Experiment No 1

Title: Study of NIST model of cloud computing.

Objective: Understand deployment models, service models, advantages of cloud computing

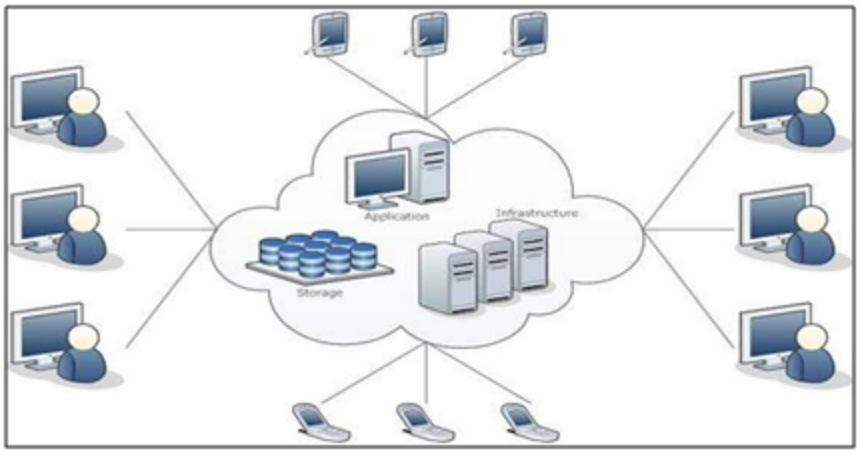
Theory:

What is Cloud?

The term **Cloud** refers to a **Network** or **Internet.** In other words, we can say that Cloud is something, which is present at a remote location. Cloud can provide services over public and private networks, i.e., WAN, LAN or VPN. Applications such as e-mail, web conferencing, customer relationship management (CRM) execute on cloud.

What is 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 offers **platform independency,** as the software is not required to be installed locally on the PC. Hence, the Cloud Computing is making our business applications **mobile** and **collaborative.**

Basic Concepts

There are certain services and models working behind the scene making the cloud computing feasible and accessible to end users. Following are the working models for cloud computing:

- Deployment Models
- Service Models

Deployment Models

Deployment models define the type of access to the cloud, i.e., how the cloud is located? Cloud can have any of the four types of access: Public, Private, Hybrid, and Community.

PUBLIC CLOUD

The **public cloud** allows systems and services to be easily accessible to the general public. Public cloud may be less secure because of its openness.

PRIVATE CLOUD

The **private cloud** allows systems and services to be accessible within an organization. It is more secured because of its private nature.

COMMUNITY CLOUD

The **community cloud** allows systems and services to be accessible by a group of organizations.

HYBRID CLOUD

The **hybrid cloud** is a mixture of public and private cloud, in which the critical activities are performed using private cloud while the non-critical activities are performed using public cloud.

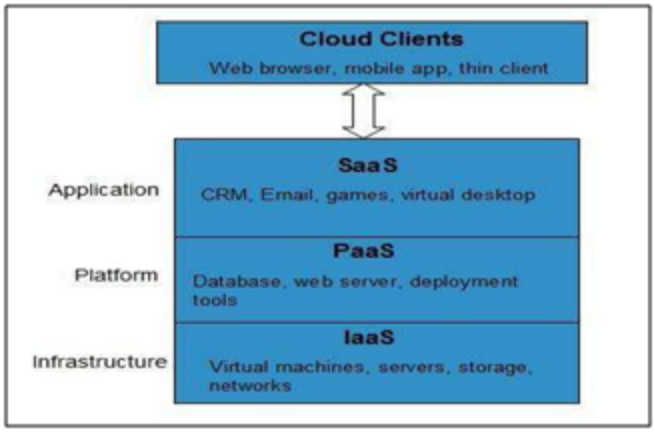
Service Models

Cloud computing is based on service models. These are categorized into three basic service models which are -

- Infrastructure-as–a-Service (IaaS)
- Platform-as-a-Service (PaaS)
- Software-as-a-Service (SaaS)

Anything-as-a-Service (XaaS) is yet another service model, which includes Network-as-a-Service, Business-as-a-Service, Identity-as-a-Service, Database-as-a-Service or Strategy-as-a-Service.

The **Infrastructure-as-a-Service (IaaS)** is the most basic level of service. Each of the service models inherit the security and management mechanism from the underlying model, as shown in the following diagram:



INFRASTRUCTURE-AS-A-SERVICE (IAAS)

IaaS provides access to fundamental resources such as physical machines, virtual machines, virtual storage, etc.

PLATFORM-AS-A-SERVICE (PAAS)

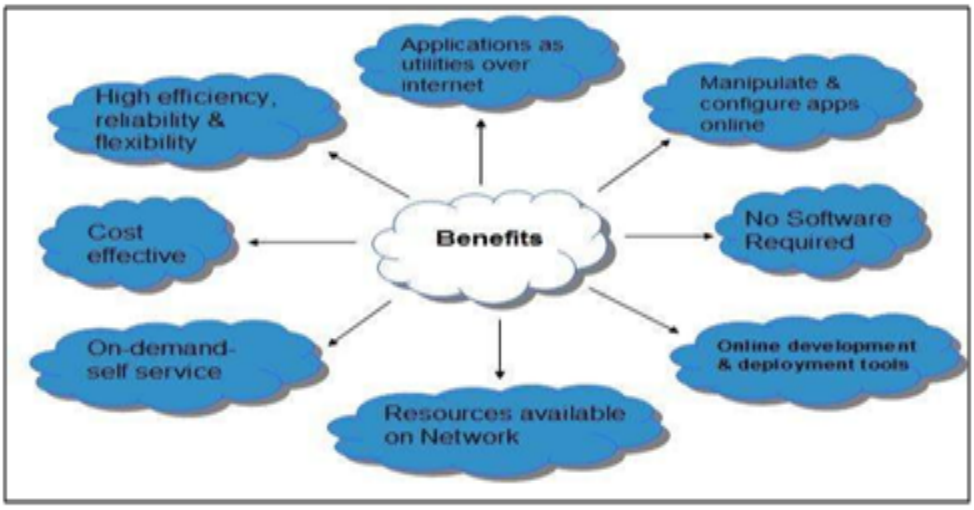
PaaS provides the runtime environment for applications, development and deployment tools, etc.

SOFTWARE-AS-A-SERVICE (SAAS)

SaaS model allows software applications to be used as a service to end-users.

History of Cloud Computing

The concept of **Cloud Computing** came into existence in the year 1950 with implementation of mainframe computers, accessible via **thin/static clients.** Since then, cloud computing has evolved from static clients to dynamic ones and from software to services. The following diagram explains the evolution of cloud computing:



Risks related to Cloud Computing

Although cloud Computing is a promising innovation with various benefits in the world of computing, it comes with risks. Some of them are discussed below:

a. Security and Privacy

It is the biggest concern about cloud computing. Since data management and infrastructure management in cloud is provided by third-party, it is always a risk to handover the sensitive information to cloud service providers. Although the cloud computing vendors ensure highly secured password protected accounts, any sign of security breach may result in loss of customers and businesses.

b. Lock In

It is very difficult for the customers to switch from one **Cloud Service Provider (CSP)** to another. It results in dependency on a particular CSP for service.

c. Isolation Failure

This risk involves the failure of an isolation mechanism that separates storage, memory, and routing between the different tenants.

d. Management Interface Compromise

In the case of public cloud providers, the customer management interfaces are accessible through the Internet.

e. Insecure or Incomplete Data Deletion

It is possible that the data requested for deletion may not get deleted. It happens because either of the following reasons

- Extra copies of data are stored but are not available at the time of deletion
- Disks that store data of multiple tenants are destroyed.

Characteristics of Cloud Computing

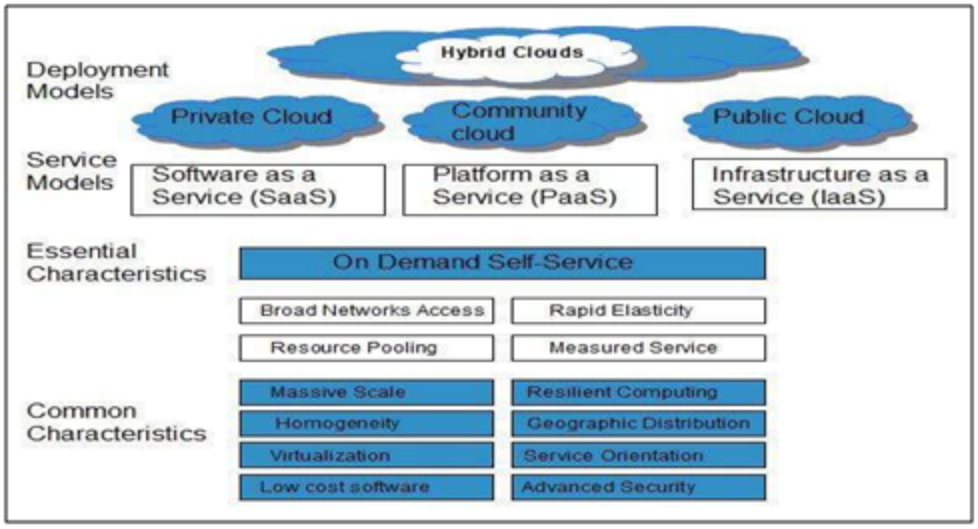
There are five key characteristics of cloud computing.

1. On Demand Self Service

Cloud Computing allows the users to use web services and resources on demand. One can logon to a website at any time and use them.

2. Broad Network Access

Since cloud computing is completely web based, it can be accessed from anywhere and at any time.



3. Resource Pooling

Cloud computing allows multiple tenants to share a pool of resources. One can share single physical instance of hardware, database and basic infrastructure.

4. Rapid Elasticity

It is very easy to scale the resources vertically or horizontally at any time. Scaling of resources means the ability of resources to deal with increasing or decreasing demand.

The resources being used by customers at any given point of time are automatically monitored.

5. Measured Service

In this service cloud provider controls and monitors all the aspects of cloud service. Resource optimization, billing, and capacity planning etc. depend on it.

Conclusion: We studied deployment models, service models, types of cloud computing in detail.

Experiment No 2

Title: Virtualization.
Objective: Understand different types of virtualizations, Host and bare metal hypervisors and implement horizontal scalability.

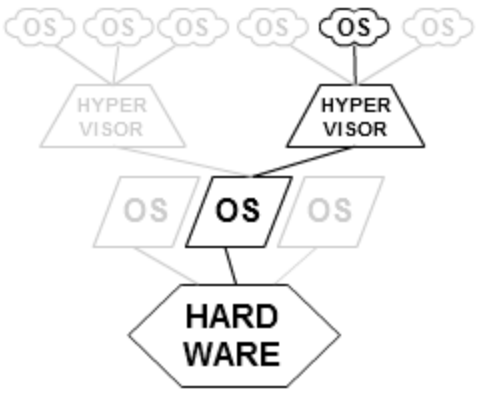
Technology: XEN/ Vmwares EXSi

PART A

Aim-: To Study and implement Desktop Virtualization
Theory:

Desktop virtualization is the concept of separating the logical desktop from the physical machine. Virtualization is the ability to run multiple operating systems on a single physical system and share the underlying hardware resources. In cloud computing Virtualization can be achieved using Hypervisor. A hypervisor, also known as a virtual machine manager/monitor (VMM), is computer hardware platform virtualization software that allows several operating systems to share a single hardware host. Each operating system appears to have the host's processor, memory, and resources to it. There are various hypervisors Products are available in the market some of them are- Xen, KVM, VMWare, Oracle Virtual Box.

Desktop virtualization is software technology that separates the desktop environment and associated application software from the physical client device that is used to access it.The Desktop Virtualization is based on Hosted Hypervisor where Vm runs over base os.The hosted hypervisors run within a conventional operating-system environment. With the hypervisor layer as a distinct second software level, guest operating-systems run at the third level above the hardware. The common examples of hosted hypervisor are VMware Workstation,Oracle Vm, Virtual Box,Qemu etc.



Hosted Hypervisor

Steps to implement Desktop Virtualization

Step 1-: Install Desktop Virtualization Software like VMware WorkStation or Oracle Virtual box Vm on Existing Operating System.

Step 2-: Once the installation is Over Create new Vm by clicking New Button which runs host OS allows you to Specify Resources for Guest VM like VCPU,Memory,Disk Size, network Configuration etc.

Step 3-: Once Vm is created Start the installation by Clicking Start button and Specify path of iso file through which installation will be done.

Step 4 -: Once OS is installed it can be accessed inside the VMBOX.

Screenshots:

Conclusion -: As Desktop Virtualization intended to run multiple Desktop OS on Single machine but it's not feasible if the base machine has lower hardware Configuration and hosted hypervisor logically shares Hardware resources to Guest OS so there is no guarantee of getting resources at critical time.

Experiment No 3

Aim-: To Study and implement Software as a Service using Ulteo

Theory

Ulteo Open Virtual Desktop (OVD) is Open Source virtual desktop and application delivery platform that supports Windows Remote Desktop Services,Linux hosted desktop and application sessions. Ulteo system mainly includes three parts: Ulteo Application server (Linux), Ulteo Session Manager Server & Admin Console and Ulteo Application server (Windows). Ulteo Session Manager Server & Admin Console is mainly responsible for delivering and managing user sessions with their user, group management. Ulteo Application server (Linux) offers users for the Linux applications; Ulteo Application server (Windows) offers users for the Windows application programs. The desktop programs include two types of applications: one group is based on the Linux application, another group is based on the Windows application

Procedure: Installation of Ulteo OVD Server

1) If you haven't an Ulteo OVD DVD-ROM yet, please download the corresponding ISO file from this place at www.ulteo.com and burn it to a fresh DVD.
2) Insert the Ulteo OVD DVD-ROM into your computer and restart it. If you selected the DVD-ROM as first boot device you'll see the boot loader Screen.
3) Select Install Ulteo Option
4) The first step is used to select the system language. Choose your language from the list and click on Forward.
5) In the second step, the system asks you to define your location. Either select a point on the map or choose one from the Selected city form and click on Forward.

6) The third step is used to define the keyboard layout. Select yours and click on Forward.
7) Then, you have to select the partitioning method. We suggest the automatic method: Erase and use the entire disk.
8) These questions are about the installed operating system itself, user login and password used to access the OS, along with the hostname of the machine.
9) Type a password and confirm it. Useful address is displayed to you for a near future use of OVD.
10) Then read carefully the installation summary, then click on Install and wait til installation completes
11) Finally, click on Restart now to finish installation process.
Once the installation is done OVD should be configured by using ip address of Session.To Configure it admin console need to be open using ipaddress/ovd/admin on browser where admin can install softwares,can create groups,can creates users or can assign applications to users.At client side user will get Hosted application with linux Desktop on browser.
Screenshots:
Conclusion -:Thus we have successfully studied implementation of SaaS using Ulteo.

Experiment No 4

Aim-: To Study and implement AWS Services-IAM,Cloudwatch and EC2
Theory:

IAM:AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely. Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.IAM is a feature of your AWS account offered at no additional charge. You will be charged only for use of other AWS services by your users.

Cloudwatch: Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers (SREs), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers. You can use CloudWatch to detect anomalous behavior in your environments, set alarms,

visualize logs and metrics side by side, take automated actions, troubleshoot issues, and discover insights to keep your applications running smoothly.

EC2(Infrastructure as a Service(IaaS): Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 offers the broadest and deepest compute platform with choice of processor, storage, networking, operating system, and purchase model. We offer the fastest processors in the cloud and we are the only cloud with 400 Gbps ethernet networking. We have the most powerful GPU instances for machine learning training and graphics workloads, as well as the lowest cost-per-inference instances in the cloud. More SAP, HPC, Machine Learning, and Windows workloads run on AWS than any other cloud.

Procedure:

a)Creation of AWS account:

b)Identity Access Management(IAM)

1)Go to IAM service once you login with AWS.
2) Go to create user and groups
3) Create users and groups into it
4) Enable Multi-factor authentication in the root user and check for logging in using MFA with the help of Google Authenticator app.
5) Check for logging in with the created user.

c)Cloudwatch

1) Go to Cloudwatch service once you login to AWS
2) Create a billing alarm, which will notify if the usage reaches some $ value, since in AWS we pay as per usage
3) Create a Simple Notification Service(SNS) and add the email id on which you will get the notification once usage reaches the set limit set in billing alarm
4) Check the configured billing alarm

d)EC2

1) Go to EC2 service once you login to AWS
2) Launch Instance in EC2 with the amazon image of linux
3) Select the resources mentioned for the free version account.
4) Once, the instance is launched, try for firing some linux commands

5) The next step is to add chrome extension secure shell
6) Using secure shell app, try to access the EC2 image instance created and fire some linux commands into it

Screenshots:

Conclusion -:Thus we have successfully studied implementation of AWS Services-IAM,Cloudwatch and EC2

Experiment No 5

Aim-: To Study and implement Storage as a Service(SaaS) using AWS Service S3.

Theory:

S3:

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements. Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.

Procedure:

1)Go to S3 service once you login with AWS.
2) Click on the create bucket to create a location for storage of objects/files.
3) The bucket name should be unique world-wide while creating.

4) Once the bucket is created, upload some files from your machine/PC.
4) Once you click on the objects, you can change the default permissions of block public access.
5) Check for accessing the object url after changing permissions, whether you are able to view the file.

Screenshots:
Conclusion -:Thus we have successfully studied implementation of AWS Service S3.

Experiment No 6

Aim-: To Study and implement Platform as a Service(PaaS) using Google appEngine.
Theory:
Google appEngine:

Build highly scalable applications on a fully managed serverless platform. New customers get $300 in free credits to spend on Google Cloud during the first 90 days. All Google Cloud customers get 28 instance hours per day free of charge.

Key features:Popular programming languages:Build your application in Node.js, Java, Ruby, C#, Go, Python, or PHP—or bring your own language runtime. Open and flexible:Custom runtimes allow you to bring any library and framework to App Engine by supplying a Docker container. Fully managed:A fully managed environment lets you focus on code while App Engine manages infrastructure concerns.

Procedure:

a. Creation of Google Cloud Platform account

b. Deployment of Java application on Google App Engine

1. Login to gcloud and create new project

2. For activating cloud shell, click on icon at right top
3. For making it a full page, click on open in new window
4. Click on pencil button, i.e. to open editor, its a gui editor
5. Check whether appengine api is enabled?
Api and services---dashboard---type appengine-click on enable api and services-type appengine-click on google cloud app flexible environ-click on enable api
6. Go to github login
GoogleCloudPlatform/java-docs-samples
7.goto cloud shell environ
8.fire command Git clone {copy repository path from clone button of github}
The files cloned in our local directory will be displayed in white gui
9.navigate to apppengine-java folder
Cd java-docs-samples/appengine-java8
Cd helloworld/
10. Show helloappengine.java file by browsing in gui as
Helloworld-gradle-src-main-java-com-example-appengine-java8-Helloappengine.java which implements simple htttp servlet
11.testing application using development server, to get the development server running, u will download maven to manage compiling ur app and starting the development server
Fire command
Mvn clean
Build success
Fire command
Mvn package
2-3 min
To download and install maven and run the application fire command
Mvn appengine:run
Take some time, 5 min due to some dependency, appserver is now running
12. Go to eye button at right top-click on preview on port 8080, will run the java file in local
13. Deploy on appengine so that whole world can see
Ctrl c
Goto pom.xml in gui and click on config...projectid and add ur project id in code
14.mvn package appengine:deploy
For deploying ap to appengine
15.gcloud app browse (for getting link of application)
Click on link ..or copy link in browser, u will get the output of java file

Screenshots:

Conclusion -:Thus we have successfully studied implementation of Platform as a Service(PaaS) using Google appEngine.

Experiment No 7

Aim-: To Study and implement AWS Cloudfront service.

Theory:

AWS Cloudfront service:

Amazon CloudFront is a fast content delivery network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.

CloudFront offers the most advanced security capabilities, including field level encryption and HTTPS support, seamlessly integrated with AWS Shield, AWS Web Application Firewall and Route 53 to protect against multiple types of attacks including network and application layer DDoS attacks. These services co-reside at edge networking locations – globally scaled and connected via the AWS network backbone – providing a more secure, performant, and available experience for your users.

CloudFront works seamlessly with any AWS origin, such as Amazon S3, Amazon EC2, Elastic Load Balancing, or with any custom HTTP origin. You can customize your content delivery through CloudFront using the secure and programmable edge computing feature AWS Lambda@Edge.

Procedure:

1. Login to AWS management console.
2. Browse to Cloudfront service from the services menu.
3. Click on create new distribution.
4. Keep all default data filled in like domain name and select S3 bucket created before.
5. Click on create distribution as a final step of creation.
6. Once the distribution is created, copy the url and add the name of the file of S3 bucket after the url into the browser.
7.The file will be opened using URL of CDN distribution created.

Screenshots:

Conclusion -:Thus we have successfully studied and implemented the AWS Cloudfront service.

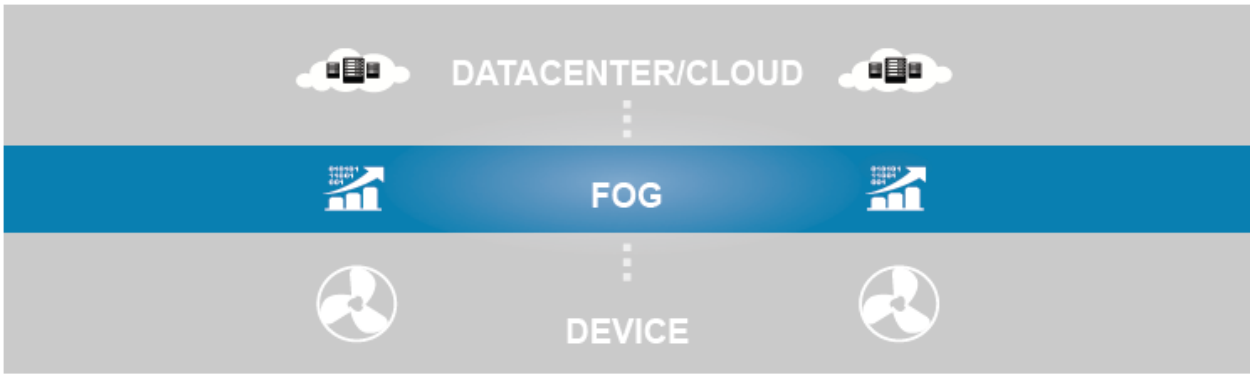
Experiment No 8

Aim-: To Study the case study on Fog Computing.

Theory:

Fog Computing:What Is It? The fog extends the cloud to be closer to the things that produce and act on IoT data (Figure 1). These devices, called fog nodes, can be deployed anywhere with a network connection: on a factory floor, on top of a power pole, alongside a railway track, in a vehicle, or on an oil rig. Any device with computing, storage, and network connectivity can be a fog node. Examples include industrial controllers, switches, routers, embedded servers, and video surveillance cameras. IDC estimates that the amount of data analyzed on devices that are physically close to the Internet of Things is approaching 40 percent.1 There is good reason: analyzing IoT data close to where it is collected minimizes latency. It offloads gigabytes of network traffic from the core network, and it keeps sensitive data inside the network. Analyzing IoT data close to where it is collected minimizes latency. It offloads gigabytes of network traffic from the core network. And it keeps sensitive data inside the network.

Figure 1. The Fog Extends the Cloud Closer to the Devices Producing Data



Examples of Fog Applications

Fog applications are as diverse as the Internet of Things itself. What they have in common is monitoring or analyzing real-time data from network-connected things and then initiating an action. The action can involve machine-to-machine (M2M) communications or human-machine interaction (HMI). Examples include locking a door, changing equipment settings, applying the brakes on a train, zooming a video camera, opening a valve in response to a pressure reading, creating a bar chart, or sending an alert to a technician to make a preventive repair. The possibilities are unlimited.

Production fog applications are rapidly proliferating in manufacturing, oil and gas, utilities, transportation, mining,and the public sector.

When to Consider Fog Computing

● Data is collected at the extreme edge: vehicles, ships, factory floors, roadways, railways, etc.

● Thousands or millions of things across a large geographic area are generating data.
● It is necessary to analyze and act on the data in less than a second.

How Does Fog Work?

Developers either port or write IoT applications for fog nodes at the network edge. The fog nodes closest to the network edge ingest the data from IoT devices. Then—and this is crucial—the fog IoT application directs different types of data to the optimal place for analysis, as shown in Table 1:

● The most time-sensitive data is analyzed on the fog node closest to the things generating the data. In a Cisco Smart Grid distribution network, for example, the most time-sensitive requirement is to verify that protection and control loops are operating properly. Therefore, the fog nodes closest to the grid sensors can look for signs of problems and then prevent them by sending control commands to actuators.
● Data that can wait seconds or minutes for action is passed along to an aggregation node for analysis and action. In the Smart Grid example, each substation might have its own aggregation node that reports the operational status of each downstream feeder and lateral.
● Data that is less time sensitive is sent to the cloud for historical analysis, big data analytics, and long-term storage (see sidebar). For example, each of thousands or hundreds of thousands of fog nodes might send periodic summaries of grid data to the cloud for historical analysis and storage.

Table 1. Fog Nodes Extend the Cloud to the Network Edge

	Fog Nodes Closest to IoT Devices	Fog Aggregation Nodes	Cloud
Response time	Milliseconds to subsecond	Seconds to minutes	Minutes, days, weeks
Application examples	M2M communication Haptics[2], including telemedicine and training	Visualization Simple analytics	Big data analytics Graphical dashboards
How long IoT data is stored	Transient	Short duration: perhaps hours, days, or weeks	Months or years
Geographic coverage	Very local: for example, one city block	Wider	Global

What Happens in the Fog and the Cloud

Fog nodes:

● Receive feeds from IoT devices using any protocol, in real time
● Run IoT-enabled applications for real-time control and analytics, with millisecond response time
● Provide transient storage, often 1–2 hours
● Send periodic data summaries to the cloud

The cloud platform:

● Receives and aggregates data summaries from many fog nodes

● Performs analysis on the IoT data and data from other sources to gain business insight
● Can send new application rules to the fog nodes based on these insights

Benefits of Fog Computing

Extending the cloud closer to the things that generate and act on data benefits the business in the following ways:

● Greater business agility: With the right tools, developers can quickly develop fog applications and deploy them where needed. Machine manufacturers can offer MaaS to their customers. Fog applications program the machine to operate in the way each customer needs.
● Better security: Protect your fog nodes using the same policy, controls, and procedures you use in other parts of your IT environment. Use the same physical security and cybersecurity solutions.
● Deeper insights, with privacy control: Analyze sensitive data locally instead of sending it to the cloud for analysis. Your IT team can monitor and control the devices that collect, analyze, and store data.
● Lower operating expense: Conserve network bandwidth by processing selected data locally instead of sending it to the cloud for analysis.

Conclusion -:Thus we have successfully studied case study on Fog Computing.