G H Raisoni College of Engineering & Management
An Empowered Autonomous Institute, Affiliated to Savitribai Phule Pune University
Approved by AICTE, New Delhi and Recognized by Govt. of Maharashtra, NAAC Accredited "A+" Grade
Navin Gat No.1200, Domkhel Road, Wagholi, Pune – 412 207 (India)
T: +91 - 9604787185/186 | E: ghrcem.pune@raisoni.net | W: www.ghrcemp.raisoni.net

---

Department of Data Science

A.Y.-2024-25
B-Tech Third Year (Sem-V)

Subject: Cloud Computing (UDSP301)
Lab Manual
Semester-V

Faculty Name:

Dr. Arti Patle
Prof. Sunita Wani

Department of Data Science

Prof. Sharayu Anap
Experiment No. 1

Title: Explore AWS Cloud Based IaaS Service

Aim: To explore AWS cloud based Iaas Service

Theory:

IaaS or Infrastructure as a service is one of Amazon Web Services that focuses on providing infrastructure services based on cloud computing technology. It's not difficult to name an organization that provides IaaS but among the top providers worldwide, AWS stands out as the front runner for IaaS cloud services. It has clients in 190 countries and 66 available Zones within 21 geographic regions. IaaS Amazon Service is used to replace physical resources, such as servers, with virtual resources hosted and managed by Amazon. System users can run any operating system or application on these leased servers, without incurring any extra fees for maintenance and operation.

AWS IaaS Benefits:

- Availability of separate development environment
- Hardware and operating system specifications for the service can be selected and used directly from the network
- Allow expanding the resources of the server in terms of quantity and functionality
- No errors or extra costs arise while upgrading the system

AWS IaaS Characteristics Characteristics that define IaaS include:

- Resources are available as a service
- Cost depends on the consumption
- Highly scalable
- Allow multiple users to access a single piece of hardware
- The organization has complete control of the infrastructure
- Dynamic and flexible

When to use Amazon AWS IaaS

AWS IaaS is suitable for:

- Startups and small companies which want to avoid spending time and money in producing hardware and software.
- Larger companies which prefer complete control over their applications and infrastructure, but want to purchase only what is needed for the app development.
- Companies with rapid growth potential that like to keep altering hardware and software as per their demands for scaling up.

AWS IaaS services Drawbacks:

- Security threats which may arise from the host or other virtual machines
- Customers' inability to access their data when vendor outages happen
- Required team training to learn how to manage new infrastructure

Architecture/Block Diagram/Design Diagram: (Print)
Procedure (Step wise snapshots print)

Questions:

1. What Is AWS EC2 and Why It Is Important ?
2. What is EC2 in AWS in simple words?
3. How Create A Instance In EC2 ?

Conclusion:

AWS's IaaS offerings provide a comprehensive suite of services that cater to various computing needs, from hosting simple websites to running complex applications. The flexibility, scalability, and cost-effectiveness of AWS IaaS make it an attractive option for businesses looking to leverage cloud technology.

Experiment No. 2

Title: Implement Virtualization using VirtualBox/ VMware Workstation

Aim: To implement Virtualization using VirtualBox/ VMware Workstation

Theory:
Virtualization is a technique of how to separate a service from the underlying physical delivery of that service. It is the process of creating a virtual version of something like computer hardware. It was initially developed during the mainframe era. It involves using specialized software to create a virtual or software-created version of a computing resource rather than the actual version of the same resource. With the help of Virtualization, multiple operating systems and applications can run on same machine and its same hardware at the same time, increasing the utilization and flexibility of hardware.

In other words, one of the main cost effective, hardware reducing, and energy saving techniques used by cloud providers is virtualization. Virtualization allows sharing a single physical instance of a resource or an application among multiple customers and organizations at one time. It does this by assigning a logical name to a physical storage and providing a pointer to that physical resource on demand. The term virtualization is often synonymous with hardware virtualization, which plays a fundamental role in efficiently delivering Infrastructure-as-a-Service (IaaS) solutions for cloud computing. Moreover, virtualization technologies provide a virtual environment for not only executing applications but also for storage, memory, and networking.

BENEFITS OF VIRTUALIZATION

1. More flexible and efficient allocation of resources.
2. Enhance development productivity.
3. It lowers the cost of IT infrastructure.
4. Remote access and rapid scalability.
5. High availability and disaster recovery.
6. Pay peruse of the IT infrastructure on demand.
7. Enables running multiple operating systems.

Types of Virtualization:

1.Application Virtualization.

2.Network Virtualization.
3.Desktop Virtualization.
4.Storage Virtualization.
5.Server Virtualization.
6.Data virtualization.

1. Application Virtualization:

Application virtualization helps a user to have remote access of an application from a server. The server stores all personal information and other characteristics of the application but can still run on a local workstation through the internet. Example of this would be a user who needs to run two different versions of the same software. Technologies that use application virtualization are hosted applications and packaged applications.

2. Network Virtualization:

The ability to run multiple virtual networks with each has a separate control and data plan. It co-exists together on top of one physical network. It can be managed by individual parties that potentially confidential to each other. Network virtualization provides a facility to create and provision virtual networks—logical switches routers, firewalls, load balancer, Virtual Private Network (VPN), and workload security within days or even in weeks.

3. Desktop Virtualization:

Desktop virtualization allows the users' OS to be remotely stored on a server in the data centre. It allows the user to access their desktop virtually, from any location by a different machine. Users who want specific operating systems other than Windows Server will need to have a virtual desktop. Main benefits of desktop virtualization are user mobility, portability, easy management of software installation, updates, and patches.

4. Storage Virtualization:

Storage virtualization is an array of servers that are managed by a virtual storage system. The servers aren't aware of exactly where their data is stored, and instead function more like worker bees in a hive. It makes managing storage from multiple sources to be managed and utilized as a single repository. storage virtualization software maintains smooth operations, consistent performance and a continuous suite of advanced functions despite changes, break down and differences in the underlying equipment.

5. Server Virtualization:

This is a kind of virtualization in which masking of server resources takes place. Here, the central-server(physical server) is divided into multiple different virtual servers by changing the identity number, processors. So, each system can operate its own operating systems in isolate

Manner. Where each sub-server knows the identity of the central server. It causes an increase in the performance and reduces the operating cost by the deployment of main server resources into a sub-server resource. It's beneficial in virtual migration, reduce energy consumption, reduce infrastructural cost, etc.

6. Data virtualization:
This is the kind of virtualization in which the data is collected from various sources and managed that at a single place without knowing more about the technical information like how data is collected, stored & formatted then arranged that data logically so that its virtual view can be accessed by its interested people and stakeholders, and users through the various cloud services remotely. Many big giant companies are providing their services like Oracle, IBM, At scale, Cdata, etc.

Conclusion: The implementation of virtualization using VM presents a pivotal solution for contemporary computing infrastructures. By leveraging the power of VM, organizations can efficiently create, manage, and deploy virtualized environments, thereby maximizing hardware utilization and minimizing operational overheads. Through seamless integration with existing systems and robust support for diverse workloads, virtualization fosters scalability, reliability, and agility within IT environments.

Questions:
1. What is VirtualBox/ VMware Workstation and how does it differ from other virtualization technologies?
2. How do you install and configure VirtualBox/ VMware Workstation?
3. What are the system requirements for installing VirtualBox/ VMware Workstation?

Experiment No. 3

Title: Creating a Warehouse Application in SalesForce.com PaaS
Aim: To create a Warehouse Application in SalesForce.com PaaS

Theory:
Paas(Platform as a service):
Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications. You purchase the resources you need from a cloud service provider on a pay-as-you-go basis and access them over a secure Internet connection.

The main offerings included by PaaS vendors are:

- Development tools

- Middleware

- Operating systems

- Database management

- Infrastructure

Different vendors may include other services as well, but these are the core PaaS services.
Development tools:

PaaS vendors offer a variety of tools that are necessary for software development, including a source code editor, a debugger, a compiler, and other essential tools. These tools may be offered together as a framework. The specific tools offered will depend on the vendor, but PaaS offerings should include everything a developer needs to build their application.

Middleware:

Platforms offered as a service usually include middleware, so that developers don't have to build it themselves. Middleware is software that sits in between user-facing applications and the machine's operating system; for example, middleware is what allows software to access input from the keyboard and mouse. Middleware is necessary for running an application, but end users don't interact with it.

Operating systems:

A PaaS vendor will provide and maintain the operating system that developers work on and the application runs on.

Databases:

PaaS providers administer and maintain databases. They will usually provide developers with a database management system as well.

Infrastructure:

PaaS is the next layer up from IaaS in the cloud computing service model, and everything included in IaaS is also included in PaaS. A PaaS provider either manages servers, storage, and physical data centers, or purchases them from an IaaS provider.

Advantages of Paas:

Faster time to market PaaS is used to build applications more quickly than would be possible if developers had to worry about building, configuring, and provisioning their own platforms and backend infrastructure. With PaaS, all they need to do is write the code and test the application, and the vendor handles the rest.

One environment from start to finish: PaaS permits developers to build, test, debug, deploy, host, and update their applications all in the same environment. This enables developers to be sure a web application will function properly as hosted before they release, and it simplifies the application development lifecycle.

Price: PaaS is more cost-effective than leveraging IaaS in many cases. Overhead is reduced because PaaS customers don't need to manage and provision virtual machines. In addition, some providers have a pay-as-you-go pricing structure, in which the vendor only charges for the computing resources used by the application, usually saving customers money. However, each vendor has a slightly different pricing structure, and some platform providers charge a flat fee per month.

Ease of licensing: PaaS providers handle all licensing for operating systems, development tools, and everything else included in their platform.

Potential drawbacks of using PaaS:

Vendor lock-in: It may become hard to switch PaaS providers, since the application is built using the vendor's tools and specifically for their platform. Each vendor may have different architecture requirements. Different vendors may not support the same languages, libraries, APIs, architecture, or operating system used to build and run the application. To switch vendors, developers may need to either rebuild or heavily alter their application.

Vendor dependency:

The effort and resources involved in changing PaaS vendors may make companies more dependent on their current vendor. A small change in the vendor's internal processes or infrastructure could have a huge impact on the performance of an application designed to run efficiently on the old configuration. Additionally, if the vendor changes their pricing model, an application may suddenly become more expensive to operate.

Security and compliance challenges:

In a PaaS architecture, the external vendor will store most or all of an application's data, along with hosting its code. In some cases the vendor may actually store the databases via a further third party, an IaaS provider. Though most PaaS vendors are large companies with strong security in place, this makes it difficult to fully assess and test the security measures protecting the application and its data. In addition, for companies that have to comply with strict data security regulations, verifying the compliance of additional external vendors will add more hurdles to going to market.

What is salesforce?

Salesforce is a company that makes cloud-based software designed to help businesses find more prospects, close more deals, and wow customers with amazing service. It also helps teams work better together. We can use a single Customer 360 app, or a combination of many. By improving team communications and productivity, business drive greater success.

Conclusion:

Salesforce provides a comprehensive platform for developing cloud computing applications tailored to various business needs. By leveraging its powerful tools, integration capabilities, and robust ecosystem, you can create scalable and effective applications that enhance productivity and drive growth. Whether you are a small business or a large enterprise, Salesforce can help you harness the power of cloud computing to achieve your goals.

Department of Data Science

Questions:

1. What is SalesForce Cloud?
2. Compare salesforce cloud with AWS cloud.
3. Write installation steps of creating application in salesforce cloud.

Experiment No. 4

Title: Explore Cloud Services using CloudSim Simulator

Aim: To explore Cloud Services using CloudSim Simulator

Theory:

CloudSim is an open-source framework, which is used to simulate cloud computing infrastructure and services. It is developed by the CLOUDS Lab organization and is written entirely in Java. It is used for modeling and simulating a cloud computing environment as a means for evaluating a hypothesis prior to software development in order to reproduce tests and results.

For example, if you were to deploy an application or a website on the cloud and wanted to test the services and load that your product can handle and also tune its performance to overcome bottlenecks before risking deployment, then such evaluations could be performed by simply coding a simulation of that environment with the help of various flexible and scalable classes provided by the CloudSim package, free of cost.

Benefits of Simulation over the Actual Deployment:

Following are the benefits of CloudSim:

- No capital investment involved. With a simulation tool like CloudSim there is no installation or maintenance cost.
- Easy to use and Scalable. You can change the requirements such as adding or deleting resources by changing just a few lines of code.
- Risks can be evaluated at an earlier stage. In Cloud Computing utilization of real testbeds limits the experiments to the scale of the testbed and makes the reproduction of results an extremely difficult undertaking. With simulation, you can test your product against test cases and resolve issues before actual deployment without any limitations.
- No need for try-and-error approaches. Instead of relying on theoretical and imprecise evaluations which can lead to inefficient service performance and revenue generation, you can test your services in a repeatable and controlled environment free of cost with CloudSim.

Why use CloudSim?

Below are a few reasons to opt for CloudSim:

- Open source and free of cost, so it favors researchers/developers working in the field.
- Easy to download and set-up.
- It is more generalized and extensible to support modeling and experimentation.
- Does not require any high-specs computer to work on.
- Provides pre-defined allocation policies and utilization models for managing resources, and allows implementation of user-defined algorithms as well.

- The documentation provides pre-coded examples for new developers to get familiar with the basic classes and functions.
- Tackle bottlenecks before deployment to reduce risk, lower costs, increase performance, and raise revenue.

CloudSim Layered Architecture

CloudSim Core Simulation Engine provides interfaces for the management of resources such as VM, memory and bandwidth of virtualized Datacenters. CloudSim layer manages the creation and execution of core entities such as VMs, Cloudlets, Hosts etc. It also handles network-related execution along with the provisioning of resources and their execution and management.
User Code is the layer controlled by the user. The developer can write the requirements of the hardware specifications in this layer according to the scenario.
Some of the most common classes used during simulation are:

- Datacenter: used for modelling the foundational hardware equipment of any cloud environment, that is the Datacenter. This class provides methods to specify the functional requirements of the Datacenter as well as methods to set the allocation policies of the VMs etc.
- Host: this class executes actions related to management of virtual machines. It also defines policies for provisioning memory and bandwidth to the virtual machines, as well as allocating CPU cores to the virtual machines.
- VM: this class represents a virtual machine by providing data members defining a VM's bandwidth, RAM, mips (million instructions per second), size while also providing setter and getter methods for these parameters.
- Cloudlet: a cloudlet class represents any task that is run on a VM, like a processing task, or a memory access task, or a file updating task etc. It stores parameters defining the characteristics of a task such as its length, size, mi (million instructions) and provides methods similarly to VM class while also providing methods that define a task's execution time, status, cost and history.
- DatacenterBroker: is an entity acting on behalf of the user/customer. It is responsible for functioning of VMs, including VM creation, management, destruction and submission of cloudlets to the VM.

• CloudSim: this is the class responsible for initializing and starting the simulation environment after all the necessary cloud entities have been defined and later stopping after all the entities have been destroyed.

Features of CloudSim:

CloudSim provides support for simulation and modelling of:
1. Large scale virtualized Datacenters, servers and hosts.
2. Customizable policies for provisioning host to virtual machines.
3. Energy-aware computational resources.
4. Application containers and federated clouds (joining and management of multiple public clouds).
5. Datacenter network topologies and message-passing applications.
6. Dynamic insertion of simulation entities with stop and resume of simulation.
7. User-defined allocation and provisioning policies.

Conclusion: The cloudsim technology is supportive for the generation of several cloud applications with the creation of virtual machines and data centers. It is the easiest way for the analyzing process. There are many cloud simulators which is useful for the cloud research.

Questions:
1. What is cloudsim technology?
2. Why use CloudSim?
3. Write installation steps for CloudSim?

Experiment No. 5

Title: Explore Cloud Service using Google Cloud

Aim: To explore Cloud Service using Google Cloud

Theory:

1. Database Services

Teams can build and deploy faster, deliver transformative applications, and maintain portability and control of data with Google Cloud database.

The 5 Database Types are,

1. Relational
2. Key value
3. Document
4. In-memory
5. Additional NoSQL

Several SQL and NoSQL database services are provided by Google Cloud. A SQL database inCloud SQL, which provides either MySQL or PostgreSQL databases. A fully managed, mission-critical, relational database service in Cloud Spanner that offers transactional consistency at global scale, schemas, SQL querying, and automatic, synchronous replication for high availability.

Google Cloud provides 2 options for NoSQL data storage. They are,

1. Cloud Bigtable used for tabular data
2. Fire store used for document-like data

By using persistent disks, on Compute Engine, one can easily set up their preferred database technology. A MongoDB can be set up for NoSQL document storage.

Google Cloud Partner Services are MongoDB, Datastax, Redis Labs, and Neo4j.

2. Big Data Services

Big data services processes and query big data in the cloud to get fast answers to complicated questions. It is always time consuming and expensive to store and query huge datasets. And to do so without the right hardware and infrastructure support is a complete nightmare. Google's BigQuery is a data analysis tool/service that helps users to create schemas to organize data into tables and datasets. The key factor about BigQuery is that it is fully managed, Hence the need to deploy resources is not required.

Enterprises (big data for enterprises) prefer BigQuery for storing and querying huge datasets. BigQuery with the power of Google's infrastructure enables super-fast SQL queries. With BigQuery, enterprises can control access to both the data and project based on your business needs. BigQuery also accommodates multiple 3rd party tools such as visualizing the data or loading the data.

Advantages of BigQuery are,
1. Loads data (Includes Streaming data) from multiple sources
2. SQL-like commands are used to query massive datasets very quickly.
3. Designed and optimized for speed.
4. Use the web UI, command-line interface, or API.
5. Manage data and protect it by using permissions.
3. Networking Services

Kubernetes model is used by GKE whereas the app engine manages the networking. A set of networking services is provided by compute engine. These services help you to load-balance traffic across resources, create DNS records, and connect your existing network to Google's network, to balance the load and to create DNS records — these services are used.
We shall now investigate the main aspects of Google Kubernetes Engine networking.
Application deploying, communication between applications, app communication with Kubernetes control panel — all can be defined through Kubernetes.
It is highly important and effective to change the way we think about the application's network design and their hosts while Kubernetes run the application. Instead of focusing on the host's or VMs connections, with Kubernetes in place, think about the communication between external clients, pods, and services.
Kubernetes along with Google Cloud configures IP filtering rules dynamically, routing tables, and firewall rules on each node. All these are completely dependent on the way the
Kubernetes deployments are done and the way cluster configuration is done on Google Cloud.
4. Machine Learning Services
A variety of APIs is provided by Google Cloud helping you to enable and take advantage of Google's ML without creating and training your own models.

A few of the APIs are as follows:
1. Video Intelligence API for video analysis technology
2. Speech-to-Text API to convert audio to text
3. Cloud Vision API to integrate vision detection features
4. Cloud Natural Language API to add sentiment analysis
5. Cloud Translation API to translate the source text
6. Dialogflow API to build conversational interfaces for websites, mobile applications,

popular messaging platforms, and IoT devices.
5. Storage Services

Google Cloud provides a variety of storage services such as,
1. Consistent
2. Scalable
3. Large-capacity data storage in Cloud Storage.
Cloud Storage comes in several flavours such as,
1. Standard Cloud Storage for maximum availability
2. Cloud Storage Nearline for low-cost archival storage
3. Cloud Storage Coldline for even lower-cost archival storage
4. Cloud Storage Archive for the lowest-cost archival storage
5.Filestore instances can be used to store data from applications running on Compute Engine VM instances or GKE clusters.
6. Computing & Hosting Services

Computing and hosting options from Google Cloud are as follows:
1. Serverless environment
2. Managed application platform.
3. Leverage container technologies
4. Build own cloud-based infrastructure
7.Serverless computing : Google Cloud's functions provides a serverless environment for building and connecting cloud services. A Simple, single-purpose functions can be written using Cloud Functions that are attached to events emitted from the cloud infrastructure and services.
8.Application platform: App Engine is Google Cloud's platform as a service (PaaS). Google handles the management of the resources with App engine.
9.Containers: With container-based computing, the focus can be only on the application code. Google Kubernetes Engine (GKE), Google Cloud's containers as a service (CaaS) offering, is built on the open source Kubernetes system providing the flexibility of on-premises or hybrid clouds, in addition to Google Cloud's public cloud infrastructure.

Conclusion: The prominence of Google App Engine in Cloud Computing cannot be understated. It serves as a beacon of advanced design and user-friendliness, catering to both Developers and enterprises seamlessly. It stands out as the go-to choice for those seeking a dependable and secure Cloud infrastructure. When one aligns with Google App Engine, they're not only embracing the pinnacle of Cloud technology but also ensuring their applications consistently deliver top-tier performance.

Questions:

1. What is Cloud Computing ? What are the types of Cloud Computing.
2. Explain Google App Engine.
3. Write installation steps of Google App Engine.
4. What Are The Benefits of use of Google App Engine

Department of Data Science

Experiment No. 6

Title: Explore Cloud Service and table formation using Microsoft Azure Cloud

Aim: Explore Cloud Service and table formation using Microsoft Azure Cloud

Theory:

What is Azure?

Azure is Microsoft's cloud platform, just like Google has it's Google Cloud and Amazon has it's Amazon Web Service or AWS.000. Generally, it is a platform through which we can use Microsoft's resource. For example, to set up a huge server, we will require huge investment, effort, physical space and so on. In such situations, Microsoft Azure comes to our rescue. It will provide us with virtual machines, fast processing of data, analytical and monitoring tools and so on to make our work simpler. The pricing of Azure is also simpler and cost-effective. Popularly termed as "Pay As You Go", which means how much you use, pay only for that.

Azure History

Microsoft unveiled Windows Azure in early October 2008 but it went to live after February 2010. Later in 2014, Microsoft changed its name from Windows Azure to Microsoft Azure. Azure provided a service platform for .NET services, SQL Services, and many Live Services. Many people were still very skeptical about "the cloud". As an industry, we were entering a brave new world with many possibilities. Microsoft Azure is getting bigger and better in coming days. More tools and more functionalities are getting added. It has two releases as of now. It's famous version Microsoft Azure v1 and later Microsoft Azure v2. Microsoft Azure v1 was more like JSON script driven then the new version v2, which has interactive UI for simplification and easy learning. Microsoft Azure v2 is still in the preview version.

How Azure can help in business?

Azure can help in our business in the following ways-

- Capital less: We don't have to worry about the capital as Azure cuts out the high cost of hardware. You simply pay as you go and enjoy a subscription-based model that's kind to your cash flow. Also, to set up an Azure account is very easy. You simply register in Azure Portal and select your required subscription and get going.

- Less Operational Cost: Azure has low operational cost because it runs on its own servers whose only job is to make the cloud functional and bug-free, it's usually a whole lot more reliable than your own, on-location server.
- Cost Effective: If we set up a server on our own, we need to hire a tech support team to monitor them and make sure things are working fine. Also, there might be a situation where the tech support team is taking too much time to solve the issue incurred in the server. So, in this regard is way too pocket-friendly.
- Easy Back Up and Recovery options: Azure keep backups of all your valuable data. In disaster situations, you can recover all your data in a single click without your business getting affected. Cloud-based backup and recovery solutions save time, avoid large up-front investment and roll up third-party expertise as part of the deal.
- Easy to implement: It is very easy to implement your business models in Azure. With a couple of on-click activities, you are good to go. Even there are several tutorials to make you learn and deploy faster.
- Better Security: Azure provides more security than local servers. Be carefree about your critical data and business applications. As it stays safe in the Azure Cloud. Even, in natural disasters, where the resources can be harmed, Azure is a rescue. The cloud is always on.
- Work from anywhere: Azure gives you the freedom to work from anywhere and everywhere. It just requires a network connection and credentials. And with most serious Azure cloud services offering mobile apps, you're not restricted to which device you've got to hand.
- Increased collaboration: With Azure, teams can access, edit and share documents anytime, from anywhere. They can work and achieve future goals hand in hand.

Another advantage of the Azure is that it preserves records of activity and data. Timestamps are one example of the Azure's record keeping. Timestamps improve team collaboration by establishing transparency and increasing accountability.

Microsoft Azure Services

Some following are the services of Microsoft Azure offers:

1. Compute: Includes Virtual Machines, Virtual Machine Scale Sets, Functions for serverless computing, Batch for containerized batch workloads, Service Fabric for microservices and container orchestration, and Cloud Services for building cloud-based apps and APIs.
2. Networking: With Azure you can use variety of networking tools, like the Virtual Network, which can connect to on-premise data centers; Load Balancer; Application Gateway; VPN Gateway; Azure DNS for domain hosting, Content Delivery Network, Traffic Manager, ExpressRoute dedicated private network fiber connections; and Network Watcher monitoring and diagnostics
3. Storage: Includes Blob, Queue, File and Disk Storage, as well as a Data Lake Store, Backup and Site Recovery, among others.
4. Web + Mobile: Creating Web + Mobile applications is very easy as it includes several services for building and deploying applications.
5. Containers: Azure has a property which includes Container Service, which supports

Kubernetes, DC/OS or Docker Swarm, and Container Registry, as well as tools for microservices.
6. Databases: Azure has also includes several SQL-based databases and related tools.
7. Data + Analytics: Azure has some big data tools like HDInsight for Hadoop Spark, R Server, HBase and Storm clusters
8. AI + Cognitive Services: With Azure developing applications with artificial intelligence capabilities, like the Computer Vision API, Face API, Bing Web Search, Video Indexer, Language Understanding Intelligent.
9. Internet of Things: Includes IoT Hub and IoT Edge services that can be combined with a variety of machine learning, analytics, and communications services.

10. Security + Identity: Includes Security Center, Azure Active Directory, Key Vault and Multi-Factor Authentication Services.

11. Developer Tools: Includes cloud development services like Visual Studio Team Services, Azure DevTest Labs, HockeyApp mobile app deployment and monitoring, Xamarin cross-platform mobile development and more.

Conclusion:
In conclusion, cloud computing is recently new technological development that has the potential to have a great impact on the world. It has many benefits that it provides to it users and businesses.

Questions:
1. What is Azure?
2. How Azure can help in business?

G H Raisoni College of Engineering & Management
An Empowered Autonomous Institute, Affiliated to Savitribai Phule Pune University
Approved by AICTE, New Delhi and Recognized by Govt. of Maharashtra, NAAC Accredited "A+" Grade
Navin Gat No.1200, Domkhel Road, Wagholi, Pune – 412 207 (India)
T: +91 - 9604787185/186 | E: ghrcem.pune@raisoni.net | W: www.ghrcemp.raisoni.net

Department of Data Science

Experiment No. 7

Title: Creating a Warehouse Application in SalesForce.com PaaS

Aim: To Create a Warehouse Application in SalesForce.com PaaS

Theory:

Paas(Platform as a service):
Platform as a service (PaaS) is a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications. You purchase the resources you need from a cloud service provider on a pay-as-you-go basis and access them over a secure Internet connection.

The main offerings included by PaaS vendors are:

- Development tools
- Middleware
- Operating systems
- Database management
- Infrastructure

Different vendors may include other services as well, but these are the core PaaS services.
Development tools:

PaaS vendors offer a variety of tools that are necessary for software development, including a source code editor, a debugger, a compiler, and other essential tools. These tools may be offered together as a framework. The specific tools offered will depend on the vendor, but PaaS offerings should include everything a developer needs to build their application.

Middleware:

Platforms offered as a service usually include middleware, so that developers don't have to build it themselves. Middleware is software that sits in between user-facing applications and the

machine's operating system; for example, middleware is what allows software to access input from the keyboard and mouse. Middleware is necessary for running an application, but end users don't interact with it.

Operating systems:

A PaaS vendor will provide and maintain the operating system that developers work on and the application runs on.

Databases:

PaaS providers administer and maintain databases. They will usually provide developers with a database management system as well.

Infrastructure:

PaaS is the next layer up from IaaS in the cloud computing service model, and everything included in IaaS is also included in PaaS. A PaaS provider either manages servers, storage, and physical data centers, or purchases them from an IaaS provider.

Advantages of Paas:

Faster time to market PaaS is used to build applications more quickly than would be possible if developers had to worry about building, configuring, and provisioning their own platforms and backend infrastructure. With PaaS, all they need to do is write the code and test the application, and the vendor handles the rest.

One environment from start to finish: PaaS permits developers to build, test, debug, deploy, host, and update their applications all in the same environment. This enables developers to be sure a web application will function properly as hosted before they release, and it simplifies the application development lifecycle.

Price: PaaS is more cost-effective than leveraging IaaS in many cases. Overhead is reduced because PaaS customers don't need to manage and provision virtual machines. In addition, some providers have a pay-as-you-go pricing structure, in which the vendor only charges for the computing resources used by the application, usually saving customers money. However, each vendor has a slightly different pricing structure, and some platform providers charge a flat fee per month.

Ease of licensing: PaaS providers handle all licensing for operating systems, development tools, and everything else included in their platform.

Potential drawbacks of using PaaS:

Vendor lock-in: It may become hard to switch PaaS providers, since the application is built using the vendor's tools and specifically for their platform. Each vendor may have different architecture requirements. Different vendors may not support the same languages, libraries, APIs, architecture, or operating system used to build and run the application. To switch vendors, developers may need to either rebuild or heavily alter their application.

Vendor dependency:

The effort and resources involved in changing PaaS vendors may make companies more dependent on their current vendor. A small change in the vendor's internal processes or infrastructure could have a huge impact on the performance of an application designed to run efficiently on the old configuration. Additionally, if the vendor changes their pricing model, an application may suddenly become more expensive to operate.

Security and compliance challenges:

In a PaaS architecture, the external vendor will store most or all of an application's data, along with hosting its code. In some cases the vendor may actually store the databases via a further third party, an IaaS provider. Though most PaaS vendors are large companies with strong security in place, this makes it difficult to fully assess and test the security measures protecting the application and its data. In addition, for companies that have to comply with strict data security regulations, verifying the compliance of additional external vendors will add more hurdles to going to market.

What is salesforce?

Salesforce is a company that makes cloud-based software designed to help businesses find more prospects, close more deals, and wow customers with amazing service. It also helps teams work better together. We can use a single Customer 360 app, or a combination of many. By improving team communications and productivity, business drive greater success.

Conclusion:

Building a warehouse management application on Salesforce PaaS enables organizations to optimize their inventory and order management processes. By leveraging Salesforce's powerful tools, custom objects, automation features, and integration capabilities, you can create a comprehensive solution tailored to your warehouse needs. This approach not only enhances efficiency but also provides real-time visibility into operations, facilitating better decision-making and improved customer satisfaction.

Questions:

1. What is salesforce?
2. What are the Security and compliance challenges in Salesforce

Experiment No. 8

Title: Implement container management with Kubernetes

Aim: To Implement container management with Kubernetes

Theory:

Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.
The name Kubernetes originates from Greek, meaning helmsman or pilot. K8s as an abbreviation results from counting the eight letters between the "K" and the "s". Google open-sourced the Kubernetes project in 2014. Kubernetes combines over 15 years of Google's experience running production workloads at scale with best-of-breed ideas and practices from the community.
Let's take a look at why Kubernetes is so useful by going back in time.

Traditional deployment era: Early on, organizations ran applications on physical servers.
There was no way to define resource boundaries for applications in a physical server, and this caused resource allocation issues. For example, if multiple applications run on a physical server, there can be instances where one application would take up most of the resources, and as a result, the other applications would underperform. A solution for this would be to run each application on a different physical server. But this did not scale as resources were underutilized, and it was expensive for organizations to maintain many physical servers. Virtualized deployment era: As a solution, virtualization was introduced. It allows you to run multiple Virtual Machines (VMs) on a single physical server's CPU. Virtualization allows applications to be isolated between VMs and provides a level of security as the information of one application cannot be freely accessed by another application.
Virtualization allows better utilization of resources in a physical server and allows better scalability because an application can be added or updated easily, reduces hardware costs, and much more. With virtualization you can present a set of physical resources as a cluster of disposable virtual machines.
Each VM is a full machine running all the components, including its own operating system, on top of the virtualized hardware.
Container deployment era: Containers are similar to VMs, but they have relaxed isolation properties to share the Operating System (OS) among the applications. Therefore, containers are

considered lightweight. Similar to a VM, a container has its own filesystem, share of CPU, memory, process space, and more. As they are decoupled from the underlying infrastructure, they are portable across clouds and OS distributions. Containers have become popular because they provide extra benefits, such as:

- Agile application creation and deployment: increased ease and efficiency of container image creation compared to VM image use.
- Continuous development, integration, and deployment: provides for reliable and frequent container image build and deployment with quick and efficient rollbacks (due to image immutability).
- Dev and Ops separation of concerns: create application container images at build/release time rather than deployment time, thereby decoupling applications from infrastructure.
- Observability: not only surfaces OS-level information and metrics, but also application health and other signals.
- Environmental consistency across development, testing, and production: Runs the same on a laptop as it does in the cloud.
- Cloud and OS distribution portability: Runs on Ubuntu, RHEL, CoreOS, on-premises, on major public clouds, and anywhere else.
- Application-centric management: Raises the level of abstraction from running an OS on virtual hardware to running an application on an OS using logical resources.
- Loosely coupled, distributed, elastic, liberated micro-services: applications are broken into smaller, independent pieces and can be deployed and managed dynamically – not a monolithic stack running on one big single-purpose machine.
- Resource isolation: predictable application performance.
- Resource utilization: high efficiency and density.

Kubernetes provides :

- Service discovery and load balancing Kubernetes can expose a container using the DNS name or using their own IP address. If traffic to a container is high, Kubernetes is able to load balance and distribute the network traffic so that the deployment is stable.
- Storage orchestration Kubernetes allows you to automatically mount a storage system of your choice, such as local storages, public cloud providers, and more.
- Automated rollouts and rollbacks You can describe the desired state for your deployed containers using Kubernetes, and it can change the actual state to the desired state at a controlled rate. For example, you can automate Kubernetes to create new containers for your deployment, remove existing containers and adopt all their resources to the new container.
- Automatic bin packing You provide Kubernetes with a cluster of nodes that it can use to run containerized tasks. You tell Kubernetes how much CPU and memory (RAM) each container needs. Kubernetes can fit containers onto your nodes to make the best use of your resources.

• Self-healing Kubernetes restarts containers that fail, replaces containers, kills containers that don't respond to your user-defined health check, and doesn't advertise them to clients until they are ready to serve.

• Secret and configuration management Kubernetes lets you store and manage sensitive information, such as passwords, OAuth tokens, and SSH keys. You can deploy and update secrets and application configuration without rebuilding your container images, and without exposing secrets in your stack configuration.

Conclusion: Implementing container management with Kubernetes provides a powerful way to deploy, scale, and manage applications in a cloud-native environment. By following these steps, you can successfully set up a Kubernetes cluster, containerize your application, and manage it effectively. The robust features of Kubernetes will enhance your application's reliability and scalability, making it easier to adapt to changing business needs.

Questions:

1. What is Kubernetes?
2. What does Kubernetes provides?

Experiment No. 9

Title: Implement DevOps using Cloud

Aim: To Implement DevOps using Cloud

Theory:

DevOps in the cloud combines development and operations practices to enhance collaboration, automation, and efficiency throughout the software delivery lifecycle. Here are some key concepts and practices:

Key Concepts

1. **Infrastructure as Code (IaC)**: Use tools like Terraform, AWS CloudFormation, or Azure Resource Manager to manage infrastructure through code, enabling version control and repeatability.
2. **Continuous Integration/Continuous Deployment (CI/CD)**: Implement CI/CD pipelines using services like GitHub Actions, Jenkins, or AWS CodePipeline to automate the testing and deployment of applications.
3. **Monitoring and Logging**: Utilize cloud monitoring tools (like AWS CloudWatch, Azure Monitor, or Google Cloud Operations) to track application performance and gather logs for troubleshooting.
4. **Containerization**: Use Docker and orchestration platforms like Kubernetes or Amazon ECS to manage applications in containers, enhancing portability and scalability.
5. **Microservices Architecture**: Break down applications into smaller, manageable services that can be developed, deployed, and scaled independently.
6. **Collaboration Tools**: Leverage tools like Slack, Microsoft Teams, or Jira for improved communication and project management among teams.

Benefits of Cloud-based DevOps

- **Scalability**: Easily scale resources up or down based on demand.
- **Cost Efficiency**: Pay for only what you use, reducing overhead.
- **Rapid Deployment**: Quickly deploy and iterate applications, responding to market changes.

- **Global Accessibility**: Access resources and collaborate from anywhere, fostering remote work.

Best Practices

1. **Automate Everything**: From testing to deployment, automation reduces manual errors and speeds up processes.
2. **Version Control**: Use systems like Git to maintain code integrity and enable collaboration among team members.
3. **Security Integration**: Incorporate security practices (DevSecOps) throughout the development lifecycle to ensure applications are secure from the start.
4. **Continuous Feedback**: Gather feedback from users and stakeholders to inform future development and improvements.
5. **Documentation**: Maintain clear documentation of processes, configurations, and architectures to facilitate onboarding and knowledge sharing.

Tools and Technologies

- **CI/CD**: Jenkins, GitLab CI, CircleCI, AWS CodePipeline
- **IaC**: Terraform, Ansible, AWS CloudFormation
- **Containerization**: Docker, Kubernetes, OpenShift
- **Monitoring**: Prometheus, Grafana, Datadog
- **Collaboration**: GitHub, Bitbucket, Slack, Jira

Getting Started

1. **Assess Current Practices**: Evaluate your existing development and operational practices to identify areas for improvement.
2. **Choose the Right Tools**: Select cloud platforms and tools that align with your organization's needs and existing technology stack.
3. **Start Small**: Implement DevOps practices in a small project to demonstrate value before scaling to larger initiatives.
4. **Iterate and Improve**: Continuously refine processes based on feedback and performance metrics.

By leveraging cloud technologies in your DevOps strategy, you can create a more agile and responsive development environment that meets the demands of modern software delivery.

Conclusion:

Implementing DevOps using cloud services can significantly enhance your software development and delivery process. By leveraging automation, collaboration tools, and cloud capabilities, teams can achieve faster delivery cycles, improve quality, and foster a culture of continuous improvement. Embrace these practices, and you'll be well on your way to achieving a successful DevOps transformation in your organization.

Questions:

1. What is DevOps?
2. What are Benefits of Cloud-based DevOps

[Open Ended Experiments]
Experiment No. 10

Title: Explore Fog Computing Framework

Aim: To Explore Fog Computing Framework

Theory:
Fog computing or fog networking, also known as fogging, is pushing frontiers of computing applications, data, and services away from centralized cloud to the logical stream of the network edge. Fog networking system works on to build the control, configuration, and management over the Internet backbone rather than the primarily control by network gateways and switches those which are embedded in the LTE network. We can illuminate the fog computing framework as highly virtualized computing infrastructure which provides hierarchical computing facilities with the help of edge server nodes. These fog nodes organize the wide applications and services to store and process the contents in close proximity of end users. Sometimes, fog computing used frequently and often interchangeably the term "edge computing." However, there is a little bit difference between those two concepts. Fog and edge computing both involve pushing the processing and intelligence capabilities down to the proximity where the information is originating. The main difference between both architectures is exactly where the computing and intelligence power is placed. In both structures data is sent by the same sources or physical assets, like pumps, relays, motors, sensors, and so on.

Fog computing presents a hierarchical distributed architecture with support of the integration of technological components and services might be in near future like smart cities, smart grid system, connected vehicles, and smart homes

1. Definition and Overview

Fog computing is a decentralized computing framework that allows data processing, storage, and analysis to occur at or near the data source (the "fog") rather than relying solely on a centralized cloud infrastructure. This model supports IoT devices and applications, facilitating faster decision-making and improved resource utilization.

2. Key Components

- **Edge Devices**: These include IoT devices, sensors, and gateways that generate and send data.
- **Fog Nodes**: Intermediate devices (like routers and gateways) that process, store, and analyze data before sending it to the cloud.
- **Cloud Infrastructure**: Centralized data centers where heavy processing, storage, and long-term analytics occur.

3. Architecture

Fog computing architecture typically consists of three layers:

- **Edge Layer**: Composed of devices (e.g., sensors, cameras) that generate data.
- **Fog Layer**: Includes fog nodes that perform data processing, analytics, and storage closer to the edge devices. It may also involve local control and management.
- **Cloud Layer**: Centralized data centers that handle extensive data storage, long-term analytics, and complex computations.

4. Key Features

- **Low Latency**: By processing data closer to the source, fog computing reduces the time it takes to get insights from data, which is crucial for real-time applications.
- **Scalability**: The architecture allows for easy scaling by adding more fog nodes and edge devices without overloading the cloud.
- **Data Privacy**: Sensitive data can be processed locally, reducing the risk of exposure during transmission to the cloud.
- **Network Efficiency**: Reduces bandwidth consumption by filtering and aggregating data before sending it to the cloud.

5. Use Cases

Fog computing is particularly beneficial in various applications, including:

- **Smart Cities**: Managing traffic lights, surveillance systems, and public transportation using real-time data from various sensors.
- **Healthcare**: Enabling remote patient monitoring, wearable health devices, and real-time data analysis for better patient outcomes.
- **Industrial IoT**: Monitoring machinery and equipment in real-time, allowing for predictive maintenance and reducing downtime.
- **Autonomous Vehicles**: Processing data from sensors in real-time for navigation and decision-making.

6. Challenges

While fog computing offers numerous advantages, it also presents challenges:

- **Security**: With data being processed across multiple nodes, ensuring data security and privacy becomes more complex.
- **Interoperability**: Different devices and platforms must communicate seamlessly, which can be challenging due to varying protocols.
- **Management and Orchestration**: Managing distributed resources and ensuring efficient data flow can be complex.

7. Frameworks and Standards

Various frameworks and standards are being developed to facilitate fog computing:

- **OpenFog Reference Architecture**: Developed by the OpenFog Consortium, this framework outlines the architecture, functional components, and communication protocols for fog computing.
- **IoT Frameworks**: Many IoT platforms (like AWS IoT, Microsoft Azure IoT, and Google Cloud IoT) are incorporating fog computing capabilities.

8. Future of Fog Computing

As IoT devices proliferate and the need for real-time data processing grows, fog computing is expected to play a vital role in enabling smarter applications and infrastructure. Its integration with technologies like AI and machine learning will enhance its capabilities for predictive analytics and decision-making at the edge.

Conclusion

Department of Data Science

Fog computing represents a significant shift in how we process and manage data in an increasingly connected world. By bringing computation closer to the data source, it addresses key challenges in latency, bandwidth, and data privacy, making it an essential framework for the future of IoT and real-time applications. As the technology matures, its adoption will likely expand across various industries, driving innovation and efficiency.

Questions:

1. What is Fog Computing Framework?
2. Key Features of Fog Computing