ATAL BIHARI VAJPAYEE INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR - 474015

Cloud Computing

Case Study : IBM Cloud

Submitted To :

Dr. Neetesh Kumar

Submitted By :

Amal Shaji (2017BCS-010)

Sandarbh Yadav (2017BCS-027)

Vaibhav Garg (2017BCS-038)

Q.1. What do you understand about IBM cloud? What are major services offered by IBM Cloud? Elaborate.

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).

With IBM Cloud IaaS, organizations can deploy and access virtualized IT resources, such as compute power, storage and networking over the internet. For computers, organizations can choose between bare-metal or virtual servers.

With IBM Cloud PaaS which is based on the open source cloud platform Cloud Foundry, developers can use IBM services to create, manage, run and deploy various types of applications for the public cloud, as well as for local or on-premises environments. IBM Cloud supports various programming languages, such as Java, Node.js, PHP and Python and extends to support other languages.

IBM Cloud products and services :

IBM Cloud platform supports access to other IBM tools and services, including IBM Watson and IBM Cloud Functions for serverless computing, as well as those from third-party vendors.

The IBM Cloud Catalog lists over 170 services across categories, including :

Compute : Offers various compute resources, including bare-metal servers, virtual servers, serverless computing and containers, on which enterprises can host their workloads;

Network : Provides cloud networking services, such as a load balancer, a content delivery network (CDN), virtual private network (VPN) tunnels and firewalls;

Storage : Offers object, block and file storage for cloud data;

Management : Provides tools to manage and monitor cloud deployments, such as those for log analysis, automation and Infrastructure as Code (IaC);

Security : Includes services for activity tracking, identity and access management and authentication;

Data management : Provides SQL and NoSQL databases, as well as data querying and migration tools;

Analytics : Offers data science tools such as Apache Spark, Apache Hadoop and IBM Watson Machine Learning, as well as analytics services for streaming data;

Artificial Intelligence (AI) : Uses IBM Watson to deliver services such as machine learning, natural language processing and visual recognition;

Internet of things (IoT) : Includes the IBM IoT Platform, which provides services that connect and manage IoT devices, and analyzes the data they produce;

Mobile : Enables a development team to build and monitor mobile applications and their back-end components;

Developer tools : Includes a command-line interface (CLI), as well as a set of tools for continuous delivery, continuous release and application pipelines;

Blockchain : Provides IBM's Blockchain Platform, a software-as-a-service offering to develop apps, enforce governance and monitor a Blockchain network;

Integration : Offers services to integrate cloud and on-premises systems, or various applications, such as API Connect, App Connect and IBM Secure Gateway;

Migration : Provides tools to migrate apps to the cloud, such as IBM Lift CLI and Cloud Mass Data Migration;

VMware : Enables the migration of VMware workloads into the cloud.

Q.2. What is container concept in Cloud Computing? Explain about IBM Bluemix Container Service.

A **Container** in cloud computing is an approach to operating system virtualization. By this, the user can work with a program and its dependencies using resource procedures that are isolated. The code of the application can be bundled with configurations and dependencies in a systematic manner.

Containers are a lighter-weight, more agile way of handling virtualization. Rather than spinning up an entire virtual machine, a container packages together everything needed to run a small piece of software. The container includes all the code, its dependencies and even the operating system itself. This enables applications to run almost anywhere.

While there is still a great deal of reasons to use VMs, containers provide a level of flexibility that is perfect for the multicloud world. When a development team creates an application, they might not know all the places they’ll need to deploy it. Today, an organization might be running the application on its private cloud, but tomorrow it might need to deploy it on a public cloud from a different provider.

Containerizing applications provides teams the flexibility they need to handle the many software environments of modern IT.

IBM Bluemix is a cloud platform as a service (PaaS) developed by IBM. It supports several programming languages and services as well as integrated DevOps to build, run, deploy and manage applications on the cloud. Bluemix is based on Cloud Foundry open technology and runs on SoftLayer infrastructure. Bluemix supports several programming languages including Java, Node.js, Go, PHP, Swift, Python, Ruby Sinatra, Ruby on Rails and can be extended to support other languages such as Scala through the use of buildpacks.

IBM Bluemix includes IBM's Function as a Service (FaaS) system, or Serverless computing offering, that is built using open source from the Apache OpenWhisk incubator project largely credited to IBM for seeding. This system, equivalent to Amazon Lambda, Microsoft Azure Functions, Oracle Cloud Fn or Google Cloud Functions, allows calling of a specific function in response to an event without requiring any resource management from the developer.

Q.3. Explain the concept of Cloud Foundry in IBM Cloud.

Cloud Foundry is an open source, platform-as-a-service (PaaS) on IBM Cloud that enables you to deploy and scale apps without managing servers. Cloud Foundry's container-based architecture runs apps in any programming language over a variety of cloud service providers. This multi-cloud environment allows developers to leverage the cloud platform that suits specific application workloads and move those workloads as necessary within minutes with no changes to the application.

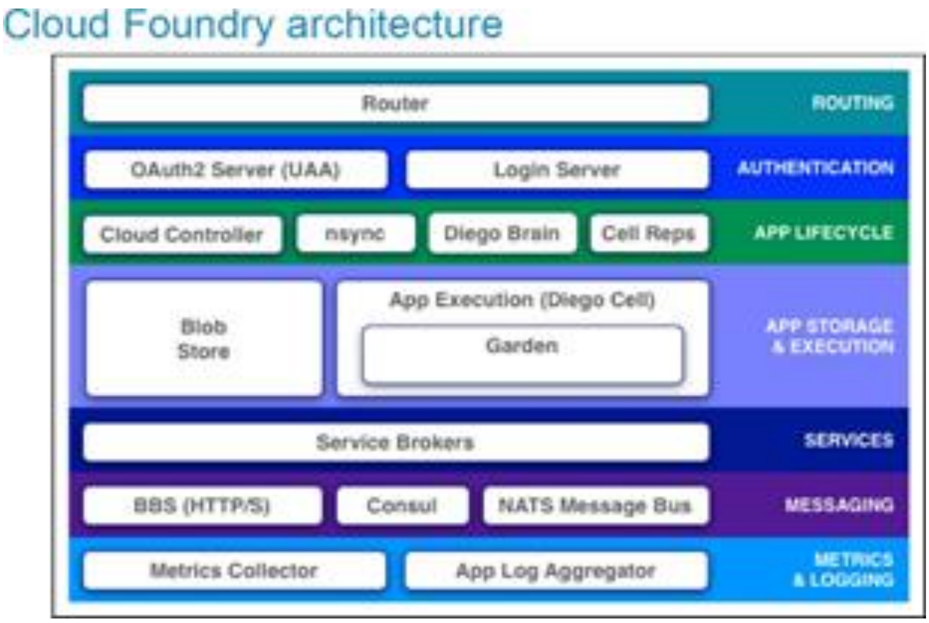
Cloud Foundry ensures that the build and deploy aspects of coding remain carefully coordinated with any attached services — resulting in quick, consistent and reliable iterating of applications.

Applications deployed to Cloud Foundry access external resources via an Open Service Broker API, which was launched in December 2016.

In a platform, all external dependencies such as databases, messaging systems, files systems and so on are considered services. Cloud Foundry allows administrators to create a marketplace of services, from which users can provision these services on-demand.

When an application is pushed to Cloud Foundry the services it needs may also be specified. This process puts the credentials in an environment variable.

The Cloud Foundry platform is available from either the Cloud Foundry Foundation as open- source software or from a variety of commercial providers as either a software product or delivered as a service. Cloud Foundry is open-source software, and hence, it is available to anyone. Deploying Cloud Foundry involves interfacing with the underlying infrastructure using the Cloud Foundry BOSH deployment system, another open-source tool governed by the Cloud Foundry Foundation.



Q.4. What is Kubernetes Service in IBM Cloud? Discuss its working mechanism.

Different software environments have different restrictions and complexities. For example, if your development team wrote the software with one operating system in mind, you might have trouble running it on a different operating system. This can also be true for the underlying hardware or cloud environment.

This creates conflict between different parts of your organization. The development team may write a piece of software that causes difficulties for operations, for example. It can also mean performance issues when you deploy your applications in new environments. Even small differences can cause large problems.

One solution to this problem is something called a container, but using containers also creates an issue of its own in the form of managing and deploying all of the containerized applications. Large enterprise systems could include a huge number of containers.

That's where Kubernetes comes in. Kubernetes is an open source solution for managing, automating and deploying containerized applications.

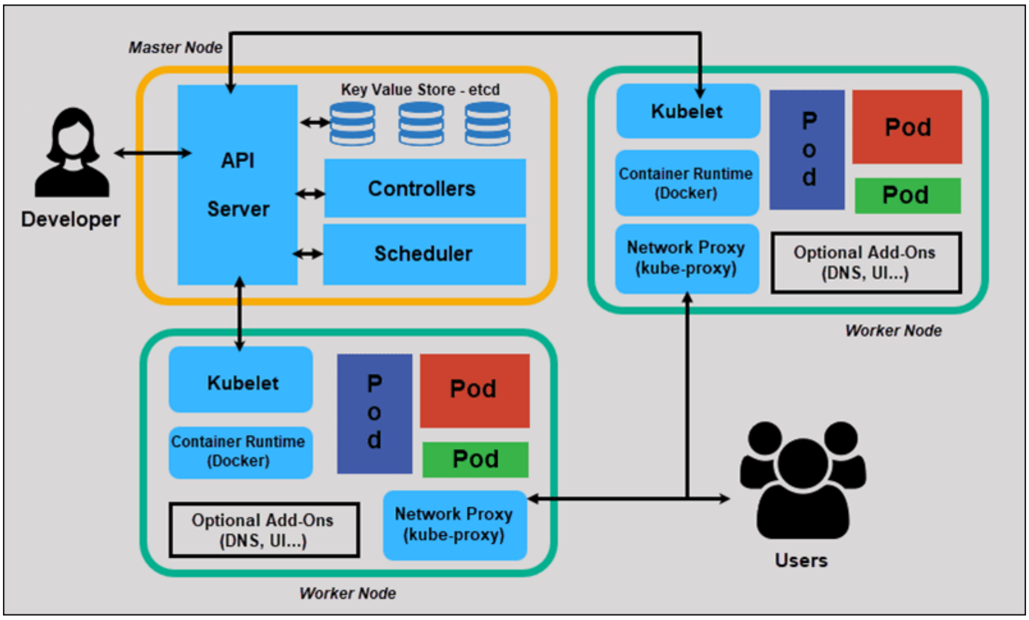
It has the following features :

Cost Savings : Kubernetes clusters are known for being low maintenance. Your team won't have to write its own container automation scripts. Team members also won't have to choose between reinventing the wheel or passing up the advantages of a shared infrastructure. You might also be able to reduce hardware costs by making more effective use of your current hardware.

Faster time to market : Kubernetes is perfect for DevOps. Good container management means that so long as the software runs, the deployment will almost always be painless. This means quicker deployment.

IT Flexibility : In the modern enterprise, software runs on any number of private and shared infrastructures. Having a container management solution means you won't have to sacrifice performance or make major adjustments to move your applications. You can run software wherever your business needs it. That makes your team nimbler.

Kubernetes architecture diagram:


Master Node
Key Value Store - etcd
API Server
Controllers
Scheduler
Developer
Kubelet
Container Runtime (Docker)
Network Proxy (kube-proxy)
Pod
Pod
Pod
Optional Add-Ons (DNS, UI...)
Worker Node
Kubelet
Container Runtime (Docker)
Pod
Pod
Pod
Optional Add-Ons (DNS, UI...)
Network Proxy (kube-proxy)
Worker Node
Users

Kubernetes follows the master-slave architecture. The components of Kubernetes can be divided into those that manage an individual node and those that are part of the control.

etcd is a persistent, lightweight, distributed, key-value data store developed by CoreOS that reliably stores the configuration data of the cluster, representing the overall state of the cluster at any given point of time. Other components watch for changes to this store to bring themselves into the desired state.

The **API server** is a key component and serves the Kubernetes API using JSON over HTTP, which provides both the internal and external interface to Kubernetes. The API server processes and validates REST requests and updates the state of the API objects in etcd, thereby allowing clients to configure workloads and containers across Worker nodes.

The **scheduler** is the pluggable component that selects which node an unscheduled pod (the basic entity managed by the scheduler) runs on, based on resource availability. Scheduler tracks resource use on each node to ensure that workload is not scheduled in excess of available resources. For this purpose, the scheduler must know the resource requirements, resource availability, and other user-provided constraints and policy directives such as quality-of-service, affinity/anti-affinity requirements, data locality, and so on. In essence, the scheduler's role is to match resource "supply" to workload "demand".

The **controller manager** is a process that runs core Kubernetes controllers like DaemonSet Controller and Replication Controller. The controllers communicate with the API server to create, update, and delete the resources they manage (pods, service endpoints, etc.)

The **Node**, also known as **Worker** or **Minion**, is a machine where containers (workloads) are deployed. Every node in the cluster must run a container runtime such as Docker, as well as the below-mentioned components, for communication with master for network configuration of these containers.

Kubelet is responsible for the running state of each node, ensuring that all containers on the node are healthy. It takes care of starting, stopping, and maintaining application containers organized into pods as directed by the control plane. Kubelet monitors the state of a pod, and if not in the desired state, the pod re-deploys to the same node. Node status is relayed every few seconds via heartbeat messages to the master. Once the master detects a node failure, the Replication Controller observes this state change and launches pods on other healthy nodes.

A **container** resides inside a pod. The container is the lowest level of a micro-service that holds the running application, libraries, and their dependencies. Containers can be exposed to the world through an external IP address. Kubernetes supports Docker containers since its first version, in July 2016 rtk container engine was added.

The **Kube-proxy** is an implementation of a network proxy and a load balancer, and it supports the service abstraction along with other networking operations. It is responsible for routing traffic to the appropriate container based on IP and port number of the incoming request.

cAdvisor is an agent that monitors and gathers resource usage and performance metrics such as CPU, memory, file and network usage of containers on each node.

Working :

1. A developer first generates a basic starting application.
2. Building the application produces a Docker container image.
3. The image is pushed to the Container Registry.
4. The application is then deployed to a Kubernetes cluster.
5. Users access the application.

Q.5. What are the pricing schemes for computing resources in IBM Cloud? Make a list of pricing schemes and computing facilities.

IBM Cloud application and developer services :

Lite (Free)	No time limit No credit card required 256 MB of Cloud Foundry memory
Pay as you go	Only pay for what you use (you can estimate your cost with the calculator)
Subscription	Fixed monthly billing. Customized discounts. Public, dedicated/local environments. (you need to contact them)

Infrastructure services :

Bare metal	1 - Intel Xeon E3-1270 v6 (₽14,541.97/mo) 4 Cores, 3.80 GHz 16 GB RAM CentOS 7.6 (64 bit) 1 - Disk controller - Non-RAID (₽0.00) Individual ₽965.18 1.00 TB SATA x 1 Network interface (₽0.00) 100 Mbps Public & Private Network Uplinks Total due per month* ₽15,507.15 (Changes according to your need)
Virtual servers	Location - CHE01 , Chennai System Configuration - B1.1x2x25 , 1vCPU, 2 GB RAM, 25 GB (SAN) ($24.50) Bandwidth - 250 GB Bandwidth Allotment [$0.00] , Many other configurations
Cloud storage	Lite - Free Standard - There is no minimum fee, so you pay only for what you use.

Q6. Discuss about features and working mechanism of IBM Watson for natural language processing, visual recognition and machine learning.

Watson was created as a question answering (QA) computing system that IBM built to apply advanced natural language processing, information retrieval, knowledge representation, automated reasoning, and machine learning technologies to the field of open domain question answering.

The key difference between QA technology and document search is that document search takes a keyword query and returns a list of documents, ranked in order of relevance to the query (often based on popularity and page ranking), while QA technology takes a question expressed in natural language, seeks to understand it in much greater detail, and returns a precise answer to the question.

Watson uses IBM's DeepQA software and the Apache UIMA (Unstructured Information Management Architecture) framework implementation. The system was written in various languages, including Java, C++, and Prolog, and runs on the SUSE Linux Enterprise Server 11 operating system using the Apache Hadoop framework to provide distributed computing.

Features of Natural Language Understanding in IBM Watson :

Powerful Insights : Analyses semantic features of text input, including categories, concepts, emotion, entities, keywords, metadata, relations, semantic roles, and sentiment.

Domain Customization : Extend Natural Language Understanding with custom models built on Watson Knowledge Studio that can identify custom entities and relations unique to your domain.

Broad Language Support : Support a variety of languages depending on which features are being analysed, including English, Arabic, Chinese (simplified), Dutch, French, German, Italian, Japanese, Korean, Portuguese, Russian, Spanish, and Swedish - with more to come.

Numerous Applications : Apply to various use cases, including content recommendation, advertising optimization, audience segmentation, voice-of- customer analysis, data mining, and more.

Features of Visual Recognition in IBM Watson:

General Model : Generate class keywords that describe the image. Use your own images, or extract relevant image URLs from publicly accessible web pages for analysis.

Custom Model : Create custom, unique visual classifiers. Use the service to recognize custom visual concepts that are not available with general models.

Face Model : Detect human faces in the image. This service also provides a general indication of age range and gender of faces.

Food Model : Utilize a specialized vocabulary of over 2,000 foods to identify meals, food items, and dishes with enhanced accuracy.

Explicit Model : Assess whether an image contains objectionable or adult content that may be unsuitable for general audiences.

Pricing Plans (Free) of Natural Language Processing and Visual Recognition in IBM Watson :

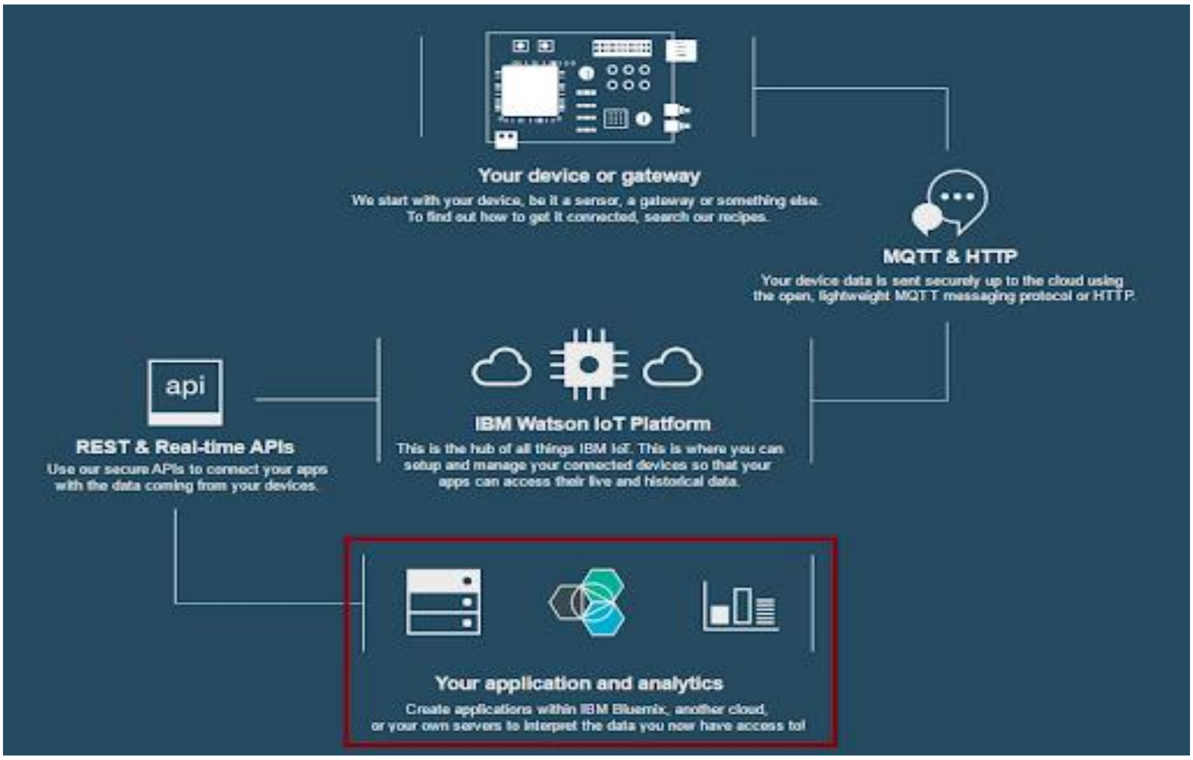
	Plan	Features
Natural Language Processing	Lite	30,000 NLU items per month ❑ 1 custom item
Visual Recognition	Lite	1,000 Events per month towards : ❑ Pre-trained model classification (General, Face, Food, Explicit) (images) ❑ Custom Model classification (images) ❑ Custom Model training (images) ❑ 2 Custom Models ❑ 1 Lite Plan instance per IBM Cloud Organization ❑ Free Exports to Core ML

Features of Machine Learning in IBM Watson :

The IBM Watson Machine Learning service is a set of REST APIs that can be called from any programming language.

The focus of the IBM Watson™ Machine Learning service is deployment, but you can use IBM® SPSS® Modeler or IBM Watson™ Studio to author and work with models and pipelines. Both SPSS® Modeler and IBM Watson™ Studio that use Spark MLlib and Python scikit-learn offer various modeling methods that are taken from machine learning, artificial intelligence, and statistics.

Although training is a critical step in the machine learning process, IBM Watson™ Machine Learning enables you to streamline the functioning of your models by deploying them and getting actual business value from them over time and through all of their iterations.



Your device or gateway

We start with your device, be it a sensor, a gateway or something else. To find out how to get it connected, search our recipes.

MQTT & HTTP

Your device data is sent securely up to the cloud using the open, lightweight MQTT messaging protocol or HTTP.

REST & Real-time APIs

Use our secure APIs to connect your apps with the data coming from your devices.

IBM Watson IoT Platform

This is the hub of all things IBM IoT. This is where you can setup and manage your connected devices so that your apps can access their live and historical data.

Your application and analytics

Create applications within IBM Bluemix, another cloud, or your own servers to interpret the data you now have access to!

Q.7. Prepare a comparative chart among EC2, Microsoft Azure, Google cloud and IBM cloud based on at least 10 common features.

The comparison of AWS, Azure, Google Cloud Platform, and IBM Cloud can be done on the basis of features, pricing, and solutions.

Features	EC2	Google Cloud	IBM Cloud	Azure

Caching	ElastiCache	Memcache	Redis	Azure Redis Cache
Containers	EC2 Container Service (ECS)	Kubernetes container	IBM Cloud Container Service	Azure Container Service (AKS)
Content delivery network	CloudFront	Google Cloud CDN	Content Delivery Network	Azure Content Delivery Network
Data Access	Amazon Athena	Big Query	SQL Query	Data Catalog
Virtual Personal Assistant	Alexa	Google Assistant	IBM Watson	Cortana
Email Service	Simple Email Service	G-Suite mailing	SendGrid	Azure SMTP Server
Messaging Service	Simple Queue Service (SQS)	Google Cloud Pub/Sub	RabbitMQ	Azure Queue Storage
Content management in cloud	Solodev DCX	Google Docs	IBM FileNet	SharePoint Online
Programmatic Interface	Command Line Interface	Cloud Shell	IBM Cloud CLI	Azure Command Line Interface (CLI)

Security mechanism	Inspector	Independent Security Evaluators (ISE)	Nessus Security Scanner	Security Center