

Follow these steps to see the list of <VCAP_SERVICES> values associated with your app in the console.

1. Click the menu in the IBM Cloud toolbar.
2. Click Dashboard.
3. Click your app.
4. Click Runtimes and select the Environment Variables tab.

Note: This environment variable is the serialization of a JSON object with one entry for each service instance that the app is bound to. The amount and type of data that each service instance provides are service-specific. When your app does not use a service, <VCAP_SERVICES> is an empty JSON object. This environment variable is used only when you add a service to your app.

Building your app by using the Cloud Foundry CLI

IBM Cloud provides several tools for you to start coding your app, for example, the ibmcloud cf command line interface and Eclipse tools. In this example you will use the ibmcloud cf command line interface to start coding your app.

1. First, download and develop your app code.
    a. Click Getting started in your app dashboard.
    b. Click the Download the sample code link to download your app code.
    c. Extract the downloaded file to a directory.
    d. Develop the code with your locally integrated development environment.

2. Install the ibmcloud cf command line interface (CLI).
    a. Download the ibmcloud cf command line tool installation program for your operating system.
    b. Follow the tool wizard to complete the installation.
    c. Use the ibmcloud cf -v command to verify the version of the ibmcloud cf command line interface.

3. After you install the ibmcloud cf command line interface, you must specify which IBM Cloud region you want to work with by using the ibmcloud cf api command. The API endpoint for the US South region is api.us-south.cf.cloud.ibm.com. Additional API endpoints for other regions can be found here. Enter the following command to connect to IBM Cloud:

```
$ ibmcloud cf api api.us-south.cf.cloud.ibm.com
```

To find other API endpoints, see Regions and Endpoints. After you specify the IBM Cloud region, the location information that you specified is saved.

4. Next, log in to IBM Cloud by using the ibmcloud cf login command.

```
$ ibmcloud cf login -u <your_user_ID> -p <password> -o
<your_org_name> -s <your_space_name>
```

If your organization uses single sign on, use ibmcloud cf login -sso.

5. After you are logged in to IBM Cloud, you are ready to deploy your app back to IBM Cloud. From your app directory, enter the following command:

```
$ ibmcloud cf push <your_appname>
```

6. Now, you can access the app by entering the following app URL in a browser:

```
http://<your_app>.us-south.cf.appdomain.cloud
```

You can also choose other tools to build your app, such as Eclipse tools. For more information, see the Getting started page of your app in the IBM Cloud console.

Binding a service by using the ibmcloud cf cli

You can also bind a service by using the ibmcloud cf command line interface. This example assumes that you want to add the IBM Cloudant service to your app with the ibmcloud cf command line interface.

To use the IBM Cloudant service within your app, create an IBM Cloudant service instance, bind your app to the service instance, and then use the service instance. The same procedure applies to all the other services.

1. **Create an IBM Cloudant NoSQL DB service instance.**

Use the ibmcloud cf create-service command to create a new instance of a service. In this example, <Lite> is the name of the plan. For example:

```
$ ibmcloud cf create-service cloudantNoSQLDB <Lite>
<your_name_for_your_cloudant_service>
```

You can also use the ibmcloud cf services command to see the list of service instances that you created.

```
$ ibmcloud cf services
```

After a service instance is created, it is available for any of your apps to bind and use.

2. **Bind the service instance to your app.**

To use a service instance, you must bind it to your app. Use the ibmcloud cf bind-service command to bind a service instance to an app by specifying the app name and the service instance that you created.

```
$ ibmcloud cf bind-service <your_app_name>
<your_name_for_your_cloudant_service>
```

Binding a service instance to an app enables IBM Cloud to communicate to the service, and to specify that a new app will communicate with that service instance. For different services, IBM Cloud might process the app and the service instance differently during the binding. For example, some services might create a new tenant for each app that communicates to the service instance. The service responds back to IBM Cloud with information, such as credentials, that must be passed to the app allowing communication between the app and the service.

Note: If the app is running when it is bound to a service instance, the <VCAP_SERVICES> environment variable is not updated until the app is restarted. To restart your app, use the ibmcloud cf restart command.

3. **Use the service instance.**

In this scenario, the <VCAP_SERVICES> environment variable includes information, such as the following items, that an app can use to connect to this instance of IBM Cloudant:

username: d72837bb-b341-4038-9c8e-7f7232916197-bluemix

password: secret

url: https://d72837bb-b341-4038-9c8e-7f7232916197-bluemix:b6fc4708942b70a88853177ee52a528d07a43fa8575a69abeb8e044a7b0a7424@d72837bb-b341-4038-9c8e-7f7232916197-bluemix.cloudant.com

For example, your Node.js app might access this information as follows:

```
if (process.env.VCAP_SERVICES) {
      var env = JSON.parse(process.env.VCAP_SERVICES);
      var cloudant = env['cloudantNoSQLDB'][0].credentials;
} else {
      var cloudant = {
              "username" : "user1",
              "password" : "secret",
              "url" : "https://user1:secret@localhost:25002"
              }
      };
```

Note: As the sample code shows, to connect to an IBM Cloudant service instance, you can check whether the <VCAP_SERVICES> environment variable exists first. If it exists, the app can use the IBM Cloudant variable's properties to access the database. However, if the <VCAP_SERVICES> environment variable is not present, the local IBM Cloudant service instance is used with the provided default values.

4. **Interact with the service instance.**

You can interact with the service instance by using the credential information. The actions that you can take include read, write, and update. The following example demonstrates how to insert a JSON object into the IBM Cloudant service instance:

```
// create a new message
var create_message = function(req, res) {
  require('cloudantdb').connect(cloudant.url, function(err, conn) {
    var collection = conn.collection('messages');

// create message record
 var parsedUrl = require('url').parse(req.url, true);
 var queryObject = parsedUrl.query;
 var name = (queryObject["name"] || 'Bluemix');
 var message = { 'message': 'Hello, ' + name, 'ts': new Date()
 };
 collection.insert(message, {safe:true}, function(err){
   if (err) { console.log(err.stack); }
   res.writeHead(200, {'Content-Type': 'text/plain'});
   res.write(JSON.stringify(message));
   res.end('\n');
     });
   });
 }
```

5. **Optional: Unbind or delete a service instance.**

You might want to unbind or delete a service instance when it is no longer used or when you want to free up some spaces. To unbind a service instance from your app, use the ibmcloud cf unbind-service command. To delete a service instance, use the ibmcloud cf delete-service command.

For more information about services, see Services. For more information about the ibmcloud cf options that you can use to manage your apps in the IBM Cloud environment, run ibmcloud cf --help in the ibmcloud cf command line interface.

Removing apps

As you build more apps, the quota might approach your limits. However, some apps that you might no longer use still occupy the quota. It’s easy to delete apps to free up some spaces in IBM Cloud at any time.

In the IBM Cloud console, go to the app Overview page, click the menu icon, and delete the app that you no longer use. You can also use the ibmcloud cf delete command to delete apps.

Cloud Best Practices

IBM Cloud CLI

What is IBM Cloud CLI?

The IBM Cloud® Command Line Interface provides commands for managing resources in IBM Cloud. When you install the standalone IBM Cloud CLI, you get only the CLI itself without any recommended plug-ins or tools.

Getting started with the IBM Cloud CLI

In this, you install the IBM Cloud® Command Line Interface, along with the option to install popular plug-ins and tools so that you can work with apps, toolchains, pipelines, Kubernetes clusters, and more in IBM Cloud.

The installation command in this tutorial installs the latest stand-alone IBM Cloud CLI version available, with the option of installing the following tools manually:

- Homebrew (Mac only)
- Git
- Docker
- Helm
- Kubectl
- curl (Linux™ only)
- IBM Cloud Functions plug-in
- IBM Cloud Object Storage plug-in
- IBM Cloud Container Registry plug-in
- IBM Cloud Kubernetes Service plug-in

Before you begin

- Depending on your IBM Cloud® account type, access to certain resources might be limited or constrained. Depending on your plan limits, certain capabilities that are required by some toolchains might not be available. For more information, see Setting up your IBM Cloud account.
- You must use the stable channel for Docker with a minimum version of 1.13.1.
- For Linux™, install the curl command for downloading packages through the command line. If curl is already installed, the installer updates it to the latest version.
- For Windows™, some functions are not supported unless you are running Windows™ 10 Pro.

Step 1: Run the installation command

The latest version of the IBM Cloud CLI is installed when you run the command. As the CLI installs, keep an eye on the command line to authenticate as needed.

- For MacOS, run the following command:

```
curl -fsSL https://clis.cloud.ibm.com/install/osx | sh
```

- For Linux™, run the following command:

```
curl -fsSL https://clis.cloud.ibm.com/install/linux | sh
```

- For Windows™ 10 Pro, run the following command in PowerShell as an administrator:

```
iex(New-Object
Net.WebClient).DownloadString('https://clis.cloud.ibm.com/install/powershell')
```

- For WSL2 on Windows™, run the following command:

```
curl -fsSL https://clis.cloud.ibm.com/install/linux | sh
```

Step 2: Verify the installation

To verify that the CLI was installed successfully, run the help command:

```
ibmcloud help
```

The output lists the usage instructions, the current version, and the supported commands.

Step 3: Install CLI plug-ins and tools

To manually install the CLI plug-ins and tools, see installing the tools and plug-ins manually.

Step 4: Configure your environment

1. Log in to IBM Cloud with your IBMid. If you have multiple accounts, you are prompted to select which account to use. If you do not specify a region with the -r flag, you must also select a region.

```
ibmcloud login
```

2. If you plan to access Cloud Foundry services, specify a Cloud Foundry org and space. You can run the following command to interactively identify the org and space:

```
ibmcloud target --cf
```

Or, if you know which org and space that the service belongs to, you can use the following command:

```
ibmcloud target -o <value> -s <value>
```

Next steps

- If you installed the CLI plug-ins and tools, you're now ready to develop and deploy your first app. For more information, see Creating and deploying apps by using the CLI.
- Stay up to date with the IBM Cloud CLI by subscribing to the IBM Cloud CLI releases repository. You'll receive notifications about new IBM Cloud CLI releases.
- IBM Cloud CLI supports a plug-in framework to extend its capability. Discover and install new CLI plugins!

Installing the stand-alone IBM Cloud CLI

The IBM Cloud® Command Line Interface provides commands for managing resources in IBM Cloud. When you install the standalone IBM Cloud CLI, you get only the CLI itself without any recommended plug-ins or tools.

Before you begin

If you need to use a 32-bit version, or a previous version other than the latest for IBM Cloud Dedicated environments, see IBM Cloud CLI releases.

Installing with an installer

Use the following steps to install the latest stand-alone IBM Cloud CLI:

1. Use a browser to access the official ibm-cloud-cli-releases GitHub repository, and select the installer of your OS to begin the download. The following operating systems are supported: macOS X 64-bit, Windows™ 64-bit, Linux™ x86 64-bit, and Linux™ LE 64-bit (ppc64le).
2. Run the installer:
   - For Mac and Windows™, run the installer.
   - For Linux™, extract the package and run the install script.
3. Log in to IBM Cloud:

   ```
   ibmcloud login
   ```

   Now, you're ready to manage IBM Cloud resources. Enter ibmcloud help to view the command descriptions.

Installing from the shell

To install the latest CLI for your OS from the shell manually, use the following command for your OS:

- For Mac, copy and paste the following command to a terminal and run it:

```
curl -fsSL https://clis.cloud.ibm.com/install/osx | sh
```

- For Linux™, copy and paste the following command to a terminal and run it:

```
curl -fsSL https://clis.cloud.ibm.com/install/linux | sh
```

- For Windows™, copy and paste the following command to a Windows™ PowerShell terminal console and run it:

```
iex(New-Object
Net.WebClient).DownloadString('https://clis.cloud.ibm.com/install/powershell')
```

If you encounter errors like The underlying connection was closed: An unexpected error occurred on a send, make sure you have .Net Framework 4.5 or later installed. Also try to enable TLS 1.2 protocol by running the following command:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

Installing the command line interface by using Homebrew

You can also install the IBM Cloud CLI by using Homebrew. For more information, see Homebrew ibm-cloud-cli.

Installing to a custom directory

When you use installers or a shell script to install the IBM Cloud CLI, it is installed in your system directories. If you want to specify a different directory, use the following steps.

1. Use a browser to access the official ibm-cloud-cli-releases GitHub repository, and select the matching binary of your platform to begin the download. The following platforms are supported: macOS, linux32, linux64, ppc64le, win32, and win64.
2. Extract the package to a directory that you specify.
   You can see the following extracted content:
   For Linux™ and Mac:

```
IBM_Cloud_CLI
├── LICENSE
├── NOTICE
├── autocomplete
│   ├── bash_autocomplete
│   └── zsh_autocomplete
├── ibmcloud
└── ibmcloud-analytics
```

   For Windows™:

```
IBM_Cloud_CLI
├── LICENSE
├── NOTICE
├── ibmcloud-analytics.exe
└── ibmcloud.exe
```

3. Add to the PATH environment variable and enable shell autocompletion.
    - Add the {YOUR_DIRECTORY}/IBM_CLOUD_CLI to the PATH environment variable.
    - For shell autocompletion support (Mac and Linux™ only), see Enabling shell autocompletion for IBM Cloud CLI.

Updating the IBM Cloud CLI

You must use the latest version of the CLI. If you aren't using the latest version, run the following command to update your CLI:

```
ibmcloud update
```

To determine your IBM Cloud CLI version, run the following command:

```
ibmcloud -v
```

If you are running the current release, the following output is displayed:

```
Checking for updates...
No update required. Your CLI is already up-to-date.
```

Installing the tools and plug-ins manually

You can manually install the IBM Cloud® Command Line Interface and other plug-ins and tools for developing applications for IBM Cloud.

Before you begin

- Install the stand-alone IBM Cloud CLI to get support for installing command line plug-ins for IBM Cloud.
- Install the curl command for downloading packages through the command line.

Installing Docker

To run and debug apps locally, install Docker.

Installing the Kubernetes command line tool

To view a local version of the Kubernetes dashboard, and to deploy apps into your clusters, install the Kubernetes command line tool for your platform:

- Mac:

```
curl --progress-bar -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/darwin/amd64/kubectl
```

- Linux™:

```
curl --progress-bar -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl
```

- Windows™:

```
curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.7.0/bin/windows/amd64/kubectl.exe
```

The prefix for running commands by using the Kubernetes command line tool is kubectl. For more information, see Setting up the CLI and API.

Installing IBM Cloud Object Storage CLI plug-in

The IBM Cloud Object Storage plug-in extends the IBM Cloud command line interface (CLI) with an API wrapper for working with Object Storage resources.

- To install the IBM Cloud Object Storage plug-in, run the following command:

```
ibmcloud plugin install cloud-object-storage
```

Installing IBM Cloud Container Registry CLI plug-in

You can use the container-registry CLI plug-in to set up your own image namespace in an IBM-hosted, and managed, private registry. Where you can store and share Docker images with all users in your IBM Cloud account.

- To install the IBM Cloud® Container Registry plug-in, run the following command:

```
ibmcloud plugin install container-registry
```

Installing IBM Cloud Kubernetes Service CLI plug-in

To create and manage Kubernetes clusters in IBM Cloud® Kubernetes Service:

- To install the Kubernetes Service plug-in, run the following command:

```
ibmcloud plugin install container-service
```

Installing Helm

Install Helm, which is a Kubernetes-based package manager.

- Mac and Linux™ users, run the following commands:

```
curl -sL https://raw.githubusercontent.com/kubernetes/helm/master/scripts/get | bash
```

- Windows™ users can download and install the Helm binary.

Installing the Cloud Functions CLI plug-in

You can use the IBM Cloud® Functions CLI plug-in to manage your code snippets in actions, bundle actions into packages, and create triggers and rules to enable your actions to respond to events.

- To install the Cloud Functions CLI plug-in, run the following command:

```
ibmcloud plugin install cloud-functions
```

Installing IBM Cloud Schematics CLI plug-in

To create and manage Schematics in IBM Cloud Schematics service:

- To install the IBM Cloud Schematics plug-in, run the following command:

```
ibmcloud plugin install schematics
```

Installing Git

You can download and install Git.

Installing and using private endpoints

To ensure that you have enhanced control and security over your data when you use the IBM Cloud Command Line Interface (CLI), you have the option of using private routes to IBM Cloud endpoints. Private routes are not accessible or reachable over the internet. By using the IBM Cloud private endpoints feature, you can protect your data from threats from the public network and logically extend your private network.

The CLI uses the private endpoint support that is provided by the IBM Cloud platform. Platform services that are used by the core CLI provide private endpoint support.

Before you begin

You must first enable virtual routing and forwarding in your account, and then you can enable the use of IBM Cloud private service endpoints. For more information about setting up your account to support the private connectivity option, see Enabling VRF and service endpoints. In addition to enabling VRF, you must be on a private network on ibmcloud. For more information, see how to connect to the IBM Cloud VPN.

To learn more about private connections on IBM Cloud, see Secure access to services by using service endpoints.

Installing from the shell

To install the latest CLI for your OS from the shell manually, use the following command for your OS:

- For Mac, copy and paste the following command to a command line and run it:

```
curl -fsSL https://ibm-cloud-cli-installer-scripts.s3.private.us.cloud-object-storage.appdomain.cloud/osx_private | sh
```

- For Linux™, copy and paste the following command to a command line and run it:

```
curl -fsSL https://ibm-cloud-cli-installer-scripts.s3.private.us.cloud-object-storage.appdomain.cloud/linux_private | sh
```

- For Windows™, copy and paste the following command to a Windows™ PowerShell command prompt and run it:

```
iex(New-Object Net.WebClient).DownloadString('https://ibm-cloud-cli-installer-scripts.s3.private.us.cloud-object-storage.appdomain.cloud/powershell_private')
```

If you encounter errors like The underlying connection was closed: An unexpected error occurred on a send, make sure you have .Net Framework 4.5 or later installed. Also, try to enable TLS 1.2 protocol by running the following command:

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

Downloading installation packages with private endpoints

Use the following steps to install the latest stand-alone IBM Cloud CLI:

1. Select the installer of your OS to begin the download. The following operating systems are supported: macOS X 64-bit, Windows™ 64-bit, Linux™ x86 64-bit, Linux™ LE 64-bit (ppc64le), and System/390 Linux™.
2. Run the installer:

- For Mac and Windows™, run the installer.
- For Linux™, extract the package and run the install script.

Installers:

- Mac OS X 64-bit
- Mac OS X M1/ARM (provided as-is)
- Windows™ 64-bit
- Windows™ 32-bit
- Linux™ 64-bit
- Linux™ 32-bit
- PowerLinux™ 64-bit
- System/390 Linux™ 64-bit

3. Log in to IBM Cloud by using the CLI:

```
ibmcloud login
```

Now, you're ready to manage IBM Cloud resources. Enter ibmcloud help to view the command descriptions.

Installing packages from previous releases for private endpoints

If you want to install these packages from previous releases, follow this template to construct the private endpoint link of the package that you want to download, replacing release_tag with the release version that you want to download.

To view previous releases, access the official ibm-cloud-cli-releases GitHub repository.

- Mac OS X 64-bit

```
https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/IBM_Cloud_CLI_<release_tag>.pkg
```

- Windows™ 64-bit

```
https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/IBM_Cloud_CLI_<release_tag>_amd64.exe
```

- Windows™ 32-bit

```
https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/IBM_Cloud_CLI_<release_tag>_386.exe
```

- Linux™ 64-bit

```
https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/IBM_Cloud_CLI_<release_tag>_amd64.tar.gz
```

- Linux™ 32-bit

```
https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/IBM_Cloud_CLI_<release_tag>_386.tar.gz
```

- PowerLinux™ 64-bit

```
https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/IBM_Cloud_CLI_<release_tag>_ppc64le.tar.gz
```

- System/390 Linux™ 64-bit

```
https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/IBM_Cloud_CLI_<release_tag>_s390x.tar.gz
```

Installing to a custom directory

When you use installers or a shell script to install the IBM Cloud CLI, it is installed in your system directories. If you want to specify a different directory, use the following steps.

1. Select the matching binary of your platform to begin the download. The following platforms are supported: macOS, linux32, linux64, ppc64le, win32, win64, and s390x.
2. Extract the package to a directory that you specify.
   You can see the following extracted content:
   For Linux™ and Mac:

```
IBM_Cloud_CLI
├── LICENSE
├── NOTICE
├── autocomplete
│   ├── bash_autocomplete
│   └── zsh_autocomplete
└── ibmcloud
```

For Windows™:

```
IBM_Cloud_CLI
├── LICENSE
├── NOTICE
└── ibmcloud.exe
```

3. Add to the PATH environment variable and enable shell autocompletion.
   - Add the {YOUR_DIRECTORY}/IBM_CLOUD_CLI to the PATH environment variable.
   - For shell autocompletion support (Mac and Linux™ only), see Enabling shell autocompletion for IBM Cloud CLI.

Binary files

- Mac OS X 64-bit
- Windows™ 64-bit
- Windows™ 32-bit
- Linux™ 64-bit
- Linux™ 32-bit
- PowerLinux™ 64-bit
- System/390 Linux™ 64-bit

If you want to install binaries from previous releases, follow this template to construct the private endpoint link of the binary that you want to download, replacing release_tag with the release version that you want to download.

To view previous releases, access the official ibm-cloud-cli-releases Github repository.

- Mac OS X 64-bit

```
https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/binaries/IBM_Cloud_CLI_<release_tag>_macos.tgz
```

- Windows 64-bit

  https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/binaries/IBM_Cloud_CLI_<release_tag>_windows_amd64.zip

- Windows 32-bit

  https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/binaries/IBM_Cloud_CLI_<release_tag>_windows_386.zip

- Linux 64-bit

  https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/binaries/IBM_Cloud_CLI_<release_tag>_linux_amd64.tgz

- Linux 32-bit

  https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/binaries/IBM_Cloud_CLI_<release_tag>_linux_386.tgz

- Power Linux 64-bit

  https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/binaries/IBM_Cloud_CLI_<release_tag>_linux_ppc64le.tgz

- System/390 Linux 64-bit

  https://ibm-cloud-cli.s3.private.us.cloud-object-storage.appdomain.cloud/<release_tag>/binaries/IBM_Cloud_CLI_<release_tag>_linux_s390x.tgz

Using private endpoints in the CLI

For instructions on how to use private endpoints in the CLI, and to view the list of commands that support private endpoints.

Extending IBM Cloud CLI with plug-ins

The IBM Cloud® Command Line Interface supports a plug-in framework to extend its capability. You can install a plug-in from a repository, a web URL, or install a plug-in binary locally.

Installing a plug-in from the IBM Cloud CLI repository

Searching for a plug-i

Use the ibmcloud plugin repo-plugins -r REPO_NAME command to look for a plug-in in the repository.

The IBM Cloud CLI provides an official plug-in repository with the name 'IBM Cloud', which you can search as shown in the following example:

```
ibmcloud plugin repo-plugins -r "IBM Cloud"
```

Status	Name	Versions	Description
Update Available	container-service/ kubernetes-service	0.3.49, 0.3.47, 0.3.34...	IBM Cloud Kubernetes Service for management of Kubernetes clusters
Update Available	cloud-functions	1.0.32, 1.0.30, 1.0.29...	Manage Cloud Functions

...

Installing the plug-in

Use the ibmcloud plugin install PLUGIN_NAME -r REPO_NAME command to install the plug-in. For example, use the following command to install a plug-in from the official IBM plug-in repo 'IBM Cloud':

```
ibmcloud plugin install code-engine

Looking up 'code-engine' from repository 'IBM Cloud'...

Plug-in 'code-engine 1.23.2' found in repository 'IBM Cloud'

Attempting to download the binary file...
```

```
 54.29 MiB / 54.29 MiB [=============================================] 100.00% 10s

56929376  bytes downloaded

Installing binary...

OK

Plug-in 'code-engine 1.23.2' was successfully installed into /Users/username/.bluemix/plugins/code-engine. Use 'ibmcloud plugin show code-engine' to show its details.
```

Installing a plug-in locally

Use the ibmcloud plugin install LOCAL_FILE_NAME command to install a plug-in binary on your local machine. For example

```
ibmcloud plugin install ./code-engine-darwin-amd64-1.23.2
```

```
Installing plugin './code-engine-darwin-amd64-1.23.2'...

OK

Plug-in 'code-engine 1.23.2' was successfully installed into /Users/username/.bluemix/plugins/code-engine. Use 'ibmcloud plugin show code-engine' to show its details.

$
```

Installing a plug-in from a web URL

Use the ibmcloud plugin install URL command to install a plug-in directly from a web URL. For example:

```
ibmcloud plugin install http://example.com/downloads/my-plugin
```

Working with multiple sessions with the IBM Cloud CLI

You can use the IBMCLOUD_HOME environment variable to work with multiple sessions at the same time in the IBM Cloud® CLI.

With the IBMCLOUD_HOME environment variable, you can set a path for the session metadata. You can specify a different directory for each session, and that's helpful when you want to run multiple IBM Cloud CLI sessions simultaneously.

Before you begin

- Set up your IBM Cloud® account. For more information, see Setting up your IBM Cloud account.
- Install the IBM Cloud CLI.

Setting the IBMCLOUD_HOME environment variable

To run multiple IBM Cloud CLI sessions in parallel, use different environments per session. You need to set up a different configuration directory for each session.

For your first environment, enter the following command:

```
export IBMCLOUD_HOME=/home/myuser/.ibmcloudenv1
ibmcloud login
```

For your second environment, enter the following command:

```
export IBMCLOUD_HOME=/home/myuser/.ibmcloudenv2
ibmcloud login
```

For more information about environment variables for the IBM Cloud CLI, see Tips and Tricks for Using the IBM Cloud CLI and IBM Cloud CLI (ibmcloud) environment variables.

Uninstalling the stand-alone IBM Cloud CLI

Use the following steps to uninstall the stand-alone IBM Cloud® Command Line Interface on specific platforms.

Uninstalling on Windows

i. Click the Start button, and then select Control Panel.

ii. In the pop-up window, click Uninstall a program.

iii. In the pop-up application list, locate IBM Cloud Command Line Interface.

iv. Right-click IBM Cloud Command Line Interface, and select Uninstall.

v. The uninstaller is started. Follow the instructions to finish the uninstallation.

Uninstalling on Linux and macOS

The uninstallation steps are different depending on the version of the CLI that is installed.

i. Check your IBM Cloud CLI version by running the following command:

```
ibmcloud -v
```

ii. Run the uninstallation commands for your version of the CLI.

- To uninstall versions 0.9.0 and later, run the following command:

```
/usr/local/ibmcloud/uninstall
```

- To uninstall versions earlier than 0.9.0, run all of the following commands:

```
rm -rf /usr/local/ibmcloud
rm -f /usr/local/bin/ibmcloud
rm -f /usr/local/bin/bluemix
rm -f /usr/local/bin/bx
rm -f /usr/local/bin/ibmcloud-analytics
```

iii. Clean up any autocompletion scripts, if you configured them. For more information, see Enabling shell autocompletion.

Creating and deploying apps by using the CLI

To create and deploy an application using the Developer Tools (ibmcloud dev) commands that are included with the IBM Cloud® Command Line Interface. The ibmcloud dev commands let you use a starter kit or cloud-enable your existing app code.

Objectives

- Create an app using the CLI commands.
- Build and run the app.
- Add a service to the app/modify the app.
- Deploy the app.
- View the app.

Before you begin

- You must install the IBM Cloud CLI and other recommended plug-ins and tools. For more information, see Getting started with the IBM Cloud CLI and Developer Tools commands.
- Docker is installed as part of the developer tools. Docker must be running for the build commands to work. You must create a Docker account, run the Docker app, and sign in.
- If you plan to deploy your app to a Kubernetes or OpenShift cluster, you must create a cluster. For more information, see Deploying apps to Kubernetes clusters or Deploying apps in OpenShift clusters.
- If you plan to deploy to Cloud Foundry, use ibmcloud target --cf to target the Cloud Foundry org and space interactively, or use ibmcloud target --cf-api ENDPOINT -o ORG -s SPACE to target the specific org and space.

Step 1: Creating an app

Creating an app from a starter kit

Creating an app from a starter kit is useful if you don't already have existing code to begin with and would rather start from a language or a framework starter template.

i. Run the ibmcloud dev create command in the directory of your choice.

ii. Select an application type of either Backend Service / Web App or Mobile App.

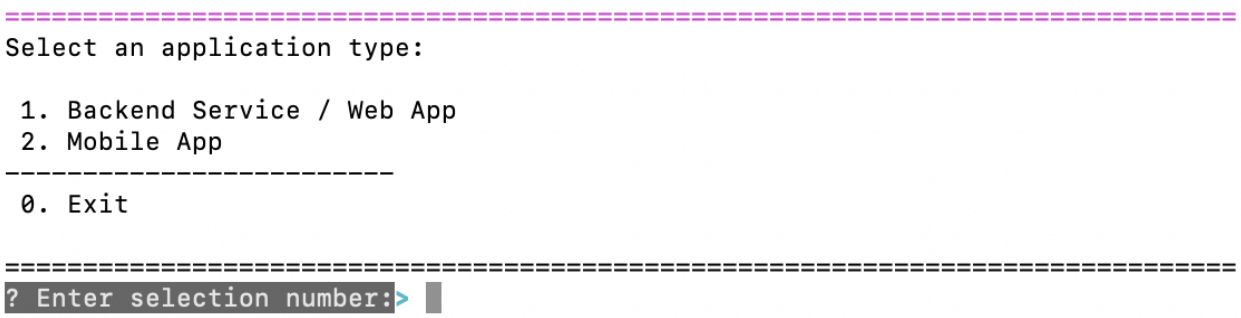


Figure 1. App types in the CLI
Reference: https://cloud.ibm.com/docs-content/v1/content/37be3ab15a6596376238e8d541867070757c1c4b/apps/images/cli-select-apptype.png

iii. Select a language.

Figure 2. Languages in the CLI
Reference: https://cloud.ibm.com/docs-content/v1/content/37be3ab15a6596376238e8d541867070757c1c4b/apps/images/cli-select-lang.png

iv. Select a starter kit to use as the basis for your app.

v. Enter a name for your app, and select the resource group that you want to use (if necessary).

vi. Optional. Select a service to add to your app. During the app creation process, you are prompted if you want to add a service to your app.

- Enter Y.
- Select Create a new service and add it to this application.
- Follow the prompts to select a service group, service, region, and pricing plan.

vii. Select a deployment option. If you want to deploy the app automatically, select a DevOps option. Otherwise, select the manual deployment option.

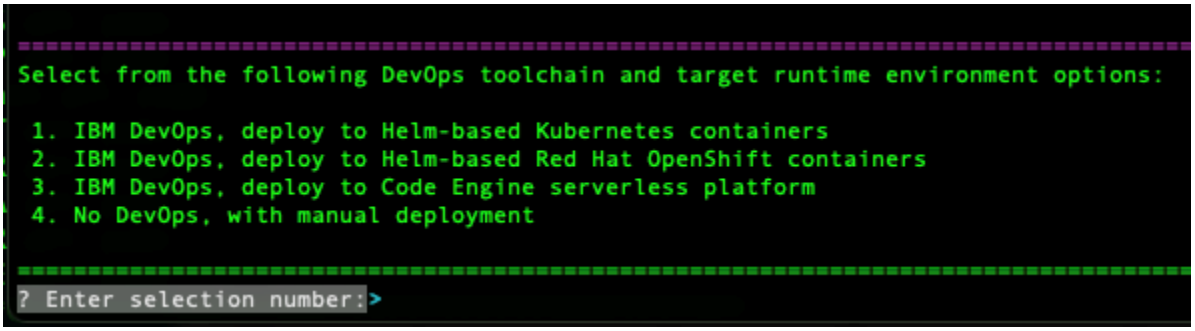


Figure 3. Deployment options in the CLI

Reference: https://cloud.ibm.com/docs-content/v1/content/37be3ab15a6596376238e8d541867070757c1c4b/apps/images/cli-deploy-options.png

viii. If you selected a DevOps deployment option, follow the remaining prompts to select a region for your toolchain, enter a name for the DevOps toolchain, and enter a host name. Otherwise, follow the prompts for a manual deployment.

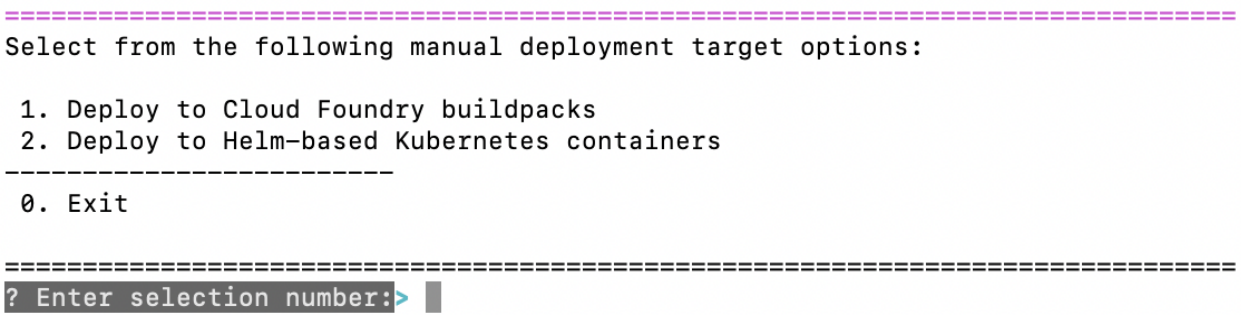


Figure 4. Manual deployment options in the CLI

Reference: https://cloud.ibm.com/docs-content/v1/content/37be3ab15a6596376238e8d541867070757c1c4b/apps/images/cli-manual-deploy-options.png

Creating the app and toolchain takes a few seconds to complete. The app is created in the current directory. Only the deployment files that are relevant for your choice of deployment target are created. You can use the ibmcloud dev edit command from the app directory to add more deployment file types if you need them.

Creating an app from your own code

This option can be used if you already have an existing codebase and want to generate deployment and cloud enablement assets for a single microservice or web app by using the ibmcloud dev enable command. This command is in beta, and not all languages or app structures are supported. Complete the following steps to use this function with a sample repository.

i. Log in to IBM Cloud by running the ibmcloud login command, and then target an org and space.

ii. Clone the Hello World sample app by running the following command in the directory of your choice.

```
git clone https://github.com/IBM-Cloud/node-helloworld.git
```

iii. Navigate to the directory where you cloned the sample app, and run the ibmcloud dev enable command.

iv. Select to continue without committing changes for now (if necessary).

v. Select to continue when you're prompted to proceed with the Node language that is detected.

vi. Select the resource group that you want to use (if necessary).

vii. Select the option to create a new IBM Cloud app that is linked to this Git repository. See Important Notes for details.

viii. Choose not to add services for now.

ix. Select a deployment option. If you want to deploy the app automatically, select a DevOps option. Otherwise, select the manual deployment option.

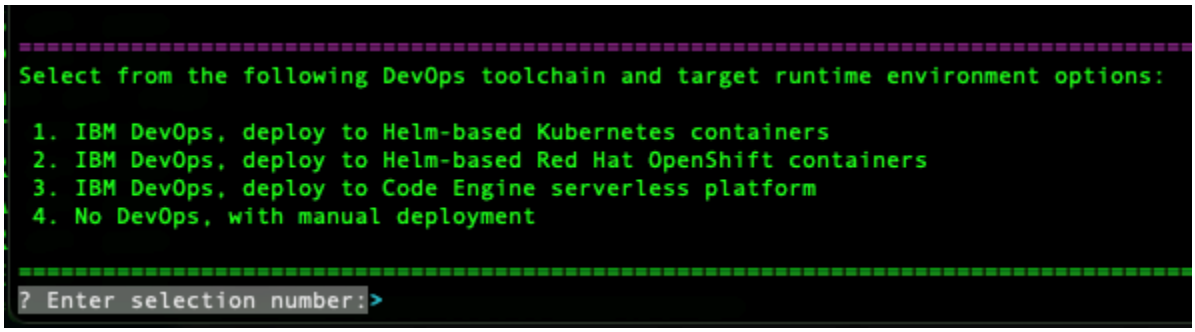


Figure 3. Deployment options in the CLI
Reference: https://cloud.ibm.com/docs-content/v1/content/37be3ab15a6596376238e8d541867070757c1c4b/apps/images/cli-deploy-options.png

x. If you selected a DevOps deployment option, follow the remaining prompts to select a region for your toolchain, enter a name for the DevOps toolchain, and enter a host name. Otherwise, follow the prompts for a manual deployment.

```
================================================================================
Select from the following manual deployment target options:

 1. Deploy to Cloud Foundry buildpacks
 2. Deploy to Helm-based Kubernetes containers
--------------------------
 0. Exit

================================================================================
? Enter selection number:>
```

Figure 4. Manual deployment options in the CLI
Reference: https://cloud.ibm.com/docs-content/v1/content/37be3ab15a6596376238e8d541867070757c1c4b/apps/images/cli-manual-deploy-options.png

xi. Wait a few seconds for the operations to complete.

xii. After the operations are complete, manually merge the deployment and cloud enablement files that are saved to the app directory. Merge new files marked .merge by using git diff or a similar tool.

- If you already created an IBM Cloud app by using the IBM Cloud console, follow steps 2 - 5 in the previous section in your app directory. For step 6, you can select the option to connect your local code to an existing app.
- To manually configure a toolchain and deployment files, see Continuous Deployment to Kubernetes. This tutorial can be useful if you're trying to configure a Continuous Delivery toolchain for more than one interrelated web apps or microservices.
- If your existing codebase isn't already in a Git repository, follow steps 2 - 5 in the previous section in your app directory. For step 6, you can select the option create a new IBM Cloud app, and deploy it to a DevOps toolchain (which has a newly created GitLab repository).

Step 2: Building your app and running it locally

Regardless of which option you used to create your app, you can now build it and run it locally.

i. Navigate to your app directory, and ensure that Docker is running on your system.

ii. Run the ibmcloud dev build command to build your app.

iii. Run the ibmcloud dev run command to start the app in the foreground. To stop the app and return the command prompt, press Ctrl+C.

iv. View your app that is running locally by navigating to http://localhost:3000 or a similar URL.

Step 3: Adding a service and modifying the code

Now that your app can run locally, you can add a service and modify some code.

i. Run the ibmcloud dev edit command from the app directory.

ii. Follow the prompts to create and connect a new data-related service to your app, such as IBM Cloudant. You might need to select a region and plan for the service.

iii. You can choose to manually merge the configuration files that are saved to your app directory when you create the service. Or you can skip this step for now.

iv. Update your code. For example, modify the /public/index.html file or a similar file. If you're using the sample ExpressJS app, you can change the Congratulations! message to something like Hello World!.

v. Save any files that you modified.

Step 4: Deploying your app

IBM Cloud provides a robust CLI and Developer Tools (ibmcloud dev) commands to help simplify the developer's workflow. You can deploy your IBM Cloud app in one of two ways, depending on how your app is configured.

i. Change to the directory where your app code is located.

ii. Update your app code, if necessary. For example, if you're using an IBM Cloud sample app and your app contains the src/main/webapp/index.html file, you can modify it and edit the Thanks for creating ... line. Ensure that the app runs locally before you deploy it to IBM Cloud.

Review the README.md file, which contains details, such as build instructions.

i. Log in to the IBM Cloud CLI with your IBMid. If you have multiple accounts, you are prompted to select which account to use. If you do not specify a region with the -r flag, you must also select a region.

```
ibmcloud login
```

If your credentials are rejected, you might be using a federated ID. To log in with a federated ID, use the --sso flag. For more information, see Logging in with a federated ID.

Deploying your app automatically

If you didn't create a DevOps toolchain for your app and your app isn't yet in a Git repository, you can run the ibmcloud dev edit command from the app directory. Follow the prompts for "Configure DevOps" and deploy to a new toolchain (and create a new GitLab repository).

After you create a DevOps toolchain for your app, deploying a new build is as simple as committing and pushing your code to the repository in your toolchain.

i. Prepare the changes to be committed

```
git add .
```

ii. Commit the changes with a brief message.

```
git commit -m "made changes"
```

iii. Push the commits on the master branch to the remote repository.

```
git push origin master
```

iv. View the DevOps toolchain for your app from the IBM Cloud console. You can view toolchain details from the App details page in the IBM Cloud console by running the ibmcloud dev console command from the app directory.

v. View the pipeline within the toolchain to verify that a new build started.

Deploying your app manually

When you use ibmcloud dev create, you're prompted to choose between a DevOps deployment or a manual deployment. When you choose the manual deployment option, the app is created and saved into your current directory, but it's not automatically deployed.

You can manually deploy your app to IBM Cloud by completing the following steps:

i. Run the ibmcloud dev deploy command.

For Cloud Foundry:

```
ibmcloud dev deploy
```

For Kubernetes:

```
ibmcloud dev deploy -t container
```

If you want to deploy your app to a different manual deployment type, run ibmcloud dev edit from the app directory, and add the other deployment files.

Step 5: Viewing your app

i. To view the URL of your app that's running on IBM Cloud, run the ibmcloud dev view command from the app directory. The app URL is opened in your default browser.

ii. To view details about your app's credentials, services, and toolchain from the IBM Cloud console, run the ibmcloud dev console command.

Best practices for successful cloud implementation

With the cloud offering so many benefits, it is natural for you to want to embrace the cloud right away. But is your organization ready for successful cloud implementation? Choosing to implement the cloud for your business is a game-changing move and that's why best practices related to implementation should not be taken lightly. That said, here are 10 cloud best practices for a successful cloud implementation:

i. Start With an End-to-End Assessment

The first step towards a successful cloud implementation is assessment. Understanding where you are today and where you want to reach through the cloud is critical. Start with assessing your current environment and envisioning the future state; identify organizational readiness, risks, opportunities, and costs with moving to the cloud.

ii. Adopt a Cloud-Based-as-a-Service Business Model

With cloud paving the way for an effective as-a-service model, it makes sense to begin your cloud journey by implementing and leveraging business applications as a service – to drive growth. Through the as-a-service model, you can use cloud resources based on your needs and accelerate the use of as-a-service, as your needs grow in the future.

iii. Devise an All-Encompassing Adoption Strategy

Devising a robust cloud adoption strategy can aid in a faster implementation with less risk. To build a comprehensive approach to cloud computing across your organization, consider the various aspects that will be impacted by the adoption: namely your business, your people, the governance strategy, security and platform considerations as well as day-to-day operations.

iv. Educate and Train Resources as Early as Possible

The success or failure of your cloud implementation also depends on how well-educated your users are. Since it is these users who will be performing day-to-day tasks using the

cloud, it is essential you provide them with in-depth training. This is not only to ensure they understand the importance and benefits of cloud adoption but also to reduce potential cloud adoption barriers.

v. Choose the Right Model

Successful cloud implementation also requires you to choose a cloud model that's right for your business. Given the various models available, it is critical you understand the need and approach for each and select one that best fits your needs. For example, with IaaS, you can get access to virtualized computing resources over the internet. With PaaS, you can have a third-party provider deliver hardware and software tools for application development. And with SaaS, you can access software online via a subscription, rather than buy and install it on individual computers.

vi. Plan and Follow a Cloud Governance Framework

The success of your cloud implementation will also depend on your ability to ensure accountability, control, and governance. Developing a cloud governance framework allows for the secure adoption of services and features. It ensures that the right programs, systems, and tools are in place to move workloads to the cloud. It also enforces the correct policies to ensure your employees are trained well enough to use the cloud effectively and safely.

vii. Select a Single Workload

Although it might seem lucrative to move all your workloads to the cloud in one go, given the high cost and risks involved, it makes most business sense to start small. Select a single function or workload that you think will benefit most and move it to the cloud – slowly expanding to additional workloads as the model is proven.

viii. Automate as Much as Possible

Given how intricate and time-consuming cloud implementation can be, it is advisable to automate as many aspects as possible – to reduce toil as well as operating costs. Consider automating the provisioning, configuration, and management of your cloud-based infrastructure and free up precious time and resources to minimize disruption and drive mission-critical innovation.

ix. Leverage the Services of an Experienced Partner

Cloud implementation is a complex undertaking and requires organizations to take each step towards it with utmost planning. Since even a single wrong step can result in a catastrophe, leveraging the knowledge and experience of experienced service providers can help improve delivery and reduce roadblocks. Highly capable partners can provide the insight and direction you need to streamline your cloud adoption journey.

x. Monitor and Optimize

The journey to the cloud doesn't end at implementation. To ensure long-lasting returns, organizations must invest in evaluating, monitoring, and managing cloud-based services, applications, and infrastructure. Invest in modern monitoring tools to drive cost optimization, performance improvements as well as security compliance.

Embrace Cloud Implementation

With the next wave of intelligent technologies that build onto the cloud emerging, the cloud is set to drive continuous innovation while changing the way customers interact with companies. Not only does cloud-free up IT resources, but it also results in performance improvements, reduced business risks, and better economies of scale.

However, the journey to the cloud is laden with obstacles; what is required is to have the right approach and strategies to emerge as winners. Embracing cloud best practices is a great way to reduce the risks associated with implementation while improving performance and reducing costs.

Best practices for cloud security

Cloud security is constantly evolving, but a handful of best practices have remained constant for ensuring the security of cloud environments. Organizations that have existing cloud solutions in place or are looking to implement them should consider these tips and tools to ensure that sensitive applications and data don't fall into the wrong hands.

i. Understand your shared responsibility model
ii. Ask your cloud provider detailed security questions
iii. Deploy an identity and access management solution
iv. Train your staff
v. Establish and enforce cloud security policies

vi. Secure your endpoints
vii. Encrypt data in motion and at rest
viii. Use intrusion detection and prevention technology
ix. Double-check your compliance requirements
x. Consider a CASB vendor
xi. Conduct audits and penetration testing
xii. Enable security logs

i. Understand Your Shared Responsibility Model

- In a private data center, the enterprise is solely responsible for all security issues. But in the public cloud, things are much more complicated. While the buck ultimately stops with the cloud customer, the cloud provider assumes the responsibility for some aspects of IT security. Cloud and security professionals call this a shared responsibility model.
- Leading IaaS and platform as a service (PaaS) vendors like Amazon Web Services (AWS) and Microsoft Azure provide documentation to their customers so all parties understand where specific responsibilities lie according to different types of deployment. The diagram below, for example, shows that application-level controls are Microsoft's responsibility with software as a service (SaaS) models, but it is the customer's responsibility in IaaS deployments. For PaaS models, Microsoft and its customers share the responsibility.

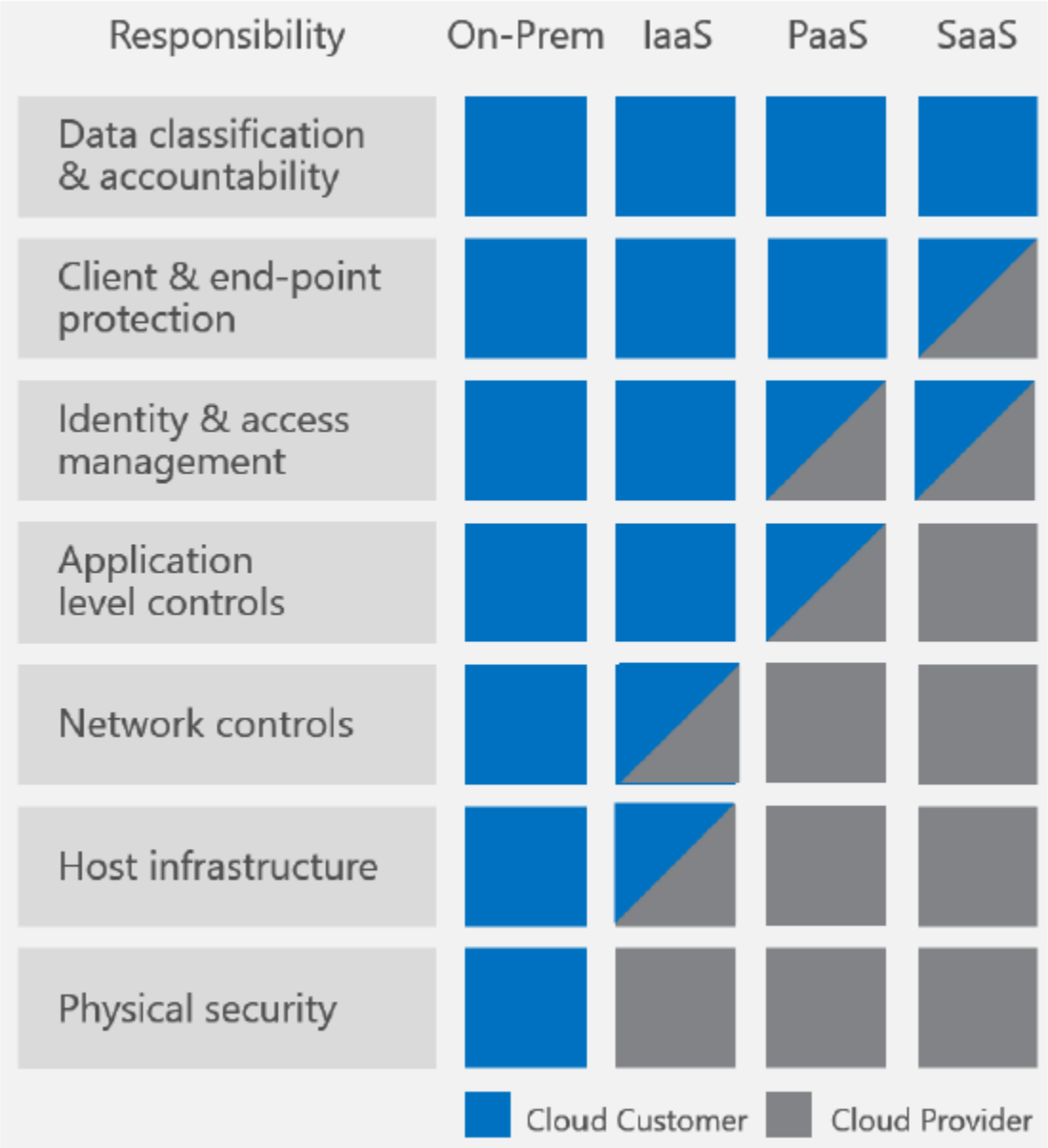


Figure: Understand Your Shared Responsibility Model
Reference: https://lh3.googleusercontent.com/92PwNkl5hzVmE3_Uq0cwvG2u2q5mBXMha-MT1zUsURtDjnoHCaw4YJPNArdcnA5-Wavql3eusE1b3bKya5AGeromfHLTXyOkhsPx9Y0zYMXKv6Q_gB_fgrcPTszc40lLpOfv0VBT=s0

- Enterprises that are considering a particular cloud vendor should review its policies about shared security responsibilities and understand who is handling the various aspects of cloud security. That can help prevent miscommunication and misunderstanding. More importantly, though, clarity about responsibilities can prevent security incidents that occur as a result of a particular security need falling through the cracks.

ii. Ask Your Cloud Provider Detailed Security Questions

- In addition to clarifying shared responsibilities, organizations should ask their public cloud vendors detailed questions about the security measures and processes they have in place. It's easy to assume that the leading vendors have security handled, but security methods and procedures can vary significantly from one vendor to the next.
- To understand how a particular cloud provider compares, organizations should ask a wide range of questions, including:
    - Where do the provider's servers reside geographically?
    - What is the provider's protocol for suspected security incidents?
    - What is the provider's disaster recovery plan?
    - What measures does the provider have in place to protect various access components?
    - What level of technical support is the provider willing to provide?
    - What are the results of the provider's most recent penetration tests?
    - Does the provider encrypt data while in transit and at rest?
    - Which roles or individuals from the provider have access to the data stored in the cloud?
    - What authentication methods does the provider support?
    - What compliance requirements does the provider support?

iii. Deploy An Identity And Access Management Solution

- The fourth biggest threat to public cloud security identified in CloudPassage's report is unauthorized access (and growing – 53 percent, up from 42 percent in 2020). While hackers' methods of gaining access to sensitive data are becoming more sophisticated with each new attack, a high-quality identity and access management (IAM) solution can help mitigate these threats.
- Experts recommend that organizations look for an IAM solution that allows them to define and enforce access policies based on least privilege. These policies should also be based on role-based permission capabilities. Additionally, multi-factor authentication (MFA) can further

reduce the risk of malicious actors gaining access to sensitive information, even if they manage to steal usernames and passwords.

- Organizations may also want to look for an IAM solution that works in hybrid environments that include private data centers as well as cloud deployments. This can simplify authentication for end users and make it easier for security staff to ensure that they are enforcing consistent policies across all IT environments.

iv. Train Your Staff

- To prevent hackers from getting their hands on access credentials for cloud computing tools, organizations should train all workers on how to spot cybersecurity threats and how to respond to them. Comprehensive training should include basic security knowledge like how to create a strong password and identify possible social engineering attacks as well as more advanced topics like risk management.
- Perhaps most importantly, cloud security training should help employees understand the inherent risk of shadow IT. At most organizations, it's all too easy for staff to implement their own tools and systems without the knowledge or support of the IT department. Without top-to-bottom visibility of all systems that interact with the company's data, there's no way to take stock of all vulnerabilities. Enterprises need to explain this risk and hammer home the potential consequences for the organization.
- Organizations also need to invest in specialized training for their security staff. The threat landscape shifts on a daily basis, and IT security professionals can only keep up if they are constantly learning about the newest threats and potential countermeasures.

v. Establish And Enforce Cloud Security Policies

- All organizations should have written guidelines that specify who can use cloud services, how they can use them, and which data can be stored in the cloud. They also need to lay out the specific security technologies that employees must use to protect data and applications in the cloud.
- Ideally, security staff should have automated solutions in place to ensure that everyone is following these policies. In some cases, the cloud vendor may have a policy enforcement feature that is sufficient to meet the organization's needs. In others, the organization may need to purchase a third party solution like CASB that offers policy enforcement capabilities.
- Zero trust is one such technology that offers a refined control over policy enforcement. Tools in this category work with other systems to determine how much access each user needs, what they can do with that access, and what it means for the broader organization.

vi. Secure Your Endpoints

- Using a cloud service doesn't eliminate the need for strong endpoint security—it intensifies it. New cloud computing projects offer an opportunity to revisit existing strategies and ensure the protections in place are adequate to address evolving threats.
- A defense-in-depth strategy that includes firewalls, anti-malware, intrusion detection, and access control has long been the standard for endpoint security. However, the array of endpoint security concerns has become so complex that automation tools are required to keep up. Endpoint detection and response (EDR) tools and/or endpoint protection platforms (EPP) can help in this area.
- EDR and EPP solutions combine traditional endpoint security capabilities with continuous monitoring and automated response. Specifically, these tools address a number of security requirements, including patch management, endpoint encryption, VPNs, and insider threat prevention among others.

vii. Encrypt Data In Motion And At Rest

- Encryption is a key part of any cloud security strategy. Not only should organizations encrypt any data in a public cloud storage service, but they should also ensure that data is encrypted during transit—when it may be most vulnerable to attacks.
- Some cloud computing providers offer encryption and key management services. Some third-party cloud and traditional software companies offer encryption options as well. Experts recommend finding an encryption product that works seamlessly with existing work processes, eliminating the need for end users to take any extra actions to comply with company encryption policies.

viii. Use Intrusion Detection And Prevention Technology

- Intrusion prevention and detection systems (IDPS) are among some of the most effective cloud security tools on the market. They monitor, analyze, and respond to network traffic across both on-premises and public cloud environments. When they encounter signature-based, protocol-based, or anomaly-based threats, IDPS solutions add them to a log, alert administrators to unusual activity, and block the threats so admins have enough time to take action.
- These tools are important for round-the-clock monitoring and real-time alerts. Without IDPS, it's nearly impossible to analyze network traffic for the telltale signs of a sophisticated attack.

ix. Double-Check Your Compliance Requirements

- Organizations that collect personally identifiable information (PII) like those in retail, healthcare, and financial services face strict regulations when it comes to customer privacy and data security. Some businesses in certain geographic locations—or businesses that store data in particular regions—may have special compliance requirements from local or state governments as well.
- Before establishing a new cloud computing service, organizations should review their particular compliance requirements and make sure that their service provider will meet their data security needs.

x. Consider A CASB Or Cloud Security Solution

- Dozens of companies offer solutions or services specifically designed to enhance cloud security. If an organization's internal security staff doesn't have cloud expertise or if the existing security solutions don't support cloud environments, it may be time to bring in outside help.
- Cloud access security brokers (CASBs) are tools purpose-built to enforce cloud security policies. They have become increasingly popular as more organizations have started using cloud services. Experts say that a CASB solution may make the most sense for organizations that use multiple cloud computing services from several different vendors. These solutions can also monitor for unauthorized apps and access too.

xi. Conduct Audits And Penetration Testing

- Whether an organization chooses to partner with an outside security firm or keep security teams in-house, experts say all enterprises should run penetration testing to determine whether existing cloud security efforts are sufficient to protect data and applications.
- Additionally, organizations should conduct regular security audits that include an analysis of all security vendors' capabilities. This should confirm that they are meeting the agreed upon security terms. Access logs should also be audited to ensure only appropriate and authorized personnel are accessing sensitive data and applications in the cloud.

xii. Enable Security Logs

- In addition to conducting audits, organizations should enable logging features for their cloud solutions. Logging helps system administrators keep track of which users are making changes to the environment—something that would be nearly impossible to do manually. If an attacker

gains access and makes changes, the logs will illuminate all their activities so they can be remediated.

- Misconfigurations are one of the most significant challenges of cloud security, and effective logging capabilities will help connect the changes that led to a particular vulnerability so they can be corrected and avoided in the future. Logging also helps identify individual users who may have more access than they actually need to do their jobs, so administrators can adjust those permissions to the bare minimum.

Cloud security requires the right tools

- Experts emphasize that, in most cases, concerns about security should not prevent organizations from using public cloud services. Often, organizations actually have fewer security issues with cloud-based workloads than with those that run in traditional data centers.
- If one thing is clear from our list of best practices, it's that strong cloud security relies on having the right tools in place. By following cloud security best practices and implementing the appropriate security tools, businesses can minimize risks and take full advantage of the benefits cloud computing offers.

References:

- https://www.javatpoint.com/introduction-to-cloud-computing
- https://www.digitalocean.com/community/tutorials/a-general-introduction-to-cloud-computing
- https://www.futurelearn.com/info/blog/introduction-to-cloud-computing
- https://www.business.qld.gov.au/running-business/it/cloud-computing/benefits
- https://www.ibm.com/cloud/learn/benefits-of-cloud-computing
- https://www.javatpoint.com/cloud-service-models
- https://patterns.arcitura.com/cloud-computing-patterns/basics/cloud-delivery-models/overview
- https://www.redhat.com/en/topics/cloud-computing/public-cloud-vs-private-cloud-and-hybrid-cloud#:~:text=There%20are%204%20main%20types,a%2DService%20(SaaS).
- https://www.ibm.com/in-en/cloud/infrastructure
- https://aws.amazon.com/about-aws/global-infrastructure/
- https://www.ibm.com/cloud/learn/what-is-cloud-database
- http://www.vlss-llc.com/blog/what-is-database-as-a-service-dbaas
- https://www.techopedia.com/definition/26538/cloud-storage-service
- https://cloud.ibm.com/docs/cloud-infrastructure?topic=cloud-infrastructure-compute
- https://cloud.ibm.com/docs/cloud-infrastructure?topic=cloud-infrastructure-network
- https://www.vmware.com/topics/glossary/content/cloud-networking.html#:~:text=Cloud%20networking%20is%20a%20type,provider%2C%20and%20available%20on%20demand.
- https://www.ibm.com/cloud/learn/vpc
- https://www.ibm.com/cloud/cloud-foundry
- https://www.ibm.com/cloud/cloud-foundry#tab_466916
- https://cloud.ibm.com/docs/solution-tutorials?topic=solution-tutorials-mean-stack
- https://cloud.ibm.com/docs/solution-tutorials?topic=solution-tutorials-application-log-analysis#application-log-analysis
- https://www.ibm.com/cloud/learn/cloud-computing-gbl
- https://www.ibm.com/cloud/learn/cloud-storage
- https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-getting-started

- https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-hosting_apps
- https://www.ibm.com/cloud/blog/explore-deployment-strategies-with-ibm-cloud-continuous-delivery
- https://cloud.ibm.com/docs/solution-tutorials?topic=solution-tutorials-multi-region-webapp
- https://cloud.ibm.com/docs/cloud-foundry-public?topic=cloud-foundry-public-deployingapps
- https://cloud.ibm.com/docs/cli?topic=cli-getting-started
- https://cloud.ibm.com/docs/cli?topic=cli-install-ibmcloud-cli
- https://cloud.ibm.com/docs/cli?topic=cli-install-devtools-manually
- https://cloud.ibm.com/docs/cli?topic=cli-cli-private-endpoints
- https://cloud.ibm.com/docs/cli?topic=cli-plug-ins
- https://cloud.ibm.com/docs/cli?topic=cli-ibmcloud-home
- https://cloud.ibm.com/docs/cli?topic=apps-create-deploy-app-cli
- https://synoptek.com/insights/it-blogs/10-best-practices-for-successful-cloud-implementation/
- https://www.esecurityplanet.com/cloud/cloud-security-best-practices/