



**Semester: B.E. Eighth Semester (CBS)**  
**Subject: Clustering & Cloud Computing**  
**Unit-1 Solution**

---

**Q.1 What are different legal issues in cloud computing.(6M)(s-17)**

**Ans:**

**Cloud computing legal issues: data location**

Organizations need to know where the data they're responsible for – both personal customer data and corporate information -- will be located at all times. In the cloud environment, location matters, especially from a legal standpoint.

Cloud computing legal issues result from where a cloud provider keeps data, including application of foreign data protection laws and surveillance. In this tip, learn about cloud computing legal issues stemming from data location, and how to avoid them.

**Cloud computing contracts and cloud outages**

When a cloud service goes down, users lose access to their data and therefore may be unable to provide services to their customers. When is a cloud user compensated for the loss of service, and to what extent? Users need to examine how cloud computing contracts account for cloud outages.

This tip discusses how a cloud outage could negatively affect business and examines some cloud computing contracts and their provisions for cloud outages.

**Cloud computing contracts: Tread carefully**

Organizations must be careful with cloud computing contracts, according to a panel of lawyers at the RSA Conference 2011. Cloud computing contracts should include many data protection provisions, but cloud computing service providers may not agree to them.

In this article, the RSA Conference 2011 panel offers advice on negotiating with cloud computing service providers and on legal considerations for organizations entering cloud service provider contracts, including data security provisions.

**Ten key provisions in cloud computing contracts**

When entering into a relationship with a cloud computing service provider, companies should pay attention to contract terms, security requirements and several other key provisions when negotiating cloud computing contracts.

Here, cloud legal expert Francoise Gilbert discusses cloud computing contracts and the ten key

provisions that companies should address when negotiating contracts with cloud computing service providers.

### **Developing cloud computing contracts**

Cloud service relationships can be complicated. The use of cloud services could sacrifice an entity's ability to comply with several laws and regulations and could put sensitive data at risk. Consequently, it's essential for those using cloud computing services to understand the scope and limitations of the services they receive, and the terms under which these services will be provided.

In this tip, Francoise Gilbert explains the critical considerations for cloud computing contracts in order to protect your organization as well as reviewing the critical steps and best practices for developing, maintaining and terminating cloud computing contracts.

---

### **Q2. Explain with diagram cloud computing stack?(7M)(S-17)**

**Ans:**

Cloud Computing is a model which enables useful, on-demand network access to a shared configurable resources (networks, storage, servers, services, and applications) which can be quickly implemented and released with minimum effort. According to the definition, these are the characteristics every cloud solution should have:

#### **On-demand self-service.**

Ability to access the service from standard platforms (desktop, laptop, tablet, mobile, etc.).

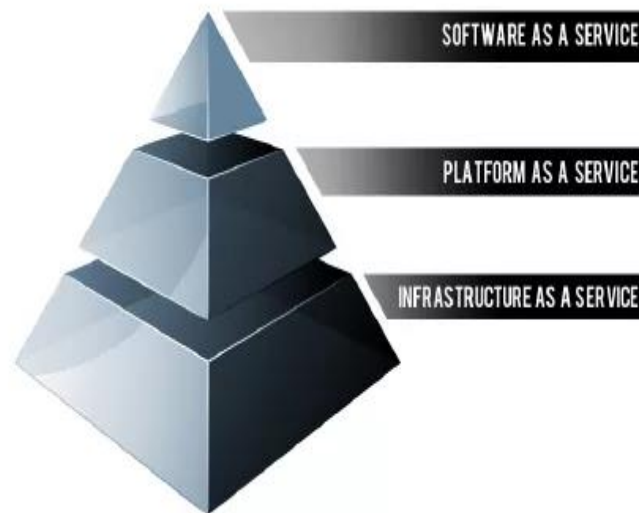
Resource pooling.

Capability to scale resources in order to cope with demand peaks.

Measured billing which is delivered as a service.

This needs to be clearly stated because in the last couple of years as the cloud has continued to explode, many traditional software vendors have tried to sell their solutions as "cloud computing" offerings even though they do not fit the definition. The diagram below shows the widely accepted Cloud Computing stack – it depicts three categories within Cloud – Infrastructure as a Service, Platform as a Service, and Software as a Service.

### **Cloud Computing Stack**



IaaS is the backbone of the cloud. The software and hardware that run the whole show from servers, switches, load balancers, etc. Amazon Web Services is the largest and most well known in this category (although they do offer the full stack, including PaaS and IaaS).

PaaS is a pack of services and tools made to make coding and deploying of applications easy and efficient in the cloud. This is not infrastructure, but rather add-on services that allow you to easier deploy, manage, and scale your infrastructure.

SaaS consists of software applications that are hosted in the cloud and delivered over the Internet to a consumer or enterprise. Microsoft Office 365 is a common example of a SaaS application, although there are countless others.

### **Infrastructure as a Service (IaaS)**

IaaS is an on-demand delivery of Cloud Computing infrastructure (storage, servers, network, etc.). Instead of buying servers, network equipment, space and software clients buy these resources as a completely outsourced on-demand service.

Here are some main IaaS characteristics:

Resources are delivered as a service Enables dynamic scaling Offers utility pricing model with variable costs Typically includes multiple users on a single piece of hardware.

The largest IaaS providers in the world at the moment are Amazon Web Services, Microsoft Azure, and Rackspace. Here are some situations that are suitable for IaaS:

a company is growing quickly and scaling of resources is too complicated demand is constantly changing – with significant ups and downs in terms of infrastructure demand new businesses without the capital for infrastructure investment

### **Platform as a Service (PaaS)**

PaaS is a platform that enables the creation of different software and applications with ease and without the need to buy the software or infrastructure needed for the job.

Some of the characteristics of PaaS:

Offers services for development, testing, deployment, hosting and maintenance of applications in the same environment. Enables UI creation, modifying and deployment with user interface creation tools Multi-user development architecture Offers scalability of used software Subscription and billing tools are also a part of the PaaS PaaS is particularly useful where many developers are working on a project together or where other parties need to interact with the process. It is also commonly used for automated testing and deployment services.

### **Software as a Service (SaaS)**

With SaaS, an application license is provided to customers either as a “pay-as-you-go” model, through a subscription, or as an on-demand service. SaaS is, like other cloud services, quickly growing and soon you’ll be able to find it almost everywhere. For that particular reason, we must know when and where it should be used. Here are main characteristics of SaaS:

---

### **Q.3) What is cloud computing? Explain essential characteristics of cloud computing?(6M)(S-17)**

**Ans:**

**Def:**

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

#### **The five essential characteristics of cloud computing:**

**On-demand self-service:** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

**Broad network access:** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops and workstations).

**Resource pooling:** The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state or datacenter). Examples of resources include storage, processing, memory and network bandwidth.

**Rapid elasticity:** Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

**Measured service:** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth and active user accounts). Resource usage can be monitored, controlled and reported, providing transparency for the provider and consumer.

**Q4)What are the difference between cluster computing. grid computing and cloud computing.(7M) (W-18)(S-17)(W-17)(W-16)**

**Ans:**

Cluster Computing	Grid Computing	Cloud Computing
Characteristics of Cluster computing 1:Tightly coupled systems 2: Single system image 3: Centralized Job management & scheduling system	Characteristics of Grid Computing 1: Loosely coupled (Decentralization) 2: Diversity and Dynamism 3: Distributed Job Management & scheduling	Characteristic of cloud computing 1: Dynamic computing infrastructure 2: IT service-centric approach 3: Self-service based usage model 4: Minimally or self-managed platform 5: Consumption-based billing
In cluster computing, a bunch of similar (or identical) computers are hooked up locally (in the same physical location, directly connected with very high speed connections) to operate as a single computer	In grid computing, the computers do not have to be in the same physical location and can be operated independently. As far as other computers are concerned each computer on the grid is a distinct computer.	In cloud computing, the computers need not to be in the same physical location.
The cluster computers all have the same hardware and OS.	The computers that are part of a grid can run different operating systems and have different hardware	The memory, storage device and network communication are managed by the operating system of the basic physical cloud units. Open source software such as LINUX can support the basic physical unit management and virtualization computing.
The whole system (all nodes) behaves like a single system view and resources are managed by centralized resource manager.	Every node is autonomous i.e. it has its own resource manager and behaves like an independent entity	Every node acts as an independent entity
The computers in the cluster are normally contained in a single location or complex.	Grid are inherently distributed by its nature over a LAN, metropolitan or WAN	Clouds are mainly distributed over MAN
More than 2 computers are connected to solve a problem	A large project is divided among multiple computers to make use of their resources.	It does just the opposite. It allows multiple smaller applications to run at the same time.
Areas of cluster computing 1. Educational resources 2.Commercial sectors for industrial promotion 3.Medical research	Areas of Grid Computing 1.Predictive Modeling and Simulations 2.Engineering Design and Automation 3.Energy Resources Exploration 4.Medical, Military and Basic Research 5.Visualization	Areas of cloud Computing 1.Banking 2. Insurance 3. Weather Forecasting 4.Space Exploration 5.Software as a service 6.PaaS 7.Infrastructure- as -a-Service
Commodity computers	High-end computers (servers, clusters)	Commodity computers and high-end servers and network attached storage
Size or scalability is 100s	Size or scalability is 1000s	Size or scalability is 100s to 1000s
One of the standard OSs (Linux, Windows)	Any standard OS (dominated by Unix)	A hypervisor (VM) on which multiple OSs run

Single Ownership	Multiple Ownership	Single Ownership
Dedicated, high-end with low latency and high bandwidth Interconnection Network	Mostly Internet with high latency and low Bandwidth Interconnection Network	Dedicated, high-end with low latency and high Bandwidth Interconnection Network
Traditional login/password-based. Medium level of privacy depends on user privileges.	Public/private key pair based authentication and mapping a user to an account. Limited support for privacy.	Each user/application is provided with a virtual machine. High security/privacy is guaranteed. Support for setting per-file access control list (ACL).
Membership services discovery	Centralized indexing and decentralized info services discovery	Membership services discovery
Limited service negotiation	Yes, SLA based service negotiation	SLA based service negotiation
User management is centralized	User management is decentralized and also virtual organization (VO)-based	User management is centralized or can be delegated to third party
Resource management is centralized	Resource management is distributed	Resource management is centralized/distributed
Virtual Interface Architecture (VIA)-based standards	Some Open Grid Forum standards	Web Services (SOAP and REST) standards
Single system image	No single system image	Yes, but optionally include Single system image
Stable and guarantee capacity	Varies, but high capacity	Provisioned on demand capacity
Failure management (Self-healing) is limited (often failed tasks/applications are restarted).	Failure management (Self-healing) is limited (often failed tasks/applications are restarted).	Strong support for failover and content replication. VMs can be easily migrated from one node to other.
Limited pricing of services but not open market	Pricing of services is dominated by public good or privately assigned	Utility pricing, discounted for larger customers
Multi-clustering within an Organization for internetworking	Limited adoption for internetworking, but being explored through research efforts such as Gridbus InterGrid	High potential, third party solution providers can loosely tie together services of different Clouds for internetworking.
Potential for building 3rd party or value-added solutions is limited due to rigid architecture	Potential for building 3rd party or value-added solutions is limited due to strong orientation for scientific Computing	High potential - can create new services by dynamically provisioning of compute, storage, and application services and offer as their own isolated or composite Cloud services to users

### 1.Cluster Computing:

Cluster computing it's a group of computers connected to each other and work together as a single computer. These computers are often linked through a LAN. The clusters came to existence for the high need for them, because the computing requirements are increasing in a high rate and there's more data to process, so the cluster has been used widely to improve performance. The cluster is a tightly coupled systems, and from its characteristics that it's a centralized job management and scheduling system. All the computers in the cluster use the same hardware and operating system, and the computers are the same physical location and connected with a very high speed connection to perform as a single computer. The resources of the cluster are managed by centralized resource manager. the cluster is single owned, to only one organization. Its interconnection network is a high-end with low latency and high bandwidth, the security in the cluster is a login/password-based, and it has a medium level of privacy depends on users privileges. it has a stable and guaranteed capacity. The self healing in the cluster is Limited, it's often restarts the failed tasks and applications. its service negotiations are limited, and the user management is centralized. The cluster computing is usually used in educational resources, commercial sectors for industrial promotion & Medical research.

### Architecture:

The architecture of cluster computing contains some main components and they are:

1. Multiple stand alone computers.
2. Operating system.
3. High performance interconnects.
4. Communication software.
5. Different application platforms [5]

#### **Advantages:**

- In the cluster software is automatically installed and configured, and the nodes of the cluster can be added and managed easily, so it's very easy to deploy, it's an open system, and very cost effective to acquire and manage, clusters have many sources of support and supply, it's fast and very flexible, the system is optimized for performance as well as simplicity and it can change software configurations at any time, also it saves the time of searching the net for latest drivers, The cluster system is very supportive as it includes software updates.

#### **Disadvantages:**

- Cluster computing contains some disadvantages such as that it's hard to be managed without experience, also when the size of cluster is large, it'll be difficult to find out something has failed, the programming environment is hard to be improved when software on some node is different from the other.

## **2.Grid Computing:**

Grid computing is a combination of resources from multiple administrative domains to reach a common target, and this group of computers can distributed on several location and each a group of grids can be connected to each other. The need of access to additional resources and the collaborating between organizations leads to the need for grid computing. Grid environments are extremely well suited to run jobs that can be split into smaller chunks and run concurrently on many nodes. The grid is a loosely coupled systems and from its characteristics that it's a distributed Job Management and scheduling, the computers in the grid are not required to be in the same physical location and can be operated independently, so each computer on the grid is concerned a distinct computer, the computers in the grid are not tied to only on operating system and can run different OSs and different hardware, when it comes to a large project, the grid divides it to multiple computers to easily use their resources. The grid is multiple owned, it could be owned by several companies. Interconnection network is mostly internet with high latency and low bandwidth. The security in the grid is public and private based on authentication and mapping user to an account. And it has limited support privacy, its capacity is not stable, it varies, but it's high. The self healing in the cluster is Limited, it's often restarts the failed tasks and applications. its service negotiations are based on service level agreements, and the user management is decentralized. The grid computing is usually used in predictive modeling and simulations, engineering design and automation, energy resources exploration, medical, military, basic Research and visualization.

#### **Architecture:**

- Fabric layer to provide the resources which shared access is mediated by grid computing.

- Connectivity layer and it means the core communication and authentication protocols required for grid specific network functions.
- Resource layer and it defines the protocols, APIs and SDK for secure negotiations, imitations, monitoring control, accounting and payment of sharing operations on individual resources.
- Collective layer which it contains protocols and services that capture interactions among a collection of resources.
- Finally the Application layer, and it's user applications that operate within VO environment.

#### **Advantages:**

One of the advantages of grid computing that you don't need to buy large servers for applications that can be split up and farmed out to smaller commodity type servers, secondly it's more efficient in use of resources. Also the grid environments are much more modular and don't have much points of failure. About policies in the grid it can be managed by the grid software, beside that upgrading can be done without scheduling downtime, and jobs can be executed in parallel speeding performance.

#### **Disadvantages:**

It needs fast interconnect between computers resources, and some applications may need to be pushed to take full advantage of the new model, and licensing across many servers may make it forbidden for some applications, and the grid environments include many smaller servers across various administrative domains. also political challenges associated with sharing resources especially across different admin domains.

### **3.Cloud Computing:**

Cloud computing is a term used when we are not talking about local devices which it does all the hard work when you run an application, but the term used when we're talking about all the devices that run remotely on a network owned by another company which it would provide all the possible services from e-mail to complex data analysis programs. This method will decrease the users' demands for software and super hardware. The only thing the user will need is running the cloud computing system software on any device that can access to the Internet . Cloud computing is useful for the small business companies to make their resources from external sources as well as for medium companies, the large companies have obtained the largest storage without the need to build internal storage centers, thus, the cloud computing has given for both small and large companies the ability to reduce the cost clearly. In return for these services, the providing companies for cloud computing requires a financial gain determined by use. The cloud is a dynamic computing infrastructure, IT service-centric approach; also it's a self-service based usage model and self-managed platform and its consumption based on billing, the computers in cloud computing are not required to be in the same physical place, wherever you are you will be served. The operating system of the basic physical cloud units manages the memory, the storage device and network communication. In the cloud you can use multiple operating systems at the same time. Every node in the cloud is an independent entity. It allows multiple smaller applications to run at the same time. The cloud is owned by only one company and it provides its services to the users, its interconnection network is a high-end with low latency and high bandwidth. The security in the cloud is high and the privacy is guaranteed, each user/application is provided with a virtual machine. Its capacity is provided on demand. The self healing in the cloud has a strong support for failover and content replication, and virtual machines can be easily migrated from one node to other. its service negotiations are based on service level agreements, and the user



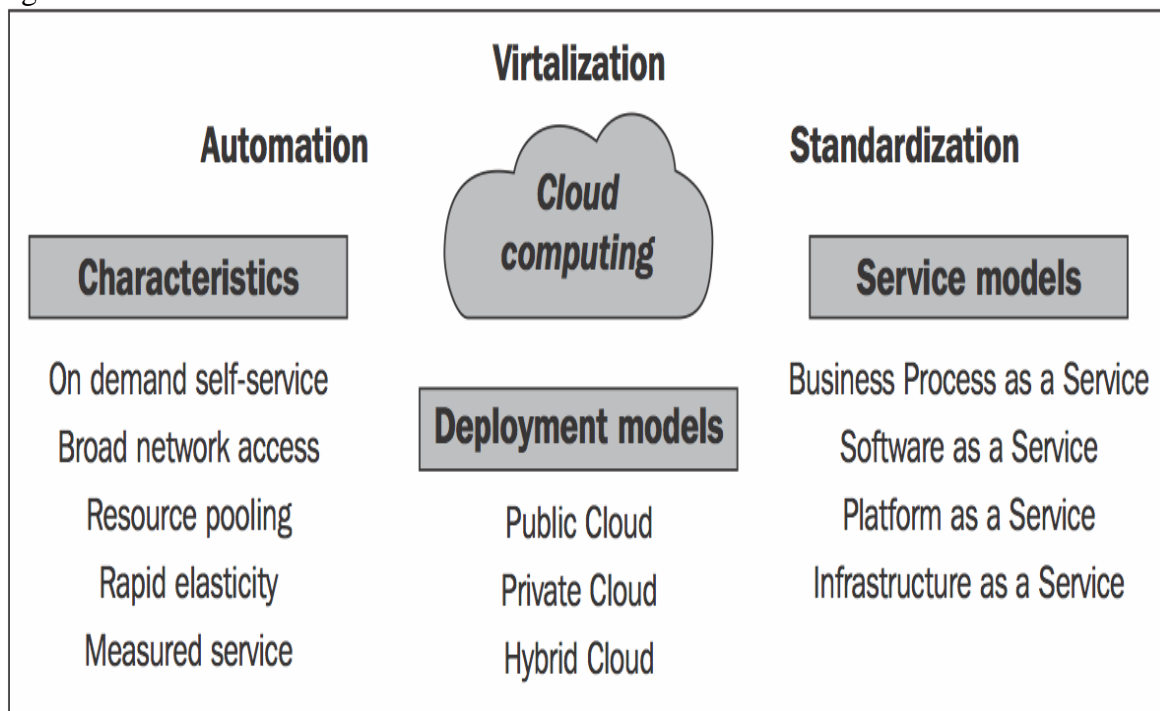
management is centralized or can be delegated to third party. The cloud computing is usually used in banking, insurance, weather forecasting, space exploration, software as a service, platform as a service, infrastructure as a service.

**Q5) What do you understand by cloud computing? Explain the characteristics of cloud computing. (7M)(W-18)(W-17)(W-16)**

*“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”*

Although this widely-adopted description of what makes a cloud computing solution is very valuable, it is not very tangible or easy to understand. So let's dive a little deeper into cloud computing and why it's different than just visualization alone, which is commonly mistaken to be cloud computing as well.

The following image shows that cloud computing is composed of five essential characteristics, three deployment models, and four service models as shown in the following figure:



Let's look a bit closer at each of the characteristics, service models, and deployment models in the next sections.

**Five essential characteristics of cloud computing**

The special publication includes the five essential characteristics of cloud computing:

1. **On-demand self-service:** A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
2. **Broad network access:** Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., **mobile** phones, tablets, laptops and workstations).

3. **Resource pooling:** The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state or datacenter). Examples of resources include storage, processing, memory and network bandwidth.
4. **Rapid elasticity:** Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.
5. **Measured service:** Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth and active user accounts). Resource usage can be monitored, controlled and reported, providing transparency for the provider and consumer.

#### =====

#### **Q6) ) Explain the challenges & legal issues in cloud computing.(7M)(W-17)(W-16)**

##### **Cloud computing legal issues: data location**

Organizations need to know where the data they're responsible for – both personal customer data and corporate information -- will be located at all times. In the cloud environment, location matters, especially from a legal standpoint.

Cloud computing legal issues result from where a cloud provider keeps data, including application of foreign data protection laws and surveillance. In this tip, learn about cloud computing legal issues stemming from data location, and how to avoid them.

##### **Cloud computing contracts and cloud outages**

When a cloud service goes down, users lose access to their data and therefore may be unable to provide services to their customers. When is a cloud user compensated for the loss of service, and to what extent? Users need to examine how cloud computing contracts account for cloud outages.

This tip discusses how a cloud outage could negatively affect business and examines some cloud computing contracts and their provisions for cloud outages.

##### **Cloud computing contracts: Tread carefully**

Organizations must be careful with cloud computing contracts, according to a panel of lawyers at the RSA Conference 2011. Cloud computing contracts should include many data protection provisions, but cloud computing service providers may not agree to them.

In this article, the RSA Conference 2011 panel offers advice on negotiating with cloud computing service providers and on legal considerations for organizations entering cloud service provider contracts, including data security provisions.

##### **Ten key provisions in cloud computing contracts**

When entering into a relationship with a cloud computing service provider, companies should pay attention to contract terms, security requirements and several other key provisions when negotiating cloud computing contracts.

Here, cloud legal expert Francoise Gilbert discusses cloud computing contracts and the ten key provisions that companies should address when negotiating contracts with cloud computing service providers.

### **Developing cloud computing contracts**

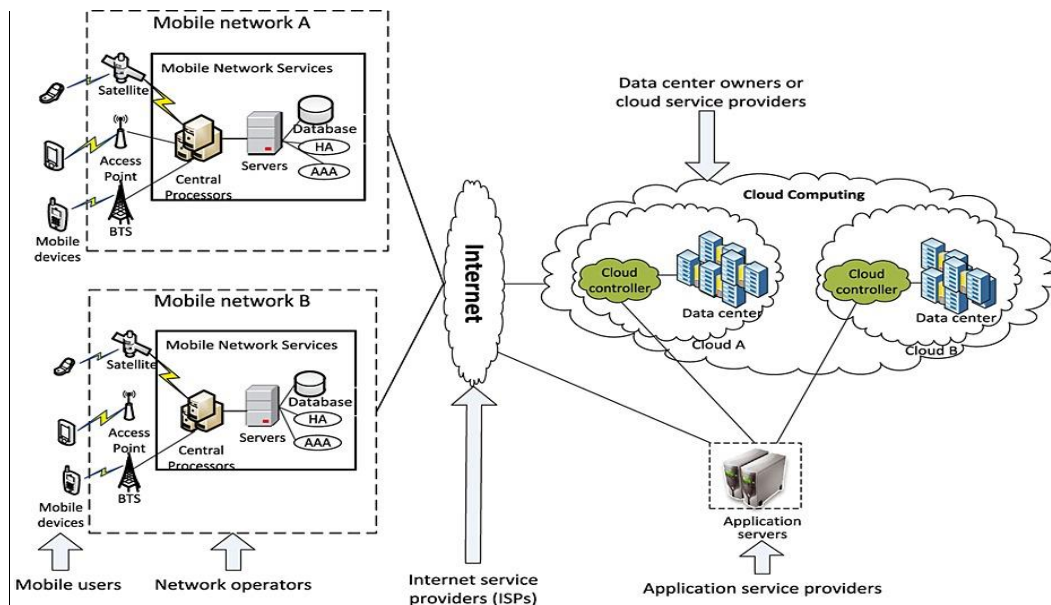
Cloud service relationships can be complicated. The use of cloud services could sacrifice an entity's ability to comply with several laws and regulations and could put sensitive data at risk. Consequently, it's essential for those using cloud computing services to understand the scope and limitations of the services they receive, and the terms under which these services will be provided.

In this tip, Francoise Gilbert explains the critical considerations for cloud computing contracts in order to protect your organization as well as reviewing the critical steps and best practices for developing, maintaining and terminating cloud computing contracts.

---

### **Q7)With the help of architecture give the overview of mobile cloud.(7M)(W-17)(W-16)**

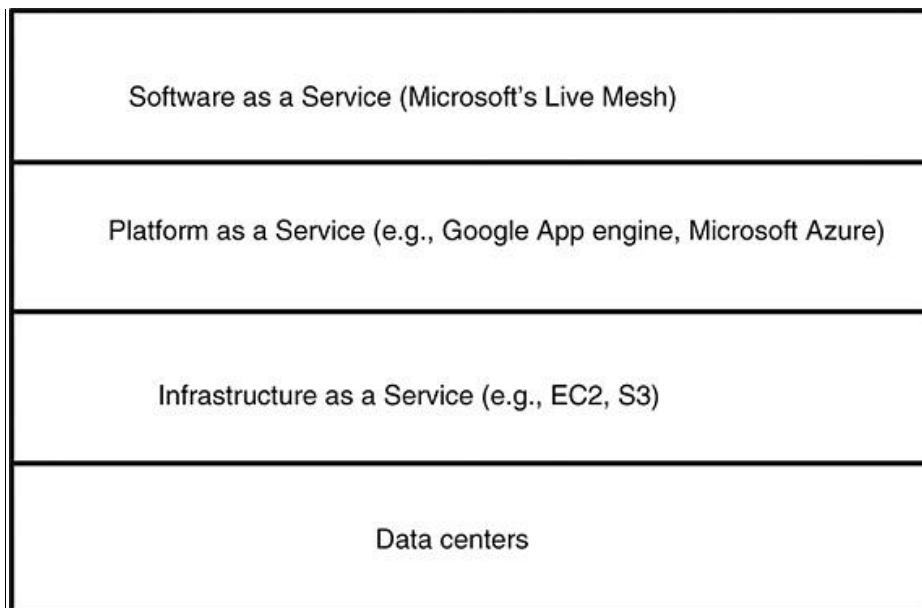
From the concept of MCC, the general architecture of MCC can be shown in Figure 1. In Figure 1, mobile devices are connected to the mobile networks via base stations (e.g., base transceiver station, access point, or satellite) that establish and control the connections (air links) and functional interfaces between the networks and mobile devices. Mobile users' requests and information (e.g., ID and location) are transmitted to the central processors that are connected to servers providing mobile network services. Here, mobile network operators can provide services to mobile users as authentication, authorization, and accounting based on the home agent and subscribers' data stored in databases. After that, the subscribers' requests are delivered to a cloud through the Internet. In the cloud, cloud controllers process the requests to provide mobile users with the corresponding cloud services. These services are developed with the concepts of utility computing, virtualization, and service-oriented architecture (e.g., web, application, and database servers).



**Figure 1:-Mobile cloud computing architecture.**

The details of cloud architecture could be different in different contexts. For example, a four-layer architecture is explained in [8](#) to compare cloud computing with grid computing. Alternatively, a service-oriented architecture, called Aneka, is introduced to enable developers to build. Microsoft.NET applications with the supports of application programming interfaces (APIs) and multiple programming models . presents an architecture for creating market-oriented clouds and proposes an architecture for web-delivered business services. In this paper, we focus on a layered architecture of CC (Figure [2](#)). This architecture is commonly used to demonstrate the effectiveness of the CC model in terms of meeting the user's requirements .

=====



**Figure 2 :- Service-oriented cloud computing architecture.**

Generally, a CC is a large-scale distributed network system implemented based on a number of servers in data centers. The cloud services are generally classified based on a layer concept (Figure 2). In the upper layers of this paradigm, Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) are stacked.

- **Data centers layer.** This layer provides the hardware facility and infrastructure for clouds. In data center layer, a number of servers are linked with high-speed networks to provide services for customers. Typically, data centers are built in less populated places, with a high power supply stability and a low risk of disaster.
- **IaaS.** Infrastructure as a Service is built on top of the data center layer. IaaS enables the provision of storage, hardware, servers, and networking components. The client typically pays on a per-use basis. Thus, clients can save cost as the payment is only based on how much resource they really use. Infrastructure can be expanded or shrunk dynamically as needed. The examples of IaaS are Amazon Elastic Cloud Computing and Simple Storage Service (S3).
- **PaaS.** Platform as a Service offers an advanced integrated environment for building, testing, and deploying custom applications. The examples of PaaS are Google App Engine, Microsoft Azure, and Amazon Map Reduce/Simple Storage Service.
- **SaaS.** Software as a Service supports a software distribution with specific requirements. In this layer, the users can access an application and information remotely via the Internet and pay only for that they use. Salesforce is one of the pioneers in providing this service model. Microsoft's Live Mesh also allows sharing files and folders across multiple devices simultaneously.

Although the CC architecture can be divided into four layers as shown in Figure 2, it does not mean that the top layer must be built on the layer directly below it. For example, the SaaS

application can be deployed directly on IaaS, instead of PaaS. Also, some services can be considered as a part of more than one layer. For example, data storage service can be viewed as either in IaaS or PaaS. Given this architectural model, the users can use the services flexibly and efficiently.

### **Advantages of mobile cloud computing**

Cloud computing is known to be a promising solution for MC because of many reasons (e.g., mobility, communication, and portability ). In the following, we describe how the cloud can be used to overcome obstacles in MC, thereby pointing out advantages of MCC.

1. ***Extending battery lifetime.*** Battery is one of the main concerns for mobile devices. Several solutions have been proposed to enhance the CPU performance and to manage the disk and screen in an intelligent manner to reduce power consumption. However, these solutions require changes in the structure of mobile devices, or they require a new hardware that results in an increase of cost and may not be feasible for all mobile devices. Computation offloading technique is proposed with the objective to migrate the large computations and complex processing from resource-limited devices (i.e., mobile devices) to resourceful machines (i.e., servers in clouds). This avoids taking a long application execution time on mobile devices which results in large amount of power consumption.
2. ***Improving data storage capacity and processing power.*** Storage capacity is also a constraint for mobile devices. MCC is developed to enable mobile users to store/access the large data on the cloud through wireless networks. First example is the Amazon Simple Storage Service which supports file storage service. Another example is Image Exchange which utilizes the large storage space in clouds for mobile users. This mobile photo sharing service enables mobile users to upload images to the clouds immediately after capturing. Users may access all images from any devices. With the cloud, the users can save considerable amount of energy and storage space on their mobile devices because all images are sent and processed on the clouds. Mobile cloud computing also helps in reducing the running cost for compute-intensive applications that take long time and large amount of energy when

performed on the limited-resource devices. CC can efficiently support various tasks for data warehousing, managing and synchronizing multiple documents online. For example, clouds can be used for transcoding, playing chess, or broadcasting multimedia services to mobile devices. In these cases, all the complex calculations for transcoding or offering an optimal chess move that take a long time when performed on mobile devices will be processed efficiently on the cloud. Mobile applications also are not constrained by storage capacity on the devices because their data now is stored on the cloud.

3. ***Improving reliability.*** Storing data or running applications on clouds is an effective way to improve the reliability because the data and application are stored and backed up on a number of computers. This reduces the chance of data and application lost on the mobile devices. In addition, MCC can be designed as a comprehensive data security model for both service providers and users. For example, the cloud can be used to protect copyrighted digital contents (e.g., video, clip, and music) from being abused and unauthorized distribution. Also, the cloud can remotely provide to mobile users with security services such as virus scanning, malicious code detection, and authentication. Also, such cloud-based security services can make efficient use of the collected record from different users to improve the effectiveness of the services.

In addition, MCC also inherits some advantages of clouds for mobile services as follows:

- ***Dynamic provisioning.*** Dynamic on-demand provisioning of resources on a fine-grained, self-service basis is a flexible way for service providers and mobile users to run their applications without advanced reservation of resources.
- ***Scalability.*** The deployment of mobile applications can be performed and scaled to meet the unpredictable user demands due to flexible resource provisioning. Service providers can easily add and expand an application and service without or with little constraint on the resource usage.
- ***Multitenancy.*** Service providers (e.g., network operator and data center owner) can share the resources and costs to support a variety of applications and large number of users.
- ***Ease of integration.*** Multiple services from different service providers can be integrated easily through the cloud and Internet to meet the user demand.

## APPLICATIONS OF MOBILE CLOUD COMPUTING

Mobile applications gain increasing share in a global mobile market. Various mobile applications have taken the advantages of MCC. In this section, some typical MCC applications are introduced.

=====

**Q8) Explain the advantages & limitations of cloud computing.(6M)(W-18)**

There is no doubt that businesses can reap huge benefits from cloud computing. However, with the many advantages, come some drawbacks as well. Take time to understand the advantages and disadvantages of cloud computing, so that you can get the most out of your business technology, whichever cloud provider you choose.

**Advantages of Cloud Computing**

**Cost Savings**

Perhaps, the most significant cloud computing benefit is in terms of IT cost savings. Businesses, no matter what their type or size, exist to earn money while keeping capital and operational expenses to a minimum. With cloud computing, you can save substantial capital costs with zero in-house server storage and application requirements. The lack of on-premises infrastructure also removes their associated operational costs in the form of power, air conditioning and administration costs. You pay for what is used and disengage whenever you like - there is no invested IT capital to worry about. It's a common misconception that only large businesses can afford to use the cloud, when in fact, cloud services are extremely affordable for smaller businesses.

**Reliability**

With a managed service platform, cloud computing is much more reliable and consistent than in-house IT infrastructure. Most providers offer a Service Level Agreement which guarantees 24/7/365 and 99.99% availability. Your organization can benefit from a massive pool of redundant IT resources, as well as quick failover mechanism - if a server fails, hosted applications and services can easily be transited to any of the available servers.

**Manageability**

Cloud computing provides enhanced and simplified IT management and maintenance capabilities through central administration of resources, vendor managed infrastructure and SLA backed agreements. IT infrastructure updates and maintenance are eliminated, as all resources are maintained by the service provider. You enjoy a simple web-based user interface for accessing software, applications and services – without the need for installation - and an SLA ensures the timely and guaranteed delivery, management and maintenance of your IT services.

**Strategic Edge**



Ever-increasing computing resources give you a competitive edge over competitors, as the time you require for IT procurement is virtually nil. Your company can deploy mission critical applications that deliver significant business benefits, without any upfront costs and minimal provisioning time. Cloud computing allows you to forget about technology and focus on your key business activities and objectives. It can also help you to reduce the time needed to market newer applications and services.

## **Disadvantages of Cloud Computing**

### **Downtime**

As cloud service providers take care of a number of clients each day, they can become overwhelmed and may even come up against technical outages. This can lead to your business processes being temporarily suspended. Additionally, if your internet connection is offline, you will not be able to access any of your applications, server or data from the cloud.

### **Security**

Although cloud service providers implement the best security standards and industry certifications, storing data and important files on external service providers always opens up risks. Using cloud-powered technologies means you need to provide your service provider with access to important business data. Meanwhile, being a public service opens up cloud service providers to security challenges on a routine basis. The ease in procuring and accessing cloud services can also give nefarious users the ability to scan, identify and exploit loopholes and vulnerabilities within a system. For instance, in a multi-tenant cloud architecture where multiple users are hosted on the same server, a hacker might try to break into the data of other users hosted and stored on the same server. However, such exploits and loopholes are not likely to surface, and the likelihood of a compromise is not great.

### **Vendor Lock-In**

Although cloud service providers promise that the cloud will be flexible to use and integrate, switching cloud services is something that hasn't yet completely evolved. Organizations may find it difficult to migrate their services from one vendor to another. Hosting and integrating current cloud applications on another platform may throw up interoperability and support issues. For instance, applications developed on Microsoft Development Framework (.Net) might not work properly on the Linux platform.

### **Limited Control**

Since the cloud infrastructure is entirely owned, managed and monitored by the service provider, it transfers minimal control over to the customer. The customer can only control and manage the applications, data and services operated on top of that, not the backend infrastructure itself. Key administrative tasks such as server shell access, updating and firmware management may not be passed to the customer or end user.

---

---



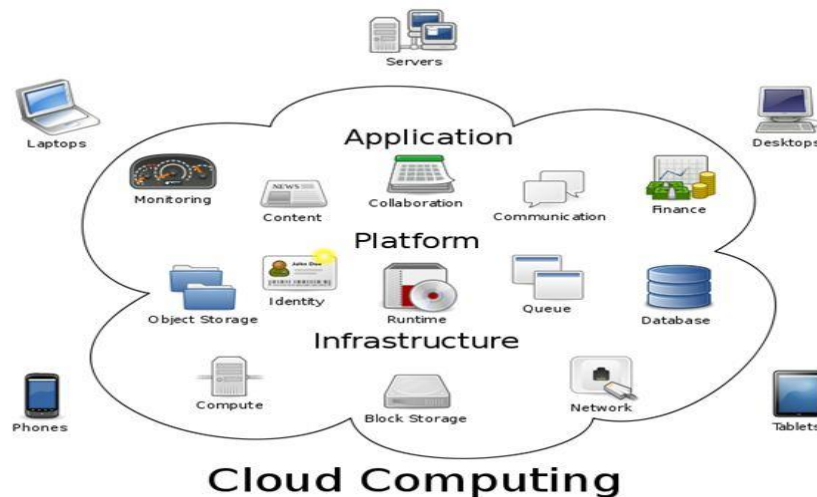
**Tulsiramji Gaikwad-Patil College of Engineering and Technology**  
 Wardha Road, Nagpur-441 108  
 NAAC Accredited

*Department of Computer Science & Engineering*

**Semester: B.E. Eighth Semester (CBS)**  
**Subject: Clustering & Cloud Computing**  
**Unit-2 Solution**

**Q.1) Explain cloud computing architecture.(6M)(S-17)(W-18)**

Ans:



When talking about a cloud computing system, it's helpful to divide it into two sections: the front end and the back end. They connect to each other through a network, usually the Internet. The front end is the side the computer user, or client, sees. The back end is the "cloud" section of the system.

The front end includes the client's computer (or computer network) and the application required to access the cloud computing system. Not all cloud computing systems have the same user interface. Services like Web-based e-mail programs leverage existing Web browsers like Internet Explorer or Firefox. Other systems have unique applications that provide network access to clients.

On the back end of the system are the various computers, servers and data storage systems that create the "cloud" of computing services. In theory, a cloud computing system could include practically any computer program you can imagine, from data processing to video games. Usually, each application will have its own dedicated server.

A central server administers the system, monitoring traffic and client demands to ensure everything runs smoothly. It follows a set of rules called protocols and uses a special kind of software called middleware. Middleware allows networked computers to communicate with each other. Most of the

time, servers don't run at full capacity. That means there's unused processing power going to waste. It's possible to fool a physical server into thinking it's actually multiple servers, each running with its own independent operating system. The technique is called server virtualization. By maximizing the output of individual servers, server virtualization reduces the need for more physical machines.

If a cloud computing company has a lot of clients, there's likely to be a high demand for a lot of storage space. Some companies require hundreds of digital storage devices. Cloud computing systems need at least twice the number of storage devices it requires to keep all its clients' information stored. That's because these devices, like all computers, occasionally break down. A cloud computing system must make a copy of all its clients' information and store it on other devices. The copies enable the central server to access backup machines to retrieve data that otherwise would be unreachable. Making copies of data as a backup is called redundancy.

=====

## 2) What are the role of web services in cloud computing?(7M)(S-17)

**Ans:**

Web services have a great role to play while considering cloud computing services. Web services can be used greatly while considering service oriented architecture within your system. Such architecture can feature free articles, services, and product listings simultaneously. Experts opine that, similar resources can be used to develop a service-oriented architecture using Web Services and Cloud Computing.

Web service based platform can help in preparing organizations towards moving to a different kind of IT platform, on the whole. Thus, web services can be used towards creating a service oriented architecture.

Major system components associated with web based cloud computing services

As far as the new kind of web based cloud computing services, are concerned, auto scaling and elastic load balance are the major drivers in hosting the same within particular system architecture. Web services can help in auto scaling in order to enable a dynamic collection of computing resources. This can simultaneously be used for hosting a particular software application within your system. This actually indicates that; the number of service instances can be dynamically adapted to the volume of data which have been requested for by the end-user.

On the other hand, the request for incoming data application traffic can be balanced using elastic load which can in turn accommodate the service requests. This improves the number of service instances till a large extent. At the same time, this can create a maximum benefit for the end-user.

Experts opine that, web services can actually help in understanding the features of user experience in cloud services. Web services can actually enable an innovative user experience which can be based on service hosting mechanisms as such. This in turn can create a service redeployment method for the cloud computing service operators.

### **Advantages with web-based cloud computing services:**

There are certain clear cut advantages of cloud computing services which are created with the help of web-enabled services. Web-enabled services can help in improving auto scaling techniques by launching the best set of service instances. Such services instances and its impact are measured

according to the distributions of end users in a holistic aspect.

At the same time, web enabled services encourage cloud computing system to extend the elastic load balance. Web enabled services hold a major advantage in order to direct user request to the lightest data load covered through a single service instance.

At the same time, web-enabled services can direct the user request to a nearby server in case of data congestion. This helps in a better protection of data in the long run.

Cloud based computing systems contains several data centers which are hosted in servers connected to the main system. Web services enable the end user to normally connect to the cloud computing system in order to get data-run applications.

=====

**Q.3) List and explain three service model of cloud computing.(7M)(W-17)(S-17)(W-16)(W-18)**

**Ans: Following are the three service model of cloud computing:**

**1. Software as a Service (SaaS)**

**2. Platform as a Service (PaaS)**

**3. Infrastructure as a Service (IaaS)**

**1)Software as a Service (SaaS):**

The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure<sup>2</sup>. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

**2)Platform as a Service (PaaS):**

The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.<sup>3</sup> The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

**3)Infrastructure as a Service (IaaS):**

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

=====

**Q.4) Explain DAAS and NAAS?(7M)(S-17)**

Ans:

### **DaaS – Data as a Service :**

Since you can get software as a service it seems reasonable to think you should be able to get data as a service as well. DaaS providers collect and make available data on a wide range of topics, from economics and finance to social media to climate science. Some DaaS providers offer application programming interfaces (APIs) can provide on demand access to data when bulk downloads are not sufficient.

### **Network as a Service (Naas):**

Network as a Service (NaaS) is sometimes listed as a separate Cloud provider along with Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). This factors out networking, firewalls, related security, etc. from IaaS.

NaaS can include flexible and extended Virtual Private Network (VPN), bandwidth on demand, custom routing, multicast protocols, security firewall, intrusions detection and prevention, Wide Area Network (WAN), content monitoring and filtering, and antivirus. There is no standard specification as to what is included in NaaS. Implementations vary.

---

## **W.5) Explain deployment models of cloud in detail.(7M)(W-17)**

Cloud computing is defined with several deployment models, each of which has specific trade-offs for agencies that are migrating services and operations to cloud-based environments. Because of the different characteristics and trade-offs of the various cloud computing deployment models, it is important the agency IT professionals have a clear understanding of their agency's specific needs as well as how the various systems can help them meet these needs. NIST's official definition for cloud computing outlines four cloud deployment models: private, community, public, and hybrid. Let's take a look at some of the key differences.

### **Private Cloud**

A private cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises

In general, federal agencies and departments opt for private clouds when sensitive or mission-critical information are involved. The private cloud allows for increased security, reliability, performance, and service. Yet, like other types of clouds, it maintains the ability to scale quickly and only pay for what is used when provided by a third party, making it economical as well.

One example of a private cloud deployment model that has been implemented in the federal government relatively recently was implemented by the Los Alamos National Laboratory, which allows researchers to access and utilize servers on demand.

### Community Cloud

The Community Cloud is a type of cloud hosting in which the setup is mutually shared between many organizations that belong to a particular community, i.e. banks and trading firms. It is a multi-tenant setup that is shared among several organizations that belong to a specific group which has similar computing apprehensions. The community members generally share similar privacy, performance and security concerns. The main intention of these communities is to achieve their business-related objectives. A community cloud may be internally managed or it can be managed by a third-party provider. It can be hosted externally or internally. The cost is shared by the specific organizations within the community, hence, community cloud has cost saving capacity. A community cloud is appropriate for organizations and businesses that work on joint ventures, tenders or research that needs a centralized cloud computing ability for managing, building and implementing similar projects.

The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns

The community cloud deployment model is ideal and optimized for agencies or independent organizations that have shared concerns, and therefore need access to shared and mutual records and other types of stored information. Examples might include a community dedicated to compliance considerations or a community focused on security requirements policy.

### Public Cloud

The general public provisions the cloud infrastructure for open use. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider. The public cloud deployment model have the unique advantage of being significantly more secure than accessing information via the Internet and tend to cost less than private clouds because services are more commoditized. Research by the 1105 Government Information Group found that federal agencies interested in public clouds are most commonly interested in the following four functions:

- Collaboration
- Social Networking
- CRM
- Storage

One example of a public cloud deployment model based solution is the Treasury Department, which has moved its website Treasury.gov to a public cloud using Amazon's EC2 cloud service to host the site and its applications. The site includes social media attributes, including Facebook, YouTube and Twitter which allows for rapid and effective communication with constituents.

### Hybrid Cloud

The cloud infrastructure is a composition of two or more distinct cloud deployment models (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

Large portions of agencies that have already switched some processes over to cloud based computing solutions have utilized hybrid cloud options. Few enterprises have the ability to switch over all of their IT services at one time, the hybrid option allows for a mix of on base and cloud options which provide an easier transition.

NASA is one example of a federal agency who is utilizing the Hybrid Cloud Computing deployment model. Its Nebula open-source cloud computing project uses a private cloud for research and development as well as a public cloud to shared datasets with external partners and the public.

The hybrid cloud computing deployment model option has also proven to be the choice option for state and local governments as well, with states like Michigan and Colorado having already declared their cloud computing intentions with plans illustrating hybrid cloud deployment models.

---

**Q.6) What do you mean by virtualization? Explain the pitfalls of virtualization.(6M)(W-17)(w-16)**

### **Definition - What does *Virtualization* mean?**

Virtualization refers to the creation of a virtual resource such as a server, desktop, operating system, file, storage or network.

The main goal of virtualization is to manage workloads by radically transforming traditional computing to make it more scalable. Virtualization has been a part of the IT landscape for decades now, and today it can be applied to a wide range of system layers, including operating system-level virtualization, hardware-level virtualization and server virtualization.

### **Pitfalls**

#### **Mismatching Servers**

This aspect is commonly overlooked especially by smaller companies that don't invest sufficient funds in their IT infrastructure and prefer to build it from several bits and pieces. This usually leads to simultaneous virtualization of servers that come with different chip technology (AMD and Intel). Frequently, migration of virtual machines between them won't be possible and server restarts will be the only solution. This is a major hindrance and actually means losing the benefits of live migration and virtualization.

### **Creating Too many Virtual Machines per Server**

One of the great things about virtual machines is that they can be easily created and migrated from server to server according to needs. However, this can also create problems sometimes because IT staff members may get carried away and deploy more Virtual Machines than a server can handle.

This will actually lead to a loss of performance that can be quite difficult to spot. A practical way to work around this is to have some policies in place regarding VM limitations and to make sure that the employees adhere to them.

### **Misplacing Applications**

A virtualized infrastructure is a more complex than a traditional one and with a number of applications deployed, losing track of applications is a distinct possibility. Within a physical server infrastructure keeping track of all the apps and the machines running them isn't a difficult task. However, once you add a significant number of virtual machines to the equation, things can get messy and App patching, software licensing and updating can turn into painfully long processes.

=====

## **Q.7) Write in brief about virtualization applications in enterprises. (6M)(W-17)**

### **Application Virtualization Features & Capabilities**

Among the most important application virtualization features are:

- Support for a wide range of applications and application types
- Capable of delivering to a wide variety of endpoints with few restrictions such as driver management, etc.
- Ease of deployment
- Ease of packaging applications into a single executable
- Access control through authentication, IP address etc.

The concept of virtualization generally refers to separating the logical from the physical, and that is at the heart of application virtualization too. The advantages of this approach to accessing application software are that any incompatibility problems between the local machine's operating system and the application are irrelevant; The user's machine is not actually using its own operating system.

### **Advantages of Application Virtualization**



Application virtualization, by decoupling the applications from the hardware on which they run has many advantages. One advantage is maintaining a standard cost-effective operating system configuration across multiple machines by isolating applications from their local operating systems. There are additional cost advantages like saving on license costs, and greatly reducing the need for support services to maintain a healthy computing environment.

---

### **Q.8) Explain infrastructure as a service (IaaS) using openstack/ owncloud. (6M)(W-16)**

#### **Infrastructure as-a-service (IaaS)**

IaaS includes the delivery of computing infrastructure such as a virtual machine, disk image library, raw block storage, object storage, firewalls, load balancers, IP addresses, virtual local area networks and other features on-demand from a large pool of resources installed in data centres. Cloud providers bill for the IaaS services on a utility computing basis; the cost is based on the amount of resources allocated and consumed.

#### **OpenStack: a free and open source cloud computing platform**

OpenStack is a free and open source, cloud computing software platform that is widely used in the deployment of infrastructure-as-a-Service (IaaS) solutions. The core technology with OpenStack comprises a set of interrelated projects that control the overall layers of processing, storage and networking resources through a data centre that is managed by the users using a Web-based dashboard, command-line tools, or by using the RESTful API. Currently, OpenStack is maintained by the OpenStack Foundation, which is a non-profit corporate organisation established in September 2012 to promote OpenStack software as well as its community. Many corporate giants have joined the project, including GoDaddy, Hewlett Packard, IBM, Intel, Mellanox, Mirantis, NEC, NetApp, Nexenta, Oracle, Red Hat, SUSE Linux, VMware, Arista Networks, AT&T, AMD, Avaya, Canonical, Cisco, Dell, EMC, Ericsson, Yahoo!, etc.

#### **OpenStack computing components**

OpenStack has a modular architecture that controls large pools of compute, storage and networking resources.

**Compute (Nova):** OpenStack Compute (Nova) is the fabric controller, a major component of Infrastructure as a Service (IaaS), and has been developed to manage and automate pools of computer resources. It works in association with a range of virtualisation technologies. It is written in Python and uses many external libraries such as Eventlet, Kombu and SQLAlchemy.

**Object storage (Swift):** It is a scalable redundant storage system, using which objects and files are placed on multiple disks throughout servers in the data centre, with the OpenStack

software responsible for ensuring data replication and integrity across the cluster. OpenStack Swift replicates the content from other active nodes to new locations in the cluster in case of server or disk failure.

**Block storage (Cinder):** OpenStack block storage (Cinder) is used to incorporate continual block-level storage devices for usage with OpenStack compute instances. The block storage system of OpenStack is used to manage the creation, mounting and unmounting of the block devices to servers. Block storage is integrated for performance-aware scenarios including database storage, expandable file systems or providing a server with access to raw block level storage. Snapshot management in OpenStack provides the authoritative functions and modules for the back-up of data on block storage volumes. The snapshots can be restored and used again to create a new block storage volume.

**Networking (Neutron):** Formerly known as Quantum, Neutron is a specialised component of OpenStack for managing networks as well as network IP addresses. OpenStack networking makes sure that the network does not face bottlenecks or any complexity issues in cloud deployment. It provides the users continuous self-service capabilities in the network's infrastructure. The floating IP addresses allow traffic to be dynamically routed again to any resources in the IT infrastructure, and therefore the users can redirect traffic during maintenance or in case of any failure. Cloud users can create their own networks and control traffic along with the connection of servers and devices to one or more networks. With this component, OpenStack delivers the extension framework that can be implemented for managing additional network services including intrusion detection systems (IDS), load balancing, firewalls, virtual private networks (VPN) and many others.

---

**Q.9) Explain the role of Networks in cloud computing and also define the various protocols used in it.(7M)(W-16)**

---

The concept of flexible sharing for more efficient use of hardware resources is nothing new in enterprise networking—but cloud computing is different. For some, the cloud computing trend sounds nebulous, but it's not so confusing when you view it from the perspective of IT professionals. For them, it is a way to quickly increase capacity without investing in new infrastructure, training more people, or licensing additional software.

Decoupling services from hardware poses a key question that must be addressed as enterprises consider cloud computing's intriguing possibilities: What data center interconnect protocol is best suited for linking servers and storage in and among cloud centers? Enterprise IT managers and carrier service planners must consider the benefits and limitations conveyed by a host of technologies—Fibre Channel over Ethernet (FCoE), InfiniBand, and 8-Gbps Fibre Channel—when enabling the many virtual machines that compose the cloud.

Understanding the cloud craze

The idea of being able to flexibly and cost-effectively mix and match hardware resources to adapt easily to new needs and opportunities is extremely enticing to the enterprise that has tried to ride the waves of constant IT change. Higher-speed connectivity, higher-density computing, e-commerce, Web 2.0, mobility, business continuity/disaster recovery capabilities... the need for a more flexible IT environment that is built for inevitable,

incessant change has become obvious and creates an enterprise marketplace that believes in the possibilities of cloud computing.

Cloud computing has successfully enabled Internet search engines, social media sites, and, more recently, traditional business services (for example, Google Docs and Salesforce.com). Today, enterprises can implement their own private cloud environment via end-to-end vendor offerings or contract for public desktop services, in which applications and data are accessed from network-attached devices. Both the private and public cloud-computing approaches promise significant reductions in capital and operating expenditures (capex and opex). The capex savings arise through more efficient use of servers and storage. Opex improvements derive from the automated, integrated management of data-center infrastructure.

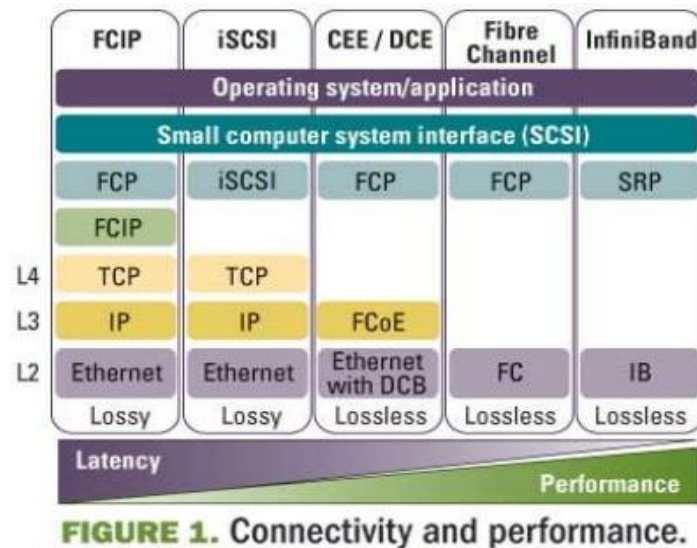
#### Ramifications for the data center

One of the most dramatic changes that cloud computing brings to the data center is in the interconnection of servers and storage.

Links among server resources traditionally were lower bandwidth, which was allowable given that few virtual machines were in use. The data center has been populated mostly with lightly utilized, application-dedicated, x86-architecture servers running one bare-metal operating system or multiple operating systems via hypervisor.

In the dynamic model emerging today, many more virtual machines are created through the clustering of highly utilized servers. Large and small businesses that use this type of service will want the ability to place “instances” in multiple locations and dynamically move them around. These are distinct locations that are engineered to be insulated from failures elsewhere. This desire leads to terrific scrutiny on the protocols used to interconnect servers and storage among these locations.

Bandwidth and latency requirements vary depending on the particular cloud application. A latency of 50 ms or more, for example, might be tolerable for the emergent public desktop service. Ultralow latency, near 1 ms, is needed for some high-end services such as grid computing or synchronous backup. Ensuring that each application receives its necessary performance characteristics over required distances is a prerequisite to cloud success.



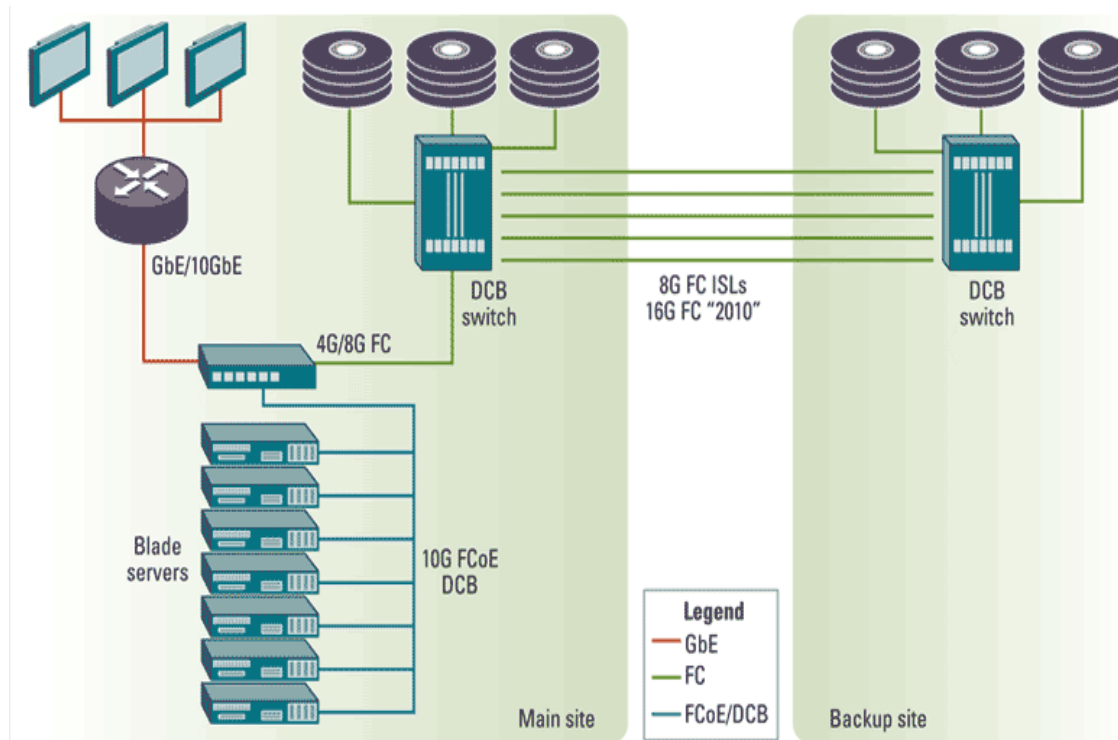
At the same time, enterprises have long sought to cost-effectively collapse LAN and SAN traffic onto a single interconnection fabric with virtualization. While IT managers cannot sacrifice the performance requirements of mission-critical applications, they also must seek to cut cloud-bandwidth costs. Some form of Ethernet and InfiniBand are most likely to eventually serve as that single, unifying interconnect; both are built to keep pace with Moore's Law, which shows the amount of data doubling every year with no end in sight. In fact, neither of these protocols, nor Fibre Channel, figures to exit the cloud center soon (see Fig. 1).

#### FCoE

FCoE/Data Center Bridging (DCB)—which proposes to converge Fibre Channel and Ethernet, the two most prevalent enterprise-networking protocols—is generating a great deal of interest. Its primary value is I/O consolidation, aggregating and distributing data traffic to existing LAN and SAN equipment from atop server racks. FCoE/DCB promises low latency and plenty of bandwidth (10 to 40 Gbps), but the emergent protocol is unproven in large-scale deployments (see Fig. 2).

Also, there are significant problems related to pathing, routing, and distance support. It is essential that the FCoE approach supports the Ethernet and IP standards, alongside Fibre Channel standards for switching, path selection, and routing.

Basically, there are issues to be overcome if we are to have a truly lossless enhanced Ethernet that delivers link-level shortest-path-first routing over distance. The aim is to ensure zero loss due to congestion in and between data centers.



**FIGURE 2.** FCoE/DCB for I/O consolidation and offsite backup.

The absence of standards defining inter-switch links (ISLs) among distributed FCoE switches, a lack of multihop support, and a shortage of native FCoE interfaces on storage equipment must be addressed. FCoE must prove itself in areas such as latency, synchronous recovery, and continuous availability over distances before it gains much of a role in the most demanding cloud-computing applications.

This means that 8G, 10G, and eventually 16G Fibre Channel ISLs will be required to back up FCoE blade servers and storage for many years to come.

#### InfiniBand

The number of InfiniBand-connected central processing unit (CPU) cores on the Top 500 list grew 63.4% last year, from 859,090 in November 2008 to 1,404,164 in November 2009. Ethernet-connected systems declined 8% in the same period.

InfiniBand is frequently the choice for the most demanding applications. For example, it's the protocol interconnecting remotely located data centers for IBM's Geographically Dispersed Parallel Sysplex (GDPS) PSIFB business-continuity and disaster-recovery offering (see Fig. 3).

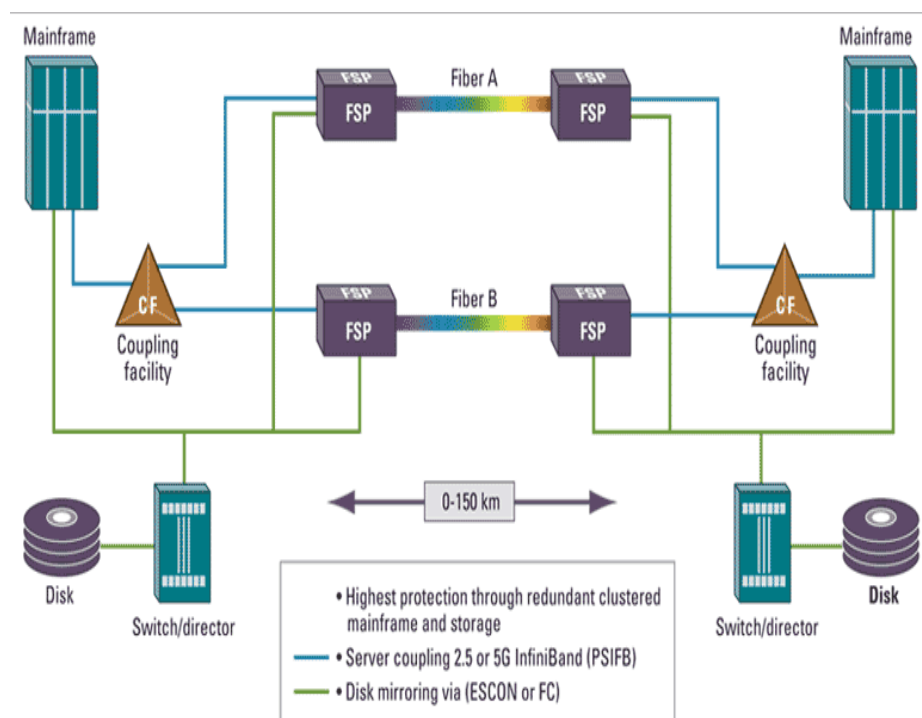
The highest-performance, transaction-sensitive business services that might be offloaded to the cloud demand the unmatched combination of bandwidth (up to 40 Gbps) and latency (as little as 1  $\mu$ s) that an InfiniBand port delivers. Only 10 Gbps of bandwidth and TCP protocol latency of at least 6  $\mu$ s are possible with an Ethernet port. Latencies as great as 40 to 50  $\mu$ s are possible, as several tiers of hierarchical switching are often necessary in networks interconnected via Ethernet ports to overcome oversubscription. Finally, InfiniBand, like Fibre Channel, will not drop packets like traditional Ethernet. 8G Fibre Channel 8G Fibre Channel, transported at native speed via DWDM, dominates for rapid backup and recovery

SAN services that must not experience degradation over distance. Early in 2009, in fact, COLT announced an 8G Fibre Channel storage service deployment including fiber spans of more than 135 km. Some carriers see 8G Fibre Channel as a dependable enabler for public cloud-computing services to high-end Fortune 500 customers today, and certainly it's a protocol that must be supported in cloud centers moving forward. For this reason, 16G Fibre Channel will be welcomed as a way to potentially bridge and back up 10G FCoE blade servers over distance.

### Other considerations

Latency and distance aren't the only factors in determining which interconnect protocols will enable cloud computing. Protocol maturity and dependability also figure into the decision.

With mission-critical applications being entrusted to clustered virtual machines, the stakes are high. Enterprise IT managers and carrier service planners are likely to employ proven implementations of InfiniBand, 8G Fibre Channel, and potentially FCoE and/or some other form of Ethernet in cloud centers for some years because they trust them. Low-latency grid computing—dependably enabled by proven 40G InfiniBand—is a perfect example of an application with particular performance requirements that must not be sacrificed for the sake of one-size-fits-all convergence.



**FIGURE 3.** IBM's GDPS PSIFB for mainframe clustering.

This is one of the chief reasons that real-world cloud centers likely will remain multiprotocol environments—despite the desire to converge on a single interconnect fabric for benefits of cost and operational simplicity, despite the hype around promising, but emergent, protocols such as FCoE/DCB.

Other issues, such as organization and behavior, cannot be ignored. Collapsing an enterprise networking group's LAN traffic and storage group's SAN traffic on the same protocol would entail significant political and technical ramifications. Convergence on a single fabric, however attractive in theory, implies nothing less than an organizational transformation in addition to a significant forklift upgrade to new enhanced (low-latency) DCB Ethernet switches.

### **Maintaining flexibility**

Given this host of factors, cloud operators should be prepared to support multiple interconnect protocols with the unifying role shouldered by DWDM, delivering protocol-agnostic, native-speed, low-latency transport across the cloud over fiber spans up to 600 km long. Today's services can be commonly deployed and managed across existing optical networks via DWDM, and operators retain the flexibility to elegantly bring on new services and protocols as needed.

The unprecedented cost efficiencies and capabilities offered by cloud computing have garnered attention from large and small enterprises across industries. When interconnecting servers to enable a cloud's virtual machines, enterprise IT managers and carrier service planners must take care to ensure that the varied performance requirements of all LAN and SAN services are reliably met.

---

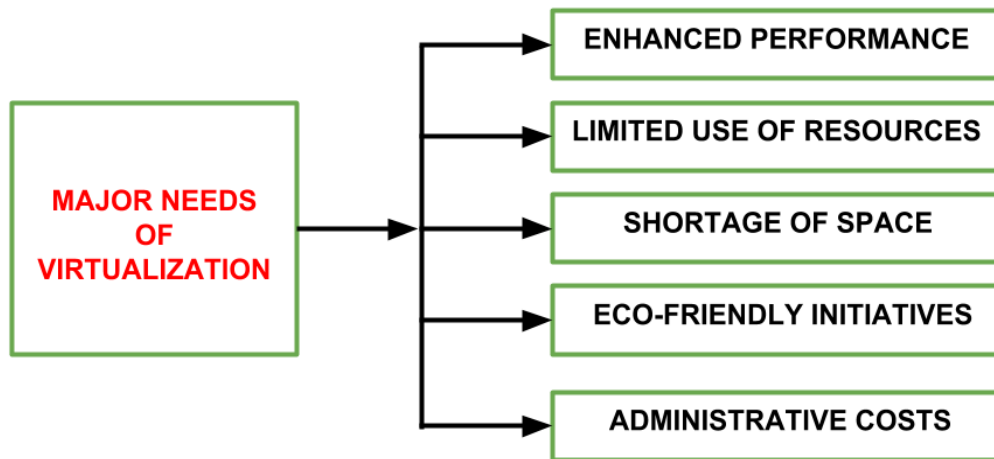
### **Q.10) Define virtualization. What is the need of virtualization in cloud computing?(6M)(W-18)**

#### **What does *Virtualization* mean?**

Virtualization refers to the creation of a virtual resource such as a server, desktop, operating system, file, storage or network.

The main goal of virtualization is to manage workloads by radically transforming traditional computing to make it more scalable. Virtualization has been a part of the IT landscape for decades now, and today it can be applied to a wide range of system layers, including operating system-level virtualization, hardware-level virtualization and server virtualization.

There are five major needs of virtualization which are described below:



*Figure: Major needs of Virtualization.*

### **1. ENHANCED PERFORMANCE-**

Currently, the end user system i.e. PC is sufficiently powerful to fulfill all the basic computation requirements of the user, with various additional capabilities which are rarely used by the user. Most of their systems have sufficient resources which can host a virtual machine manager and can perform a virtual machine with acceptable performance so far.

### **2. LIMITED USE OF HARDWARE AND SOFTWARE RESOURCES-**

The limited use of the resources leads to under-utilization of hardware and software resources. As all the PCs of the user are sufficiently capable to fulfill their regular computational needs that's why many of their computers are used often which can be used 24/7 continuously without any interruption. The efficiency of IT infrastructure could be increase by using these resources after hours for other purposes. This environment is possible to attain with the help of Virtualization.

### **3. SHORTAGE OF SPACE-**

The regular requirement for additional capacity, whether memory storage or compute power, leads data centers raise rapidly. Companies like Google, Microsoft and Amazon develop their infrastructure by building data centers as per their needs. Mostly, enterprises unable to pay to build any other data center to accommodate additional resource capacity. This heads to the diffusion of a technique which is known as server consolidation.

### **4. ECO-FRIENDLY INITIATIVES-**

At this time, corporations are actively seeking for various methods to minimize their expenditures on power which is consumed by their systems. Data centers are main power consumers and maintaining a data center operations needs a continuous power supply as well as a good amount of energy is needed to keep them cool for well-functioning. Therefore,



server consolidation drops the power consumed and cooling impact by having a fall in number of servers. Virtualization can provide a sophisticated method of **server consolidation**.

## **5. ADMINISTRATIVE COSTS-**

Furthermore, the rise in demand for capacity surplus, that convert into more servers in a data center, accountable for a significant increase in administrative costs. Hardware monitoring, server setup and updates, defective hardware replacement, server resources monitoring, and backups are included in common system administration tasks. These are personnel-intensive operations. The administrative costs is increased as per the number of servers. Virtualization decreases number of required servers for a given workload, hence reduces the cost of administrative employees.

---

### **Q.11) List & explain the advantages & limitations of different deployment models in cloud computing.(4M)(W-18)**

#### **Private Cloud**

A private cloud is cloud infrastructure that only members of your organization can utilize. It is typically owned and managed by the organization itself and is hosted on premises but it could also be managed by a third party in a secure datacenter. This deployment model is best suited for organizations that deal with sensitive data and/or are required to uphold certain security standards by various regulations.

#### **Advantages:**

- Organization specific
- High degree of security and level of control
- Ability to choose your resources (ie. specialized hardware)

#### **Disadvantages:**

- Lack of elasticity and capacity to scale (bursts)
- Higher cost
- Requires a significant amount of engineering effort

#### **Public Cloud**

Public cloud refers to cloud infrastructure that is located and accessed over the public network. It provides a convenient way to burst and scale your project depending on the use and is typically pay-per-use. Popular examples include Amazon AWS, Google Cloud Platform and Microsoft Azure.

#### **Advantages:**

- Scalability/Flexibility/Bursting
- Cost effective
- Ease of use

**Disadvantages:**

- Shared resources
- Operated by third party
- Unreliability
- Less secure

**Hybrid Cloud**

This type of cloud infrastructure assumes that you are hosting your system both on private and public cloud. One use case might be regulation requiring data to be stored in a locked down private data center but have the application processing parts available on the public cloud and talking to the private components over a secure tunnel.

Another example is hosting most of the system inside a private cloud and having a clone of the system on the public cloud to allow for rapid scaling and accommodating bursts of new usage that would otherwise not be possible on the private cloud.

**Advantages:**

- Cost effective
- Scalability/Flexibility
- Balance of convenience and security

**Disadvantages:**

- Same disadvantages as the public cloud
- 



**Tulsiramji Gaikwad-Patil College of Engineering and Technology**  
Wardha Road, Nagpur-441 108  
NAAC Accredited

---

*Department of Computer Science & Engineering*

---

**Semester: B.E. Eighth Semester (CBS)**  
**Subject: Clustering & Cloud Computing**  
**Unit-3 Solution**

---

**Q1) Explain infrastructure as a service (IaaS) using openstack/ owncloud.(6M)(W-16)****Infrastructure as-a-service (IaaS)**

IaaS includes the delivery of computing infrastructure such as a virtual machine, disk image library, raw block storage, object storage, firewalls, load balancers, IP addresses, virtual local area networks and other features on-demand from a large pool of resources installed in data centres. Cloud providers bill for the IaaS services on a utility computing basis; the cost is based on the amount of resources allocated and consumed.

**OpenStack: a free and open source cloud computing platform**

OpenStack is a free and open source, cloud computing software platform that is widely used in the deployment of infrastructure-as-a-Service (IaaS) solutions. The core technology with OpenStack comprises a set of interrelated projects that control the overall layers of processing, storage and networking resources through a data centre that is managed by the users using a Web-based dashboard, command-line tools, or by using the RESTful API. Currently, OpenStack is maintained by the OpenStack Foundation, which is a non-profit corporate organisation established in September 2012 to promote OpenStack software as well as its community. Many corporate giants have joined the project, including GoDaddy, Hewlett Packard, IBM, Intel, Mellanox, Mirantis, NEC, NetApp, Nexenta, Oracle, Red Hat, SUSE Linux, VMware, Arista Networks, AT&T, AMD, Avaya, Canonical, Cisco, Dell, EMC, Ericsson, Yahoo!, etc.

**OpenStack computing components**

OpenStack has a modular architecture that controls large pools of compute, storage and networking resources.

**Compute (Nova):** OpenStack Compute (Nova) is the fabric controller, a major component of Infrastructure as a Service (IaaS), and has been developed to manage and automate pools of computer resources. It works in association with a range of virtualisation technologies. It is written in Python and uses many external libraries such as Eventlet, Kombu and SQLAlchemy.

**Object storage (Swift):** It is a scalable redundant storage system, using which objects and files are placed on multiple disks throughout servers in the data centre, with the OpenStack software responsible for ensuring data replication and integrity across the cluster. OpenStack Swift replicates the content from other active nodes to new locations in the cluster in case of server or disk failure.

**Block storage (Cinder):** OpenStack block storage (Cinder) is used to incorporate continual block-level storage devices for usage with OpenStack compute instances. The block storage system of OpenStack is used to manage the creation, mounting and unmounting of the block devices to servers. Block storage is integrated for performance-aware scenarios including database storage, expandable file systems or providing a server with access to raw block level storage. Snapshot management in OpenStack provides the authoritative functions and modules for the back-up of

data on block storage volumes. The snapshots can be restored and used again to create a new block storage volume.

**Networking (Neutron):** Formerly known as Quantum, Neutron is a specialised component of OpenStack for managing networks as well as network IP addresses. OpenStack networking makes sure that the network does not face bottlenecks or any complexity issues in cloud deployment. It provides the users continuous self-service capabilities in the network's infrastructure. The floating IP addresses allow traffic to be dynamically routed again to any resources in the IT infrastructure, and therefore the users can redirect traffic during maintenance or in case of any failure. Cloud users can create their own networks and control traffic along with the connection of servers and devices to one or more networks. With this component, OpenStack delivers the extension framework that can be implemented for managing additional network services including intrusion detection systems (IDS), load balancing, firewalls, virtual private networks (VPN) and many others.

---

## **Q2) Explain the clustering Big data and also gives classification of Big data.(7M)(W-16)**

Clustering is an essential data mining and tool for analyzing big data. There are difficulties for applying clustering techniques to big data due to new challenges that are raised with big data. As Big Data is referring to terabytes and petabytes of data and clustering algorithms are come with high computational costs, the question is how to cope with this problem and how to deploy clustering techniques to big data and get the results in a reasonable time. This study is aimed to review the trend and progress of clustering algorithms to cope with big data challenges from very first proposed algorithms until today's novel solutions. The algorithms and the targeted challenges for producing improved clustering algorithms are introduced and analyzed, and afterward the possible future path for more advanced algorithms is illuminated based on today's available technologies and frameworks.

### **1. Structured data**

Structured Data is used to refer to the data which is already stored in databases, in an ordered manner. It accounts for about 20% of the total existing data, and is used the most in programming and computer-related activities.

There are two sources of structured data- machines and humans. All the data received from sensors, web logs and financial systems are classified under machine-generated data. These include medical devices, GPS data, data of usage statistics captured by servers and applications and the huge amount of data that usually move through trading platforms, to name a few.

Human-generated structured data mainly includes all the data a human input into a computer, such as his name and other personal details. When a person clicks a link on the internet, or even makes a move in a game, data is created- this can be used by companies to figure out their customer behaviour and make the appropriate decisions and modifications.

## 2. Unstructured data

While structured data resides in the traditional row-column databases, unstructured data is the opposite- they have no clear format in storage. The rest of the data created, about 80% of the total account for unstructured big data. Most of the data a person encounters belongs to this category- and until recently, there was not much to do to it except storing it or analysing it manually.

Unstructured data is also classified based on its source, into machine-generated or human-generated. Machine-generated data accounts for all the satellite images, the scientific data from various experiments and radar data captured by various facets of technology.

Human-generated unstructured data is found in abundance across the internet, since it includes social media data, mobile data and website content. This means that the pictures we upload to our Facebook or Instagram handles, the videos we watch on YouTube and even the text messages we send all contribute to the gigantic heap that is unstructured data.

## 3. Semi-structured data.

The line between unstructured data and semi-structured data has always been unclear, since most of the semi-structured data appear to be unstructured at a glance. Information that is not in the traditional database format as structured data, but contain some organizational properties which make it easier to process, are included in semi-structured data. For example, NoSQL documents are considered to be semi-structured, since they contain keywords that can be used to process the document easily.

Big Data analysis has been found to have a definite business value, as its analysis and processing can help a company achieve cost reductions and dramatic growth. So it is imperative that you do not wait too long to exploit the potential of this excellent business opportunity.

---

### 3)List the characteristics of HDFS. And explain HDFS operations.(7M)(W-16)

#### Features of Hadoop HDFS

##### Fault Tolerance

Fault tolerance in **HDFS** refers to the working strength of a system in unfavorable conditions and how that system can handle such situations. HDFS is highly fault-tolerant, in HDFS data is divided into blocks and multiple copies of blocks are created on different machines in the cluster (this replica creation is configurable). So whenever if any machine in the cluster goes down, then a client can easily access their data from the other machine which contains the same copy of data blocks. HDFS also maintains the replication factor by creating a replica of blocks of data on another rack. Hence if suddenly a machine fails, then a user can access data from other slaves present in another rack. To learn more about Fault Tolerance follow this Guide.

## **High Availability**

HDFS is a highly available file system, data gets replicated among the nodes in the HDFS cluster by creating a replica of the blocks on the other slaves present in HDFS cluster. Hence whenever a user wants to access this data, they can access their data from the slaves which contains its blocks and which is available on the nearest node in the cluster. And during unfavorable situations like a failure of a node, a user can easily access their data from the other nodes. Because duplicate copies of blocks which contain user data are created on the other nodes present in the HDFS cluster. To learn more about high availability follow this Guide.

## **Data Reliability**

HDFS is a distributed file system which provides reliable data storage. HDFS can store data in the range of 100s of petabytes. It also stores data reliably on a cluster of nodes. HDFS divides the data into blocks and these blocks are stored on nodes present in HDFS cluster. It stores data reliably by creating a replica of each and every block present on the nodes present in the cluster and hence provides fault tolerance facility. If node containing data goes down, then a user can easily access that data from the other nodes which contain a copy of same data in the HDFS cluster. HDFS by default creates copies of blocks containing data present in the nodes in HDFS cluster. Hence data is quickly available to the users and hence user does not face the problem of data loss. Hence HDFS is highly reliable.

## **Replication**

Data Replication is one of the most important and unique features of Hadoop HDFS. In HDFS replication of data is done to solve the problem of data loss in unfavorable conditions like crashing of a node, hardware failure, and so on. Since data is replicated across a number of machines in the cluster by creating blocks. The process of replication is maintained at regular intervals of time by HDFS and HDFS keeps creating replicas of user data on different machines present in the cluster. Hence whenever any machine in the cluster gets crashed, the user can access their data from other machines which contain the blocks of that data. Hence there is no possibility of losing of user data. Follow this guide to learn more about the data read operation.

## **Scalability**

As HDFS stores data on multiple nodes in the cluster, when requirements increase we can scale the cluster. There is two scalability mechanism available: Vertical scalability – add more resources (CPU, Memory, Disk) on the existing nodes of the cluster. Another way is horizontal scalability – Add more machines in the cluster. The horizontal way is preferred since we can scale the cluster from 10s of nodes to 100s of nodes on the fly without any downtime.

## **Distributed Storage**

In HDFS all the features are achieved via distributed storage and replication. HDFS data is stored in distributed manner across the nodes in HDFS cluster. In HDFS data is divided

into blocks and is stored on the nodes present in HDFS cluster. And then replicas of each and every block are created and stored on other nodes present in the cluster. So if a single machine in the cluster gets crashed we can easily access our data from the other nodes which contain its replica.

### **i. Interaction of Client with NameNode**

If the client has to create a file inside HDFS then he needs to interact with the namenode (as namenode is the centre-piece of the cluster which contains metadata). Namenode provides the address of all the slaves where the client can write its data. The client also gets a security token from the namenode which they need to present to the slaves for authentication before writing the block. Below are the steps which client needs to perform in order to write data in HDFS:

To create a file client executes `create()` method on `DistributedFileSystem`. Now `DistributedFileSystem` interacts with the namenode by making an RPC call for creating a new file having no blocks associated with it in the filesystem's namespace. Various checks are executed by the namenode in order to make sure that there is no such file, already present there and the client is authorized to create a new file.

If all this procedure gets the pass, then a record of the new file is created by the namenode; otherwise, file creation fails and an `IOException` is thrown to the client. An `FSDDataOutputStream` returns by the `DistributedFileSystem` for the client in order to start writing data to datanode. Communication with datanodes and client is handled by `DFSOutputStream` which is a part of `FSDDataOutputStream`.

### **ii. Interaction of client with Datanodes**

After the user gets authenticated to create a new file in the filesystem namespace, Namenode will provide the location to write the blocks. Hence the client directly goes to the datanodes and start writing the data blocks there. As in HDFS replicas of blocks are created on different nodes, hence when the client finishes with writing a block inside the slave, the slave then starts making replicas of a block on the other slaves. And in this way, multiple replicas of a block are created in different blocks. Minimum 3 copies of blocks are created in different slaves and after creating required replicas, it sends an acknowledgment to the client. In this manner, while writing data block a pipeline is created and data is replicated to desired value in the cluster.

Let's understand the procedure in great details. Now when the client writes the data, they are split into the packets by the `DFSOutputStream`. These packets are written to an internal queue, called the data queue. The data queue is taken up by the `DataStreamer`. The main responsibility of `DataStreamer` is to ask the namenode to properly allocate the new blocks on suitable datanodes in order to store the replicas. List of datanodes creates a pipeline, and here let us assume the default replication level is three; hence in the pipeline there are three nodes. The packets are streamed to the first datanode on the pipeline by the `DataStreamer`, and `DataStreamer` stores the packet and this packet is then forwarded to the second datanode in the pipeline.

In the same way, the packet is stored into the second datanode and then it is forwarded to the third (and last) datanode in the pipeline.

An internal queue known as "ack queue" of packets that are waiting to be acknowledged by datanodes is also maintained. A packet is only removed from the ack queue if it gets acknowledged by all the datanodes in the pipeline. A client calls the close() method on the stream when it has finished writing data.

When executing the above method, all the remaining packets get flushed to the datanode pipeline and before contacting the namenode it waits for acknowledgments to signal that the file is complete. This is already known to the namenode that which blocks the file is made up of, and hence before returning successfully it only has to wait for blocks to be minimally replicated.

---

**Q4) Explain Hadoop Map Reduce job execution with the help of neat diagram.(7M)(W-16)(W-18)(w-17)**

---

MapReduce is a processing technique and a program model for distributed computing based on java. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job.

The major advantage of MapReduce is that it is easy to scale data processing over multiple computing nodes. Under the MapReduce model, the data processing primitives are called mappers and reducers. Decomposing a data processing application into *mappers* and *reducers* is sometimes nontrivial. But, once we write an application in the MapReduce form, scaling the application to run over hundreds, thousands, or even tens of thousands of machines in a cluster is merely a configuration change. This simple scalability is what has attracted many programmers to use the MapReduce model.

---

**Q.5) Explain in steps starting & stopping Hadoop cluster.(7M) (W-16)(W-18)(S-17)**

---

## **SINGLE-NODE INSTALLATION**

Running Hadoop on Ubuntu (Single node cluster setup)

The report here will describe the required steps for setting up a single-node Hadoop cluster backed by the Hadoop Distributed File System, running on Ubuntu Linux. Hadoop is a framework written in



Java for running applications on large clusters of commodity hardware and incorporates features similar to those of the Google File System (GFS) and of the MapReduce computing paradigm. Hadoop's HDFS is a highly fault-tolerant distributed file system and, like Hadoop in general, designed to be deployed on low-cost hardware. It provides high throughput access to application data and is suitable for applications that have large data sets.

Before we start, we will understand the meaning of the following:

### **DataNode:**

A DataNode stores data in the Hadoop File System. A functional file system has more than one DataNode, with the data replicated across them.

### **NameNode:**

The NameNode is the centrepiece of an HDFS file system. It keeps the directory of all files in the file system, and tracks where across the cluster the file data is kept. It does not store the data of these file itself.

### **Jobtracker:**

The Jobtracker is the service within hadoop that farms out MapReduce to specific nodes in the cluster, ideally the nodes that have the data, or atleast are in the same rack.

### **TaskTracker:**

A TaskTracker is a node in the cluster that accepts tasks- Map, Reduce and Shuffle operations – from a Job Tracker.

Secondary Namenode:

Secondary Namenode whole purpose is to have a checkpoint in HDFS. It is just a helper node for namenode.

### **Prerequisites:**

Java 6 JDK

Hadoop requires a working Java 1.5+ (aka Java 5) installation.

Update the source list

```
user@ubuntu:~$ sudo apt-get update
```

or

Install Sun Java 6 JDK

If you already have Java JDK installed on your system, then you need not run the above command.

To install it

```
user@ubuntu:~$ sudo apt-get install sun-java6-jdk
```

The full JDK which will be placed in /usr/lib/jvm/java-6-openjdk-amd64 After installation, check whether java JDK is correctly installed or not, with the following command

```
user@ubuntu:~$ java -version
```

Adding a dedicated Hadoop system user

We will use a dedicated Hadoop user account for running Hadoop.

```
user@ubuntu:~$ sudo addgroup hadoop_group
```

```
user@ubuntu:~$ sudo adduser --ingroup hadoop_group hduser1
```

This will add the user hduser1 and the group hadoop\_group to the local machine. Add hduser1 to the sudo group

```
user@ubuntu:~$ sudo adduser hduser1 sudo
```

## Configuring SSH

The hadoop control scripts rely on SSH to perform cluster-wide operations. For example, there is a script for stopping and starting all the daemons in the clusters. To work seamlessly, SSH needs to be setup to allow password-less login for the hadoop user from machines in the cluster. The simplest way to achieve this is to generate a public/private key pair, and it will be shared across the cluster.

Hadoop requires SSH access to manage its nodes, i.e. remote machines plus your local machine. For our single-node setup of Hadoop, we therefore need to configure SSH access to localhost for the hduser user we created in the earlier.

We have to generate an SSH key for the hduser user.

```
user@ubuntu:~$ su - hduser1
```

```
hduser1@ubuntu:~$ ssh-keygen -t rsa -P ""
```

The second line will create an RSA key pair with an empty password.

Note:

P "", here indicates an empty password

You have to enable SSH access to your local machine with this newly created key which is done by the following command.

```
hduser1@ubuntu:~$ cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
```

The final step is to test the SSH setup by connecting to the local machine with the hduser1 user. The step is also needed to save your local machine's host key fingerprint to the hduser user's known hosts file.

```
hduser@ubuntu:~$ ssh localhost
```

If the SSH connection fails, we can try the following (optional):

Enable debugging with `ssh -vvv localhost` and investigate the error in detail.

Check the SSH server configuration in `/etc/ssh/sshd_config`. If you made any changes to the SSH server configuration file, you can force a configuration reload with `sudo /etc/init.d/ssh reload`.

## INSTALLATION

### Main Installation

Now, I will start by switching to `hduser`

```
hduser@ubuntu:~$ su - hduser1
```

Now, download and extract Hadoop 1.2.0

Setup Environment Variables for Hadoop

Add the following entries to `.bashrc` file

```
# Set Hadoop-related environment variables
```

```
export HADOOP_HOME=/usr/local/hadoop
```

```
# Add Hadoop bin/ directory to PATH
```

```
export PATH= $PATH:$HADOOP_HOME/bin
```

Configuration

```
hadoop-env.sh
```

Change the file: `conf/hadoop-env.sh`

```
#export JAVA_HOME=/usr/lib/j2sdk1.5-sun
```

to in the same file

```
# export JAVA_HOME=/usr/lib/jvm/java-6-openjdk-amd64 (for 64 bit)
```

```
# export JAVA_HOME=/usr/lib/jvm/java-6-openjdk-amd64 (for 32 bit)
```

```
conf/*-site.xml
```

Now we create the directory and set the required ownerships and permissions

```
hduser@ubuntu:~$ sudo mkdir -p /app/hadoop/tmp
```

```
hduser@ubuntu:~$ sudo chown hduser:hadoop /app/hadoop/tmp
```

```
hduser@ubuntu:~$ sudo chmod 750 /app/hadoop/tmp
```

The last line gives reading and writing permissions to the `/app/hadoop/tmp` directory

Error: If you forget to set the required ownerships and permissions, you will see a `java.io.IOException` when you try to format the name node.

Paste the following between `<configuration>`

In file conf/core-site.xml

```
<property>
  <name>hadoop.tmp.dir</name>
  <value>/app/hadoop/tmp</value>
  <description>A base for other temporary directories.</description>
</property>
<property>
  <name>fs.default.name</name>
  <value>hdfs://localhost:54310</value>
  <description>The name of the default file system. A URI whose
  scheme and authority determine the FileSystem implementation. The
  uri's scheme determines the config property (fs.SCHEME.impl) naming
  the FileSystem implementation class. The uri's authority is used to
  determine the host, port, etc. for a filesystem.</description>
</property>
```

In file conf/mapred-site.xml

```
<property>
  <name>mapred.job.tracker</name>
  <value>localhost:54311</value>
  <description>The host and port that the MapReduce job tracker runs
  at. If "local", then jobs are run in-process as a single map
  and reduce task.
  </description>
</property>
```

In file conf/hdfs-site.xml

```
<property>
  <name>dfs.replication</name>
  <value>1</value>
  <description>Default block replication.
```

The actual number of replications can be specified when the file is created.

The default is used if replication is not specified in create time.

</description>

</property>

Formatting the HDFS filesystem via the NameNode

To format the filesystem (which simply initializes the directory specified by the dfs.name.dir variable). Run the command

```
hduser@ubuntu:~$ /usr/local/hadoop/bin/hadoop namenode -format
```

Starting your single-node cluster

Before starting the cluster, we need to give the required permissions to the directory with the following command

```
hduser@ubuntu:~$ sudo chmod -R 777 /usr/local/hadoop
```

Run the command

```
hduser@ubuntu:~$ /usr/local/hadoop/bin/start-all.sh
```

This will startup a Namenode, Datanode, Jobtracker and a Tasktracker on the machine.

```
hduser@ubuntu:/usr/local/hadoop$ jps
```

### **Q7) Define big data, explain the characteristics of big data in detail.(6M)(W-18)(S-17)(W-17)**

“Data” is defined as ‘the quantities, characters, or symbols on which operations are performed by a computer, which may be stored and transmitted in the form of electrical signals and recorded on magnetic, optical, or mechanical recording media’, as a quick google search would show.

The concept of Big Data is nothing complex; as the name suggests, “Big Data” refers to copious amounts of data which are too large to be processed and analysed by traditional tools, and the data is not stored or managed efficiently. Since the amount of Big Data increases exponentially- more than 500 terabytes of data are uploaded to Face book alone, in a single day- it represents a real problem in terms of analysis.

However, there is also huge potential in the analysis of Big Data. The proper management and study of this data can help companies make better decisions based on usage statistics and user interests, thereby helping their growth. Some companies have even come up with new products and services, based on feedback received from Big Data analysis opportunities.

Classification is essential for the study of any subject. So Big Data is widely classified into

three main types, which are-

### **1. Structured data**

Structured Data is used to refer to the data which is already stored in databases, in an ordered manner. It accounts for about 20% of the total existing data, and is used the most in programming and computer-related activities.

There are two sources of structured data- machines and humans. All the data received from sensors, web logs and financial systems are classified under machine-generated data. These include medical devices, GPS data, data of usage statistics captured by servers and applications and the huge amount of data that usually move through trading platforms, to name a few.

Human-generated structured data mainly includes all the data a human input into a computer, such as his name and other personal details. When a person clicks a link on the internet, or even makes a move in a game, data is created- this can be used by companies to figure out their customer behaviour and make the appropriate decisions and modifications.

### **2. Unstructured data**

While structured data resides in the traditional row-column databases, unstructured data is the opposite- they have no clear format in storage. The rest of the data created, about 80% of the total account for unstructured big data. Most of the data a person encounters belongs to this category- and until recently, there was not much to do to it except storing it or analysing it manually.

Unstructured data is also classified based on its source, into machine-generated or human-generated. Machine-generated data accounts for all the satellite images, the scientific data from various experiments and radar data captured by various facets of technology.

Human-generated unstructured data is found in abundance across the internet, since it includes social media data, mobile data and website content. This means that the pictures we upload to our Facebook or Instagram handles, the videos we watch on YouTube and even the text messages we send all contribute to the gigantic heap that is unstructured data.

### **3. Semi-structured data.**

The line between unstructured data and semi-structured data has always been unclear, since most of the semi-structured data appear to be unstructured at a glance. Information that is not in the traditional database format as structured data, but contain some organizational properties which make it easier to process, are included in semi-structured data. For example, NoSQL documents are considered to be semi-structured, since they contain keywords that can be used to process the document easily.

Big Data analysis has been found to have a definite business value, as its analysis and processing can help a company achieve cost reductions and dramatic growth. So it is imperative that you do not wait too long to exploit the potential of this excellent business

opportunity.

---

**Q.8)List the different techniques used for clustering the big data. Explain k-means clustering.(5M)(W-18)**

### Types of Clustering

Broadly speaking, clustering can be divided into two subgroups :

- **Hard Clustering:** In hard clustering, each data point either belongs to a cluster completely or not. For example, in the above example each customer is put into one group out of the 10 groups.
- **Soft Clustering:** In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned. For example, from the above scenario each customer is assigned a probability to be in either of 10 clusters of the retail store.

### 3. Types of clustering algorithms

Since the task of clustering is subjective, the means that can be used for achieving this goal are plenty. Every methodology follows a different set of rules for defining the '*similarity*' among data points. In fact, there are more than 100 clustering algorithms known. But few of the algorithms are used popularly, let's look at them in detail:

- **Connectivity models:** As the name suggests, these models are based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away. These models can follow two approaches. In the first approach, they start with classifying all data points into separate clusters & then aggregating them as the distance decreases. In the second approach, all data points are classified as a single cluster and then partitioned as the distance increases. Also, the choice of distance function is subjective. These models are very easy to interpret but lacks scalability for handling big datasets. Examples of these models are hierarchical clustering algorithm and its variants.
- **Centroid models:** These are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters. K-Means clustering algorithm is a popular algorithm that falls into this category. In these models, the no. of clusters required at the end have to be mentioned beforehand, which makes it important to have prior knowledge of the dataset. These models run iteratively to find the local optima.
- **Distribution models:** These clustering models are based on the notion of how probable is it that all data points in the cluster belong to the same distribution (For

example: Normal, Gaussian). These models often suffer from overfitting. A popular example of these models is Expectation-maximization algorithm which uses multivariate normal distributions.

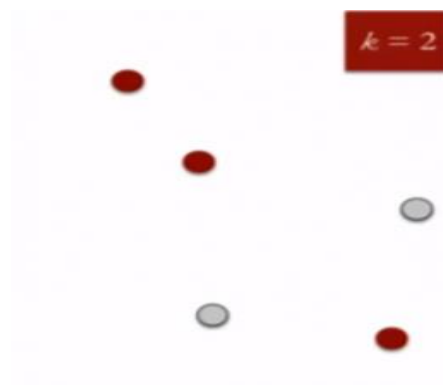
- **Density Models:** These models search the data space for areas of varied density of data points in the data space. It isolates various different density regions and assign the data points within these regions in the same cluster. Popular examples of density models are DBSCAN and OPTICS.

Now I will be taking you through two of the most popular clustering algorithms in detail – K Means clustering and Hierarchical clustering. Let's begin.

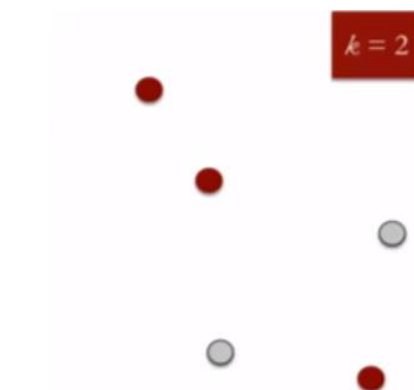
#### 4. K Means Clustering

K means is an iterative clustering algorithm that aims to find local maxima in each iteration. This algorithm works in these 5 steps :

1. Specify the desired number of clusters K : Let us choose  $k=2$  for these 5 data points in 2-D space.

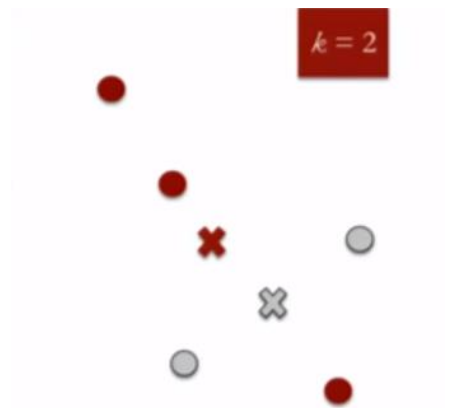


2. Randomly assign each data point to a cluster : Let's assign three points in cluster 1 shown using red color and two points in cluster 2 shown using grey color.

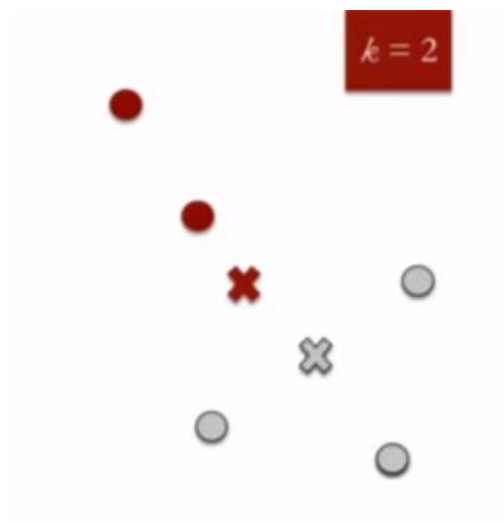




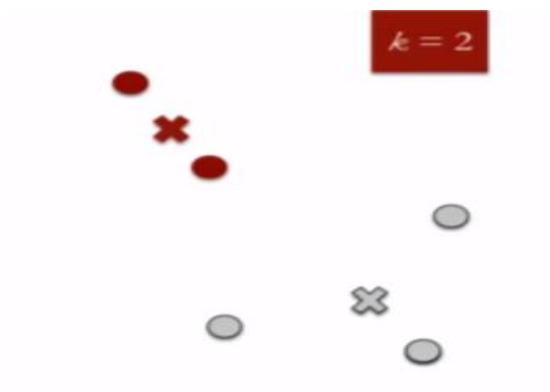
3. Compute cluster centroids : The centroid of data points in the red cluster is shown using red cross and those in grey cluster using grey cross.



4. Re-assign each point to the closest cluster centroid : Note that only the data point at the bottom is assigned to the red cluster even though its closer to the centroid of grey cluster. Thus, we assign that data point into grey cluster.



5. Re-compute cluster centroids : Now, re-computing the centroids for both the clusters.

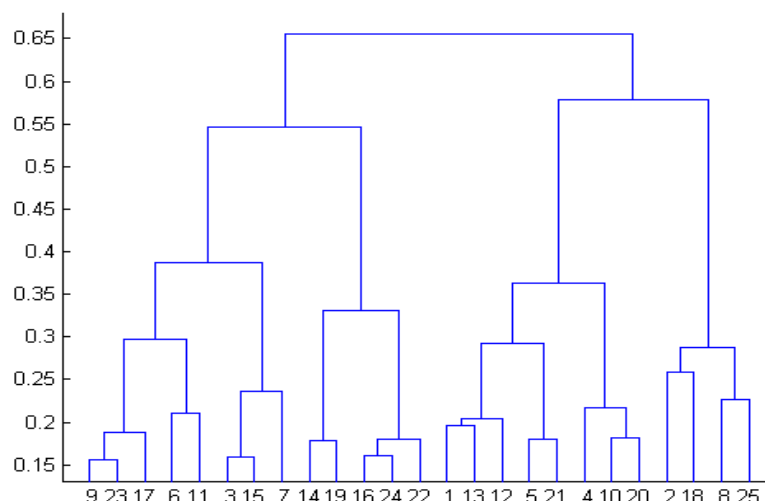


6. Repeat steps 4 and 5 until no improvements are possible : Similarly, we'll repeat the 4<sup>th</sup> and 5<sup>th</sup> steps until we'll reach global optima. When there will be no further switching of data points between two clusters for two successive repeats. It will mark the termination of the algorithm if not explicitly mentioned.

## 5. Hierarchical Clustering

Hierarchical clustering, as the name suggests is an algorithm that builds hierarchy of clusters. This algorithm starts with all the data points assigned to a cluster of their own. Then two nearest clusters are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.

The results of hierarchical clustering can be shown using dendrogram. The dendrogram can be interpreted as:

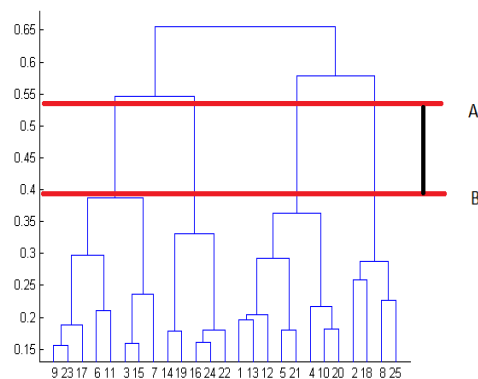


At the bottom, we start with 25 data points, each assigned to separate clusters. Two closest clusters are then merged till we have just one cluster at the top. The height in the dendrogram at which two clusters are merged represents the distance between two clusters in the data space.

The decision of the no. of clusters that can best depict different groups can be chosen by observing the dendrogram. The best choice of the no. of clusters is the no. of vertical lines in the dendrogram cut by a horizontal line that can transverse the maximum distance vertically without intersecting a cluster.

In the above example, the best choice of no. of clusters will be as the red horizontal line in the dendrogram below covers maximum vertical distance AB.

---



Two important things that you should know about hierarchical clustering are:

- This algorithm has been implemented above using bottom up approach. It is also possible to follow top-down approach starting with all data points assigned in the same cluster and recursively performing splits till each data point is assigned a separate cluster.
- The decision of merging two clusters is taken on the basis of closeness of these clusters. There are multiple metrics for deciding the closeness of two clusters :
  - Euclidean distance:  $\|a-b\|_2 = \sqrt{\sum (a_i - b_i)^2}$
  - Squared Euclidean distance:  $\|a-b\|_2^2 = \sum (a_i - b_i)^2$
  - Manhattan distance:  $\|a-b\|_1 = \sum |a_i - b_i|$
  - Maximum distance:  $\|a-b\|_{\text{INFINITY}} = \max_i |a_i - b_i|$
  - Mahalanobis distance:  $\sqrt{(a-b)^T S^{-1} (a-b)}$  {where, s : covariance matrix}

## 6. Difference between K Means and Hierarchical clustering

- Hierarchical clustering can't handle big data well but K Means clustering can. This is because the time complexity of K Means is linear i.e.  $O(n)$  while that of hierarchical clustering is quadratic i.e.  $O(n^2)$ .
- In K Means clustering, since we start with random choice of clusters, the results produced by running the algorithm multiple times might differ. While results are reproducible in Hierarchical clustering.
- K Means is found to work well when the shape of the clusters is hyper spherical (like circle in 2D, sphere in 3D).
- K Means clustering requires prior knowledge of K i.e. no. of clusters you want to divide your data into. But, you can stop at whatever number of clusters you find appropriate in hierarchical clustering by interpreting the dendrogram

## 7. Applications of Clustering

Clustering has a large no. of applications spread across various domains. Some of the most popular applications of clustering are:

- Recommendation engines
- Market segmentation
- Social network analysis
- Search result grouping
- Medical imaging
- Image segmentation
- Anomaly detection

## 8. Improving Supervised Learning Algorithms with Clustering

Clustering is an unsupervised machine learning approach, but can it be used to improve the accuracy of supervised machine learning algorithms as well by clustering the data points into similar groups and using these cluster labels as independent variables in the supervised machine learning algorithm? Let's find out.

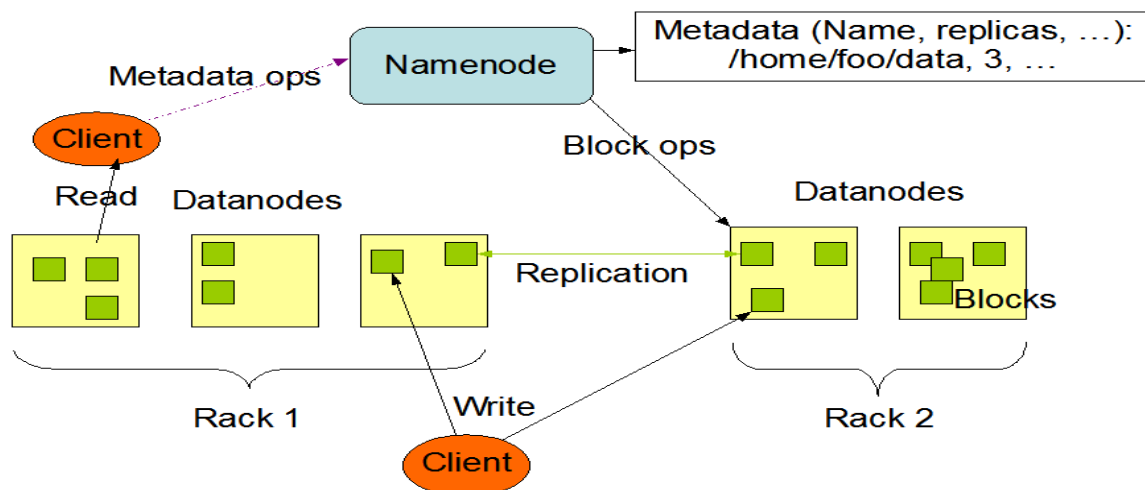
Let's check out the impact of clustering on the accuracy of our model for the classification problem using 3000 observations with 100 predictors of stock data to predicting whether the stock will go up or down using R. This dataset contains 100 independent variables from X1 to X100 representing profile of a stock and one outcome variable Y with two levels : 1 for rise in stock price and -1 for drop in stock price

---

**Q9) Explain with diagram HDFS architecture?(7M)(S-17)**

**Ans:**

## HDFS Architecture



### NameNode and DataNodes:

HDFS has a master/slave architecture. An HDFS cluster consists of a single NameNode, a master server that manages the file system namespace and regulates access to files by clients. In addition, there are a number of DataNodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS exposes a file system namespace and allows user data to be stored in files. Internally, a file is split into one or more blocks and these blocks are stored in a set of DataNodes. The NameNode executes file system namespace operations like opening, closing, and renaming files and directories. It also determines the mapping of blocks to DataNodes. The DataNodes are responsible for serving read and write requests from the file system's clients. The DataNodes also perform block creation, deletion, and replication upon instruction from the NameNode. The NameNode and DataNode are pieces of software designed to run on commodity machines. These machines typically run a GNU/Linux operating system (OS). HDFS is built using the Java language; any machine that supports Java can run the NameNode or the DataNode software. Usage of the highly portable Java language means that HDFS can be deployed on a wide range of machines. A typical deployment has a dedicated machine that runs only the NameNode software. Each of the other machines in the cluster runs one instance of the DataNode software. The architecture does not preclude running multiple DataNodes on the same machine but in a real deployment that is rarely the case. The existence of a single NameNode in a cluster greatly simplifies the architecture of the system. The NameNode is the arbitrator and repository for all HDFS metadata. The system is designed in such a way that user data never flows through the NameNode.

### The File System Namespace:

HDFS supports a traditional hierarchical file organization. A user or an application can create directories and store files inside these directories. The file system namespace hierarchy is similar to most other existing file systems; one can create and remove files, move a file from one directory to another, or rename a file. HDFS does not yet implement user quotas. HDFS does not support hard links or soft links. However, the HDFS architecture does not preclude implementing these features.

The NameNode maintains the file system namespace. Any change to the file system namespace or its properties is recorded by the NameNode. An application can specify the number of replicas of a file that should be maintained by HDFS. The number of copies of a file is called the replication factor of that file. This information is stored by the NameNode.

### Data Replication:

HDFS is designed to reliably store very large files across machines in a large cluster. It stores each file as a sequence of blocks; all blocks in a file except the last block are the same size. The blocks of a

file are replicated for fault tolerance. The block size and replication factor are configurable per file. An application can specify the number of replicas of a file. The replication factor can be specified at file creation time and can be changed later. Files in HDFS are write-once and have strictly one writer at any time.

The NameNode makes all decisions regarding replication of blocks. It periodically receives a Heartbeat and a Blockreport from each of the DataNodes in the cluster. Receipt of a Heartbeat implies that the DataNode is functioning properly. A Blockreport contains a list of all blocks on a DataNode.

=====

**Q.10) Explain Hadoop core component.(7M)(S-17)(W-17)**

**Ans:**

There are basically 3 important core components of hadoop -

**1. For computational processing i.e. MapReduce:**

MapReduce is the data processing layer of Hadoop. It is a software framework for easily writing applications that process the vast amount of structured and unstructured data stored in the Hadoop Distributed Filesystem (HDFS). It processes huge amount of data in parallel by dividing the job (submitted job) into a set of independent tasks (sub-job).

In Hadoop, MapReduce works by breaking the processing into phases: Map and Reduce. The Map is the first phase of processing, where we specify all the complex logic/business rules/costly code. Reduce is the second phase of processing, where we specify light-weight processing like aggregation/summation.

**2. For storage purpose i.e., HDFS :**

Acronym of Hadoop Distributed File System - which is basic motive of storage. It also works as the Master-Slave pattern. In HDFS NameNode acts as a master which stores the metadata of data node and Data node acts as a slave which stores the actual data in local disc parallel.

**3. Yarn :**

which is used for resource allocation. YARN is the processing framework in Hadoop, which provides Resource management, and it allows multiple data processing engines such as real-time streaming, data science and batch processing to handle data stored on a single platform.

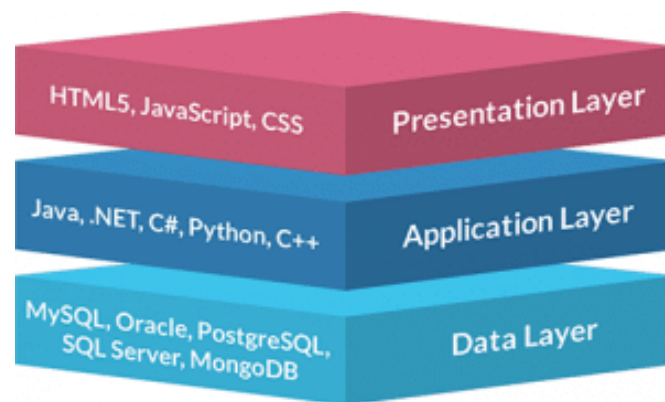
=====

**Q.11) Explain the three tiers of Big data framework.(5M)(w-17)**

A 3-tier architecture is a type of software architecture which is composed of three “tiers” or “layers” of logical computing. They are often used in applications as a specific type of client-server system. 3-tier architectures provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers. Doing so gives greater flexibility to development teams by allowing them to update a specific part of an application independently of the other parts. This added flexibility can improve overall time-to-market and decrease development cycle times by giving development teams the ability to replace or upgrade independent tiers without affecting the other parts of the system.

For example, the user interface of a web application could be redeveloped or modernized without affecting the underlying functional business and data access logic underneath. This architectural system is often ideal for embedding and integrating 3rd party software into an

existing application. This integration flexibility also makes it ideal for embedding analytics software into pre-existing applications and is often used by embedded analytics vendors for this reason. 3-tier architectures are often used in cloud or on-premises based applications as well as in software-as-a-service (SaaS) applications.



- **Presentation Tier-** The presentation tier is the front end layer in the 3-tier system and consists of the user interface. This user interface is often a graphical one accessible through a web browser or web-based application and which displays content and information useful to an end user. This tier is often built on web technologies such as HTML5, JavaScript, CSS, or through other popular web development frameworks, and communicates with others layers through API calls.
- **Application Tier-** The application tier contains the functional business logic which drives an application's core capabilities. It's often written in Java, .NET, C#, Python, C++, etc.
- **Data Tier-** The data tier comprises of the database/data storage system and data access layer. Examples of such systems are MySQL, Oracle, PostgreSQL, Microsoft SQL Server, MongoDB, etc. Data is accessed by the application layer via API calls.



**Fig:-Tier aArchitecture: JReport.**

The typical structure for a 3-tier architecture deployment would have the presentation tier deployed to a desktop, laptop, tablet or mobile device either via a web browser or a web-based application utilizing a web server. The underlying application tier is usually hosted on one or more application servers, but can also be hosted in the cloud, or on a dedicated workstation depending on the complexity and processing power needed by the application. And the data layer would normally

comprise of one or more relational databases, big data sources, or other types of database systems hosted either on-premises or in the cloud.

A simple example of a 3-tier architecture in action would be logging into a media account such as Netflix and watching a video. You start by logging in either via the web or via a mobile application. Once you've logged in you might access a specific video through the Netflix interface which is the presentation tier used by you as an end user. Once you've selected a video that information is passed on to the application tier which will query the data tier to call the information or in this case a video back up to the presentation tier. This happens every time you access a video from most media sites.

---



**Tulsiramji Gaikwad-Patil College of Engineering and Technology**

**Wardha Road, Nagpur-441 108**

**NAAC Accredited**

---

*Department of Computer Science & Engineering*

---

**Semester: B.E. Eighth Semester (CBS)**

**Subject: Clustering & Cloud Computing**

**Unit-4 Solution**

---

**Q1) What are different cloud security challenges?(7M)(S-17)**

**Ans:**

Cloud computing security challenges fall into three broad categories:

**Data Protection:** Securing your data both at rest and in transit

**User Authentication:** Limiting access to data and monitoring who accesses the data

**Disaster and Data Breach:** Contingency Planning

### **Data Protection**

Implementing a cloud computing strategy means placing critical data in the hands of a third party, so ensuring the data remains secure both at rest (data residing on storage media) as well as when in transit is of paramount importance. Data needs to be encrypted at all times, with clearly defined roles when it comes to who will be managing the encryption keys. In most cases, the only way to truly ensure confidentiality of encrypted data that resides on a cloud provider's storage servers is for the client to own and manage the data encryption keys.

### **User Authentication**

Data resting in the cloud needs to be accessible only by those authorized to do so, making it critical to both restrict and monitor who will be accessing the company's data through the cloud. In order to ensure the integrity of user authentication, companies need to be able to view data access logs and audit trails to verify that only authorized users are accessing the data. These access logs and audit trails additionally need to be secured and maintained for as long as the company needs or legal purposes require. As with all cloud computing security challenges, it's the responsibility of the customer to ensure that the cloud provider has taken all necessary security measures to protect the



customer's data and the access to that data.

### Contingency Planning

With the cloud serving as a single centralized repository for a company's mission-critical data, the risks of having that data compromised due to a data breach or temporarily made unavailable due to a natural disaster are real concerns. Much of the liability for the disruption of data in a cloud ultimately rests with the company whose mission-critical operations depend on that data, although liability can and should be negotiated in a contract with the services provider prior to commitment. A comprehensive security assessment from a neutral third-party is strongly recommended as well.

Companies need to know how their data is being secured and what measures the service provider will be taking to ensure the integrity and availability of that data should the unexpected occur. Additionally, companies should also have contingency plans in place in the event their cloud provider fails or goes bankrupt. Can the data be easily retrieved and migrated to a new service provider or to a non-cloud strategy if this happens? And what happens to the data and the ability to access that data if the provider gets acquired by another company?

=====

### Q.2) Explain virtual machine security in cloud computing.(7M)(S-17)

**Ans:**

2011 ended with the popularization of an idea: Bringing VMs (virtual machines) onto the cloud. Recent years have seen great advancements in both cloud computing and virtualization. On one hand there is the ability to pool various resources to provide software-as-a-service, infrastructure-as-a-service and platform-as-a-service. At its most basic, this is what describes cloud computing. On the other hand, we have virtual machines that provide agility, flexibility, and scalability to the cloud resources by allowing the vendors to copy, move, and manipulate their VMs at will. The term virtual machine essentially describes sharing the resources of one single physical computer into various computers within itself. VMware and virtual box are very commonly used virtual systems on desktops. Cloud computing effectively stands for many computers pretending to be one computing environment. Obviously, cloud computing would have many virtualized systems to maximize resources.

Keeping this information in mind, we can now look into the security issues that arise within a cloud-computing scenario. As more and more organizations follow the "Into the Cloud" concept, malicious hackers keep finding ways to get their hands on valuable information by manipulating safeguards and breaching the security layers (if any) of cloud environments. One issue is that the cloud-computing scenario is not as transparent as it claims to be. The service user has no clue about how his information is processed and stored. In addition, the service user cannot directly control the flow of data/information storage and processing. The service provider usually is not aware of the details of the service running on his or her environment. Thus, possible attacks on the cloud-computing environment can be classified in to:

### Resource attacks:

These kinds of attacks include manipulating the available resources into mounting a large-scale botnet attack. These kinds of attacks target either cloud providers or service providers.

**Data attacks:** These kinds of attacks include unauthorized modification of sensitive data at nodes, or performing configuration changes to enable a sniffing attack via a specific device etc. These attacks are focused on cloud providers, service providers, and also on service users.

**Denial of Service attacks:** The creation of a new virtual machine is not a difficult task, and thus, creating rogue VMs and allocating huge spaces for them can lead to a Denial of Service attack for service providers when they opt to create a new VM on the cloud. This kind of attack is generally called virtual machine sprawling.

**Backdoor:** Another threat on a virtual environment empowered by cloud computing is the use of backdoor VMs that leak sensitive information and can destroy data privacy.

Having virtual machines would indirectly allow anyone with access to the host disk files of the VM to take a snapshot or illegal copy of the whole System. This can lead to corporate espionage and piracy of legitimate products.

With so many obvious security issues (and a lot more can be added to the list), we need to enumerate some steps that can be used to secure virtualization in cloud computing.

The most neglected aspect of any organization is its physical security. An advanced social engineer can take advantage of weak physical-security policies an organization has put in place. Thus, it's important to have a consistent, context-aware security policy when it comes to controlling access to a data center. Traffic between the virtual machines needs to be monitored closely by using at least a few standard monitoring tools.

After thoroughly enhancing physical security, it's time to check security on the inside. A well-configured gateway should be able to enforce security when any virtual machine is reconfigured, migrated, or added. This will help prevent VM sprawls and rogue VMs. Another approach that might help enhance internal security is the use of third-party validation checks, preformed in accordance with security standards.

Checking virtual systems for integrity increases the capabilities for monitoring and securing environments. One of the primary focuses of this integrity check should be the seamless integration of existing virtual systems like VMware and virtual box. This would lead to file integrity checking and increased protection against data losses within VMs. Involving agentless anti-malware intrusion detection and prevention in one single virtual appliance (unlike isolated point security solutions) would contribute greatly towards VM integrity checks. This will greatly reduce operational overhead while adding zero footprints.

A server on a cloud may be used to deploy web applications, and in this scenario an OWASP top-ten vulnerability check will have to be performed. Data on a cloud should be encrypted with suitable encryption and data-protection algorithms. Using these algorithms, we can check the integrity of the user profile or system profile trying to access disk files on the VMs. Profiles lacking in security protections can be considered infected by malwares. Working with a system ratio of one user to one machine would also greatly reduce risks in virtual computing platforms. To enhance the security aspect even more, after a particular environment is used, it's best to sanitize the system (reload) and destroy all the residual data. Using incoming IP addresses to determine scope on Windows-based machines, and using SSH configuration settings on Linux machines, will help maintain a secure one-to-one connection.

=====

=

**Q3) Explain Identity Access Management? (7M)(S-17)****Ans:**

Identity and access management (IAM) is a framework for business processes that facilitates the management of electronic or digital identities. The framework includes the organizational policies for managing digital identity as well as the technologies needed to support identity management.

With IAM technologies, IT managers can control user access to critical information within their organizations. Identity and access management products offer role-based access control, which lets system administrators regulate access to systems or networks based on the roles of individual users within the enterprise.

In this context, access is the ability of an individual user to perform a specific task, such as view, create or modify a file. Roles are defined according to job competency, authority and responsibility within the enterprise.

Systems used for identity and access management include single sign-on systems, multifactor authentication and access management. These technologies also provide the ability to securely store identity and profile data as well as data governance functions to ensure that only data that is necessary and relevant is shared.

These products can be deployed on premises, provided by a third party vendor via a cloud-based subscription model or deployed in a hybrid cloud.

=====

=====

**Q.4) What do you mean by cloud contract? Explain in details(7M)(S-17)****Ans:**

"Cloud computing" means accessing computer capacity and programming facilities online or "in the cloud". Customers are spared the expense of purchasing, installing and maintaining hardware and software locally.

Customers can easily expand or reduce IT capacity according to their needs. This essentially transforms computing into an on-demand utility. An added boon is that data can be accessed and processed from anywhere via the Internet.

Unfortunately, consumers and companies are often reluctant to take advantage of cloud computing services either because contracts are unclear or are unbalanced in favour of service providers. Existing regulations and national contract laws may not always be adapted to cloud-based services. Protection of personal data in a cloud environment also needs to be addressed. Adapting contract law is therefore an important part of the Commission's cloud computing strategy.

**Safe and fair contracts for cloud computing**

The Commission is working towards cloud computing contracts that contain safe and fair terms and conditions for all parties. On 18 June 2013, the Commission set up a group of experts to define safe and fair conditions and identify best practices for cloud computing contracts. The Commission has

also launched a comparative study on cloud computing contracts to supplement the work of the Expert Group.

-----

-----

### **Q5) Justify cloud security challenges in detail.(7M)(W-17)(W-18)**

Cloud computing security challenges fall into three broad categories:

**Data Protection:** Securing your data both at rest and in transit

**User Authentication:** Limiting access to data and monitoring who accesses the data

**Disaster and Data Breach:** Contingency Planning

#### **Data Protection**

Implementing a cloud computing strategy means placing critical data in the hands of a third party, so ensuring the data remains secure both at rest (data residing on storage media) as well as when in transit is of paramount importance. Data needs to be encrypted at all times, with clearly defined roles when it comes to who will be managing the encryption keys. In most cases, the only way to truly ensure confidentiality of encrypted data that resides on a cloud provider's storage servers is for the client to own and manage the data encryption keys.

#### **User Authentication**

Data resting in the cloud needs to be accessible only by those authorized to do so, making it critical to both restrict and monitor who will be accessing the company's data through the cloud. In order to ensure the integrity of user authentication, companies need to be able to view data access logs and audit trails to verify that only authorized users are accessing the data. These access logs and audit trails additionally need to be secured and maintained for as long as the company needs or legal purposes require. As with all cloud computing security challenges, it's the responsibility of the customer to ensure that the cloud provider has taken all necessary security measures to protect the customer's data and the access to that data.

#### **Contingency Planning**

With the cloud serving as a single centralized repository for a company's mission-critical data, the risks of having that data compromised due to a data breach or temporarily made unavailable due to a natural disaster are real concerns. Much of the liability for the disruption of data in a cloud ultimately rests with the company whose mission-critical operations depend on that data, although liability can and should be negotiated in a contract with the services provider prior to commitment. A comprehensive security assessment from a neutral third-party is strongly recommended as well.

Companies need to know how their data is being secured and what measures the service provider will be taking to ensure the integrity and availability of that data should the unexpected occur. Additionally, companies should also have contingency plans in place in the event their cloud provider fails or goes bankrupt. Can the data be easily retrieved and migrated to a new service provider or to a

non-cloud strategy if this happens? And what happens to the data and the ability to access that data if the provider gets acquired by another company?

---

#### **Q.6) Explain the contracting models in cloud.(6M)(W-17)(W-16)**

Organizations can only reap the advantages of Cloud computing once the contract for such a service has been agreed and is water-tight. This article provides a guide for what contract managers need to consider when negotiating a deal for their organizations' 'Cloud'.

Today it is possible to contract for 'on-demand' computing which is internet-based whereby shared resources and information are provided, i.e., the phenomenon known as 'Cloud' computing. In addition to economies of scale enjoyed by multiple users, one of the big advantages is that work can be allocated depending upon the time zone users are operating in. Thus, entities operating during European business hours would not be operating at the same time that users would be operating in North America. The result is a reduced overall cost of the resources including reduced operating and maintenance expenses. In addition, multiple users can obtain access to a single server without the need to purchase licenses for different applications. Moreover, it has the advantages of a 'pay-as-you-go' model as well as a host of other potential advantages which have been referred to as 'agility', 'device and location independence', 'scalability and elasticity', etc. Nevertheless, questions remain and Cloud computing presents potential commercial and contractual pitfalls for the unwary.

It is clear that despite some unanswered questions, computing resources delivered over the internet are here today and here to stay. Analogous to a utility, the advantages of the 'Cloud' allow the cost of the infrastructure, platform and service delivery to be shared amongst many users. But does this in any way change basic contracting principles and time honored sound contract management practices? The short answer is "No". However, this does not detract from the fact that contracting for and managing contracts for Cloud computing services can be a challenge. The complexity associated with such contracts can be reduced by addressing some early threshold questions.

Definitions are of vital importance in any contract, including ones for Cloud computing.

A key concern is data security. Thus, it is important to define what is meant by 'data' and distinguish between 'personal data' and 'other data'. A distinction can be made between data that is identified to or provided by the customer and information that is derived from the use of that data, e.g., metadata. Careful attention should be paid to how the contract defines 'consent' to use derived data. Generally, any such data should be explicit and based upon a meaningful understanding of how the derived data is going to be used.

Security standards might warrant different levels of security depending upon the nature of the data. Likewise, what is meant by 'security'? The tendency is to define security only in technical terms, but security should be defined to include a broad range of data protection obligations. There are, of course, many other potential key terms that warrant careful

definition in contracts for Cloud computing services. However, this is nothing new to the field of good contracting and sound contract management practices.

Naturally, if personal or confidential information is going to be entrusted to a third party, the recipient must comply with appropriate contractual controls and statutory requirements regarding privacy and confidentiality. This is why taking the time to define things carefully is so important. Simply asserting that those security considerations will be ‘reasonable’ or comply with ‘industry standards’ falls short of what is necessary. Abstract promises should be rejected in favor of specific protocols and clear audit requirements as well as the obligation to comply with specific legal requirements. This is true for all transactions.

‘Notice’ provisions are common in contracts. It follows that if you are contracting for computing resources delivered over the internet you’d want clearly defined notice provisions that would require notice of any security breaches as well as any discovery requests made in the context of litigation. ‘Storage’ is also a key concept and term to be addressed and warrants special attention. From a risk management standpoint you’d also want to understand the physical location of the equipment and data storage. Perhaps geographical distance and diversity is both a challenge and an opportunity in terms of risk management.

Defining success is always a challenge in any contract. The enemy of all good contracts is ambiguity. When it comes to ‘availability’, users should avoid notions that the service provider will use its ‘best efforts’ and exercise ‘reasonable care’. Clear availability targets are preferred since there must be a way to measure availability. Usually, availability measured in terms of an expressed percentage ends up being difficult if not impossible to understand let alone enforce. Expressing availability in terms of units of time (e.g., a specified number of minutes per day of down time) is preferable.



Early on, it makes sense to focus on the deployment model that works best for your organization. These fall into three basic categories: Private Cloud, Public Cloud and Hybrid Cloud. As the name suggests, a Private Cloud is an infrastructure operated for or by a single entity whether managed by that entity or some other third party. A Private Cloud can, of course, be hosted internally or externally. A Public Cloud is when services are provided over a network that is open to the public and may be free. Examples of Public Cloud include Google, Amazon and Microsoft and provide services generally available via the internet. A Hybrid Cloud (sometimes referred to as a ‘Community Cloud’) is composed of both private and/or public entities, but decided to enter into some arrangement.

Consider the type of contractual arrangement. Is the form of contract essentially a ‘service’, ‘license’, ‘lease’ or some other form of contractual arrangement? Service agreements, licenses and leases have different structures. Perhaps the contract for Cloud computing services contains aspects of all these different types of agreements, including ones for IT infrastructure. Best to consider this early. Yet, such considerations are common to all contracting efforts.

A threshold question is whether the data being stored or processed is being sent out of the country and any associated special legal compliance issues. However, essentially all the normal contractual concerns apply to contracts involving the ‘Cloud’. These include termination or suspension as well as the return of data in the case of threats to security or data integrity. Likewise, data ownership, data comingling, access to data, service provider viability, integration risks, publicity, service levels, disaster recovery and changes in control or ownership are all important in all contracts involving personal, sensitive or proprietary information, including contracts involving Cloud computing services.

How such services are taxed at the local or even international levels also presents some interesting questions, the answers to which may vary by jurisdiction and over time. However, the tax implications of cross boarder transactions by multinationals is hardly a new topic.

Although the issues are many, they are closely related to what any good negotiator or contract manager would consider early on. Developing a checklist can often be a useful exercise, especially when dealing with a new topic like Cloud computing.

---

#### **Q.7) Write short notes on any three.(13M)(W-17)(W-16)(W-18)**

- a) Host Level Security**
- b) Network Level Security**
- c) Application Level Security**
- d) Virtual Machine Security**

##### **a) Host Security**

Host security describes how your server is set up for the following tasks:

- Preventing attacks.
- Minimizing the impact of a successful attack on the overall system.
- Responding to attacks when they occur.

It always helps to have software with no security holes. Good luck with that! In the real world, the best approach for preventing attacks is to assume your software has security holes. As I noted earlier in this chapter, each service you run on a host presents a distinct attack vector into the host. The more attack vectors, the more likely an attacker will find one with a security exploit. You must therefore minimize the different kinds of software running on a server.

Given the assumption that your services are vulnerable, your most significant tool in preventing attackers from exploiting a vulnerability once it becomes known is the rapid rollout of security patches. Here's where the dynamic nature of the cloud really alters what you can do from a security perspective. In a traditional data center, rolling out security patches across an entire infrastructure is time-consuming and risky. In the cloud, rolling out a patch across the infrastructure takes three simple steps:

1. Patch your AMI with the new security fixes.
2. Test the results.
3. Relaunch your virtual servers.

### **b) Network Level Security**

All data on the network need to be secured. Strong network traffic encryption techniques such as Secure Socket Layer (SSL) and the Transport Layer Security (TLS) can be used to prevent leakage of sensitive information. Several key security elements such as data security, data integrity, authentication and authorization, data confidentiality, web application security, virtualization vulnerability, availability, backup, and data breaches should be carefully considered to keep the cloud up and running continuously.

## **C)APPLICATION LEVEL SECURITY**

Studies indicate that most websites are secured at the network level while there may be security loopholes at the application level which may allow information access to unauthorized users. Software and hardware resources can be used to provide security to applications. In this way, attackers will not be able to get control over these applications and change them. XSS attacks, Cookie Poisoning, Hidden field manipulation, SQL injection attacks, DoS attacks, and Google Hacking are some examples of threats to application level security which resulting from the unauthorized usage of the applications.

### **D) Virtual Machine Security**

Recent years have seen great advancements in both cloud computing and virtualization. On one hand there is the ability to pool various resources to provide software-as-a-service, infrastructure-as-a-service and platform-as-a-service. At its most basic, this is what describes cloud computing. On the other hand, we have virtual machines that provide agility, flexibility, and scalability to the cloud resources by allowing the vendors to copy, move, and manipulate their VMs at will. The term *virtual machine* essentially describes sharing the resources of one single physical computer into various computers within itself. *VMware* and *virtual box* are very commonly used virtual systems on desktops. Cloud computing effectively stands for many computers pretending to be one computing environment. Obviously, cloud computing would have many virtualized systems to maximize resources.

Keeping this information in mind, we can now look into the security issues that arise within a cloud-computing scenario. As more and more organizations follow the "Into the Cloud" concept, malicious hackers keep finding ways to get their hands on valuable information by manipulating



safeguards and breaching the security layers (if any) of cloud environments. One issue is that the cloud-computing scenario is not as transparent as it claims to be. The service user has no clue about how his information is processed and stored. In addition, the service user cannot directly control the flow of data/information storage and processing. The service provider usually is not aware of the details of the service running on his or her environment. Thus, possible attacks on the cloud-computing environment can be classified in to:

1. **Resource attacks:** These kinds of attacks include manipulating the available resources into mounting a large-scale botnet attack. These kinds of attacks target either cloud providers or service providers.
  2. **Data attacks:** These kinds of attacks include unauthorized modification of sensitive data at nodes, or performing configuration changes to enable a sniffing attack via a specific device etc. These attacks are focused on cloud providers, service providers, and also on service users.
  3. **Denial of Service attacks:** The creation of a new virtual machine is not a difficult task, and thus, creating rogue VMs and allocating huge spaces for them can lead to a Denial of Service attack for service providers when they opt to create a new VM on the cloud. This kind of attack is generally called virtual machine sprawling.
  4. **Backdoor:** Another threat on a virtual environment empowered by cloud computing is the use of backdoor VMs that leak sensitive information and can destroy data privacy.
  5. Having virtual machines would indirectly allow anyone with access to the host disk files of the VM to take a snapshot or illegal copy of the whole System. This can lead to corporate espionage and piracy of legitimate products.
- 

#### **Q.8)Write short note on Infrastructure security in cloud Computing.(6M)(W-16)**

Cloud infrastructure refers to a virtual infrastructure that is delivered or accessed via a network or the internet. This usually refers to the on-demand services or products being delivered through the model known as infrastructure as a service (IaaS), a basic delivery model of cloud computing. This is a highly automated offering where computing resources complemented with storage and networking services are provided to the user. In essence, users have an IT infrastructure that they can use for themselves without ever having to pay for the construction of a physical infrastructure.

#### ***Cloud Infrastructure***

Cloud infrastructure is one of the most basic products delivered by cloud computing services through the IaaS model. Through the service, users can create their own IT infrastructure complete with processing, storage and networking fabric resources that can be configured in any way, just as with a physical data center enterprise infrastructure. In most cases, this provides more flexibility in infrastructure design, as it can be easily set up, replaced or deleted as opposed to a physical one, which requires manual work, especially when network connectivity needs to be modified or reworked.

---

A cloud infrastructure includes virtual machines and components such as:

---

- Virtual servers
- Virtual PCs
- Virtual network switches/hubs/routers
- Virtual memory
- Virtual storage clusters

All of these elements combine to create a full IT infrastructure that works just as well as a physical one, but boasts such benefits as:

- Low barrier to entry
  - Low capital requirement
  - Low total cost of ownership
  - Flexibility
  - Scalability
- 

### **Q9) Explain Identity Access management. (7M)(W16)(W-18)**

Security in any system involves primarily ensuring that the right entity gets access to only the authorized data in the authorized format at an authorized time and from an authorized location. Identity and access management (IAM) is of prime importance in this regard as far as Indian businesses are concerned. This effort should be complemented by the maintenance of audit trails for the entire chain of events from users logging in to the system, getting authenticated and accessing files or running applications as authorized.

Even in a closed, internal environment with a well-established “trust boundary”, managing an Active Directory server, an LDAP server or other alternatives, is no easy task. And for IAM in the cloud, the challenges and problems are magnified many times over. An Indian organization moving to the cloud could typically have applications hosted on the cloud and a database maintained internally, with users logging on and getting authenticated internally on a local Active Directory server. Just imagine attempting single sign-on (SSO) functionality in such a scenario! Cloud delivery models comprising mainly SaaS, PaaS and IaaS require seamless integration between cloud services and the organization’s IAM practices, processes and procedures, in a scalable, effective and efficient manner.

### **Identity provisioning challenges**

The biggest challenge for cloud services is **identity provisioning**. This involves secure and timely management of on-boarding (provisioning) and off-boarding (deprovisioning) of users in the cloud.

When a user has successfully authenticated to the cloud, a portion of the system resources in terms of CPU cycles, memory, storage and network bandwidth is allocated. Depending on the capacity identified for the system, these resources are made available on the system even if

no users have been logged on. Based on projected capacity requirements, cloud architects may decide on a 1:4 scale or even 1:2 or lower ratios. If projections are exceeded and more users logon, the system performance may be affected drastically. Simultaneously, adequate measures need to be in place to ensure that as usage of the cloud drops, system resources are made available for other objectives; else they will remain unused and constitute a dead investment.

---



**Tulsiramji Gaikwad-Patil College of Engineering and Technology**  
Wardha Road, Nagpur-441 108  
NAAC Accredited

---

*Department of Computer Science & Engineering*

---

**Semester: B.E. Eighth Semester (CBS)**  
**Subject: Clustering & Cloud Computing**  
**Unit-5 Solution**

---

**Q.1) Explain object oriented concept in C# .net.(6M)(S-17)(W-16)(W-18)**

**Ans:**

Introduction to Object Oriented Programming (OOP) concepts in C#: Abstraction, Encapsulation, Inheritance and Polymorphism.

### **OOP Features**

Object Oriented Programming (OOP) is a programming model where programs are organized around objects and data rather than action and logic.

OOP allows decomposition of a problem into a number of entities called objects and then builds data and functions around these objects.

- The software is divided into a number of small units called objects. The data and functions are built around these objects.
- The data of the objects can be accessed only by the functions associated with that object.
- The functions of one object can access the functions of another object.

OOP has the following important features.

### **Class**

A class is the core of any modern Object Oriented Programming language such as C#.

In OOP languages it is mandatory to create a class for representing data.

A class is a blueprint of an object that contains variables for storing data and functions to perform operations on the data.

A class will not occupy any memory space and hence it is only a logical representation of data.

To create a class, you simply use the keyword "class" followed by the class name:

```
class Employee  
  
{  
  
}
```

## Object

Objects are the basic run-time entities of an object oriented system. They may represent a person, a place or any item that the program must handle.

"An object is a software bundle of related variable and methods."

"An object is an instance of a class"

A class will not occupy any memory space. Hence to work with the data represented by the class you must create a variable for the class, that is called an object.

When an object is created using the new operator, memory is allocated for the class in the heap, the object is called an instance and its starting address will be stored in the object in stack memory.

When an object is created without the new operator, memory will not be allocated in the heap, in other words an instance will not be created and the object in the stack contains the value null.

When an object contains null, then it is not possible to access the members of the class using that object.

```
class Employee  
  
{  
  
}
```

Syntax to create an object of class Employee:

```
Employee objEmp = new Employee();
```

All the programming languages supporting Object Oriented Programming will be supporting these three main concepts,

- 1)Encapsulation
- 2)Inheritance
- 3)sPolymorphism

## Abstraction

Abstraction is "To represent the essential feature without representing the background details."

Abstraction lets you focus on what the object does instead of how it does it.

Abstraction provides you a generalized view of your classes or objects by providing relevant information.

Abstraction is the process of hiding the working style of an object, and showing the information of an object in an understandable manner.

### **Encapsulation**

Wrapping up a data member and a method together into a single unit (in other words class) is called Encapsulation.

Encapsulation is like enclosing in a capsule. That is enclosing the related operations and data related to an object into that object.

Encapsulation is like your bag in which you can keep your pen, book etcetera. It means this is the property of encapsulating members and functions.

Encapsulation means hiding the internal details of an object, in other words how an object does something.

Encapsulation prevents clients from seeing its inside view, where the behaviour of the abstraction is implemented.

Encapsulation is a technique used to protect the information in an object from another object.

Hide the data for security such as making the variables private, and expose the property to access the private data that will be public.

### **Inheritance**

When a class includes a property of another class it is known as inheritance.

Inheritance is a process of object reusability.

### **Polymorphism**

Polymorphism means one name, many forms. One function behaves in different forms. In other words, "Many forms of a single object is called Polymorphism."

=====

**Q.2) Write a program in C#.net- to demonstrate object oriented concept: Design user Interface.(7M)(S-17)**

**Ans:**

```
1)namespace CLASS_DEMO
```

```
{
```

```
    class person
```

```
{  
    private  
    string name;  
    int age;  
    double salary;  
    public void getdata()  
    {  
        Console.Write("Enter Your Name:-");  
        name = Console.ReadLine();  
        Console.Write("Enter the age:-");  
        age = Convert.ToInt32(Console.ReadLine());  
        Console.Write("Enter the salary:-");  
        salary = Convert.ToDouble(Console.ReadLine());  
    }  
    public void putdata()  
    {  
        Console.WriteLine("NAME==>" + name);  
        Console.WriteLine("AGE==>" + age);  
        Console.WriteLine("SALARY==>" + salary);  
    }  
}  
  
class Program  
{  
    static void Main(string[] args)  
    {  
        person obj = new person();  
        obj.getdata();  
        obj.putdata();  
        Console.ReadLine();  
    }  
}
```

```
    }  
}  
}  
2)  
namespace single_inheritance  
{  
    class Animal  
    {  
        public void Eat()  
        {  
            Console.WriteLine("Every animal eats something.");  
        }  
        public void dosomething()  
        {  
            Console.WriteLine("Every animal does something.");  
        }  
    }  
    class cat : Animal  
    {  
        static void Main(string[] args)  
        {  
            cat objcat = new cat();  
            objcat.Eat();  
            objcat.dosomething();  
            Console.ReadLine();  
        }  
    }  
}
```

3)

```

namespace Method_Overloading_1
{
    class Area
    {
        static public int CalculateArea(int len, int wide)
        {
            return len * wide;
        }

        static public double CalculateArea(double valone, double valtwo)
        {
            return 0.5 * valone * valtwo;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            int length = 10;
            int breath = 22;
            double tbase = 2.5;
            double theight = 1.5;

            Console.WriteLine("Area of Rectangle:-" + Area.CalculateArea(length, breath));
            Console.WriteLine("Area of triangle:-" + Area.CalculateArea(tbase, theight));
            Console.ReadLine();
        }
    }
}

```

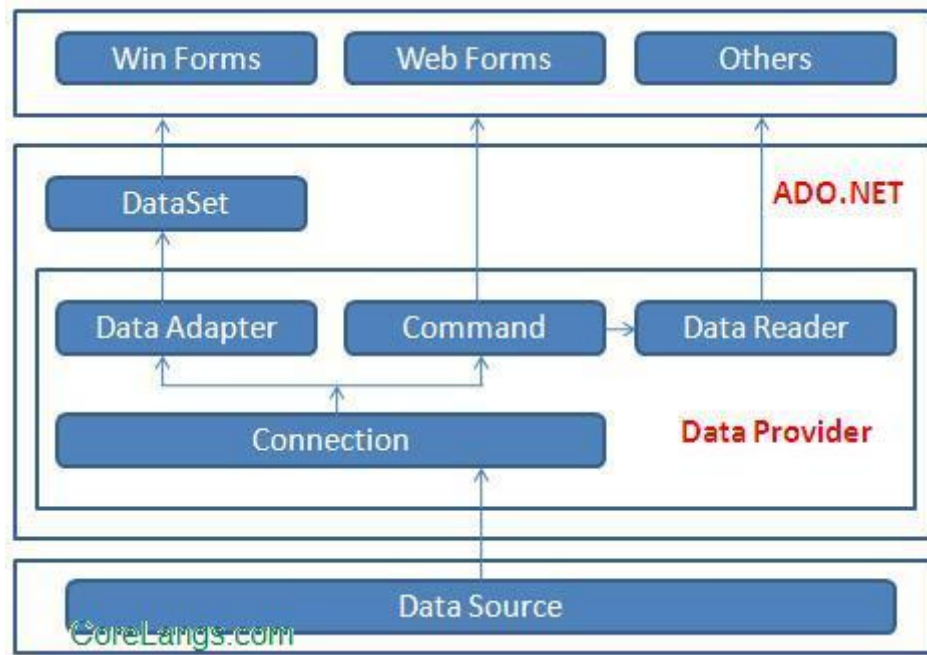
=====

**Q.3) Explain the Architecture of ADO. Net(6M)(S-17)(w-17)(W-16)(W-18)**



Ans:

## ADO.NET

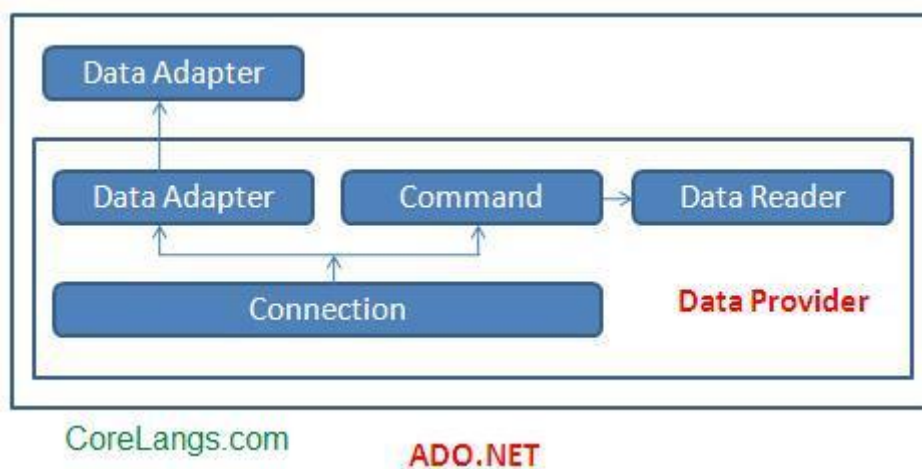


**Fig:- asp.net-ado.net-architecture**

ADO.NET consist of a set of Objects that expose data access services to the .NET environment. It is a data access technology from Microsoft .Net Framework , which provides communication between relational and non relational systems through a common set of components .

System.Data namespace is the core of ADO.NET and it contains classes used by all data providers. ADO.NET is designed to be easy to use, and Visual Studio provides several wizards and other features that you can use to generate ADO.NET data access code.

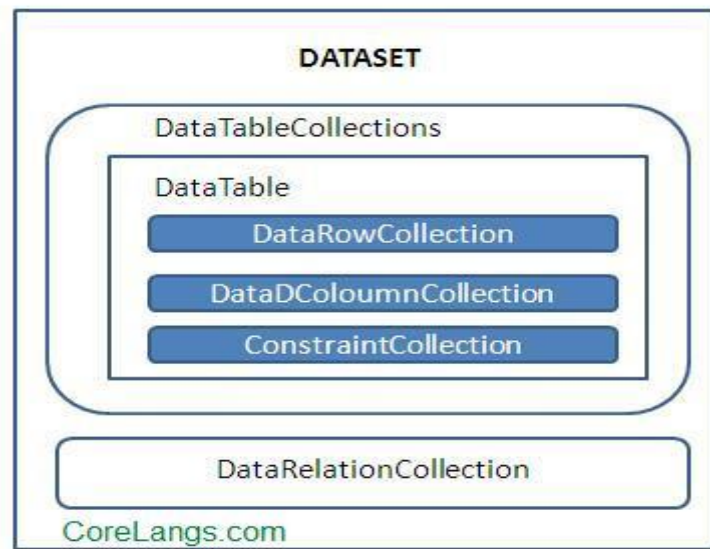
Data Providers and DataSet



**Fig:-asp.net-ado.net**

The two key components of ADO.NET are Data Providers and DataSet . The Data Provider classes

## DataSet



**Fig:-asp.net-dataset**

DataSet provides a disconnected representation of result sets from the Data Source, and it is completely independent from the Data Source. DataSet provides much greater flexibility when dealing with related Result Sets.

DataSet contains rows, columns, primary keys, constraints, and relations with other DataTable objects. It consists of a collection of DataTable objects that you can relate to each other with DataRelation objects. The DataAdapter Object provides a bridge between the DataSet and the Data Source.

---

**Q.4) Write and explain code in ASP. NET to create login page.(7M)(S-17)**

**Ans:**

**Introduction**

This article demonstrates how to create a login page in an ASP.NET Web Application, using C# connectivity by SQL server. This article starts with an introduction of the creation of the database and table in SQL Server. Afterwards, it demonstrates how to design ASP.NET login page. In the end, the article discusses how to create a connection ASP.NET Web Application to SQL Server.

**Prerequisites**

VS2010/2012/2013/15/17, SQL Server 2005/08/2012

Project used version

VS2013, SQL SERVER 2012

**Step 1**

Creating a database and a table

To create a database, write the query in SQL Server

Create database abcd //Login is my database name

Use abcd //Select database or use database

Create table Ulogin // create table Ulogin is my table name

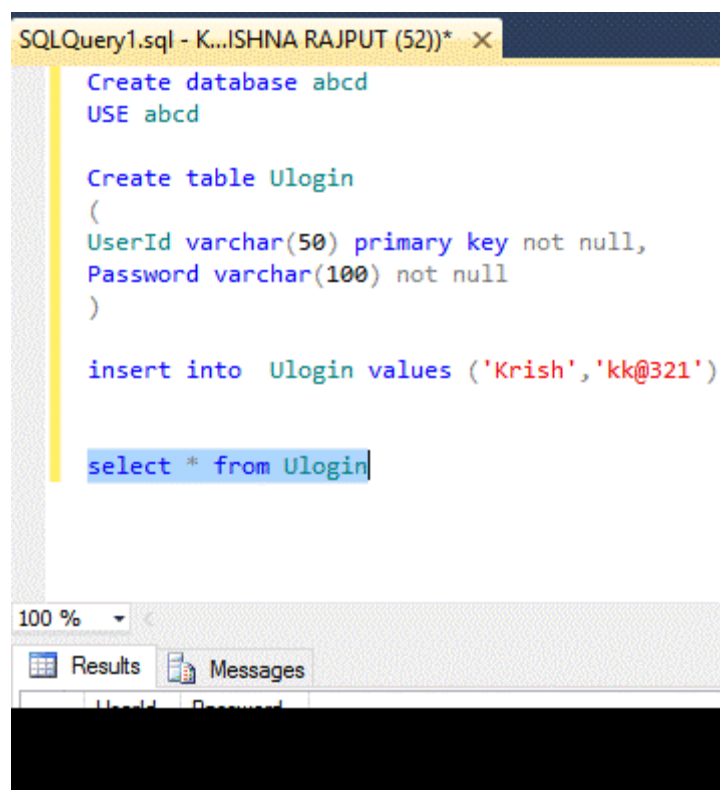
(

    UserId varchar(50) primary key not null, //primary key not accept null value

    Password varchar(100) not null

)

insert into Ulogin values ('Krish','kk@321') //insert value in Ulogin table



```

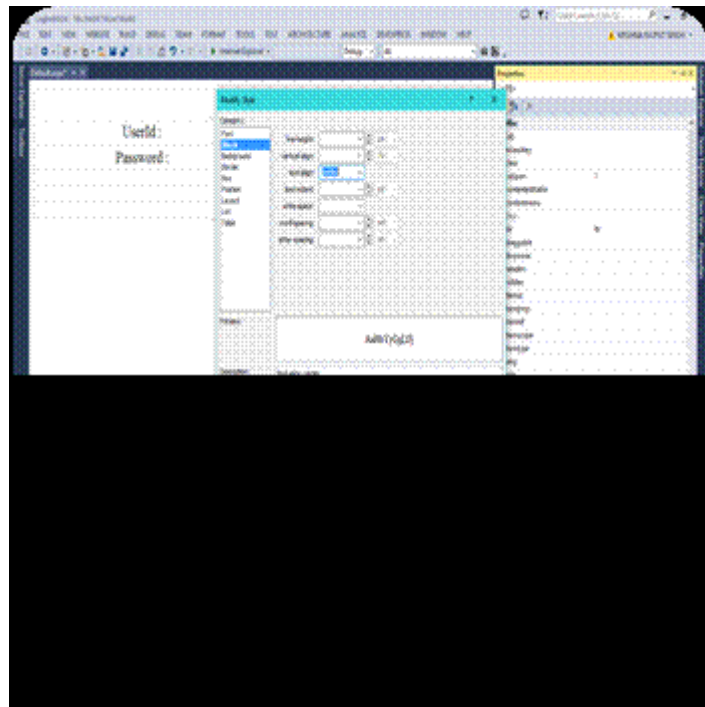
SQLQuery1.sql - K...ISHNA RAJPUT (52))* X
Create database abcd
USE abcd

Create table Ulogin
(
    UserId varchar(50) primary key not null,
    Password varchar(100) not null
)

insert into Ulogin values ('Krish','kk@321')

select * from Ulogin
  
```

Let's start design login view in ASP.NET Web Application. I am using simple design to view this article is not the purpose of design, so let's start opening VS (any version) and go to File, select New select Web site. You can also use shortcut key (Shift+Alt+N). When you are done with expanding Solution Explorer, right click on your project name, select add click Add New Item (for better help, refer the screenshot given below). Select Web Form, if you want to change Web form name. You can save it as it is. Default.aspx is added in my project. Now, let's design my default.aspx page in <div>tag insert table, as per required the rows and columns and set the layout style of the table. If you want all tools set in center, go to Table properties and click Style text align.



```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>
```

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
```

```
<title></title>
```

```
<style type="text/css">
```

```
.auto-style1 {
    width: 100%;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<form id="form1" runat="server">
```

```
<div>
```

```
<table class="auto-style1">
```

```
<tr>
```

```

        <td colspan="6" style="text-align: center; vertical-align: top">
    </tr>
</table>

```

```

    </div>
</form>
</body>

```

Afterwards, drag and drop two labels, two textbox and one button below design view source code. Set the password textbox properties Text Mode as a password.

Complete source code is given below.

```

<% @ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>

```

```

<!DOCTYPE html>

```

```

<html xmlns="http://www.w3.org/1999/xhtml">

```

```

<head runat="server">

```

```

    <title></title>

```

```

    <style type="text/css">

```

```

        .auto-style1 {
            width: 100%;
        }

```

```

    </style>

```

```

</head>

```

```

<body>

```

```

    <form id="form1" runat="server">

```

```

        <div>

```

```

            <table class="auto-style1">

```

```

                <tr>

```

```

                    <td colspan="6" style="text-align: center; vertical-align: top">

```

```

        <asp:Label ID="Label1" runat="server" Font-Bold="True" Font-Size="XX-Large" Font-Underline="True" Text="Log In "></asp:Label>

```

```

    </td>

```

```

</tr>

```

```

<tr>

```

```

    <td> </td>

```

```

    <td style="text-align: center">

```

```

        <asp:Label ID="Label2" runat="server" Font-Size="X-Large" Text="UserId : "></asp:Label>

```

```

    </td>

```

```

    <td style="text-align: center">

```

```

        <asp:TextBox ID="TextBox1" runat="server" Font-Size="X-Large"></asp:TextBox>

```

```

    </td>

```

```

    <td> </td>

```

```

    <td> </td>

```

```

    <td> </td>

```

```

</tr>

```

```

<tr>

```

```

    <td> </td>

```

```

    <td style="text-align: center">

```

```

        <asp:Label ID="Label3" runat="server" Font-Size="X-Large" Text="Password : "></asp:Label>

```

```

    </td>

```

```

    <td style="text-align: center">

```

```

        <asp:TextBox ID="TextBox2" runat="server" Font-Size="X-Large" TextMode="Password"></asp:TextBox>

```

```

    </td>

```

```

    <td> </td>

```

```

    <td> </td>

```

```

    <td> </td>

```

```

</tr>

```

```

<tr>

```

```

    <td> </td>

```

```
<td> </td>

<td> </td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td style="text-align: center">

    <asp:Button ID="Button1" runat="server" BorderStyle="None" Font-Size="X-Large"
OnClick="Button1_Click" Text="Log In" />

</td>

<td> </td>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

<td>

    <asp:Label ID="Label4" runat="server" Font-Size="X-Large"></asp:Label>

</td>

<td> </td>

<td> </td>

<td> </td>

</tr>

</table>

</div>

</form>

</body>
```



&lt;/html&gt;

---

**Q.5) Give the anatomy of .ASPX page & also explain with an example how to create a web page using ASP .Net.(13M)(W-17)(W-16)(W-18)**

---

ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web applications for PC, as well as mobile devices. ASP.NET works on top of the HTTP protocol, and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation. ASP.NET is a part of Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework.

The ASP.NET application codes can be written in any of the following languages:

- C#
- Visual Basic.Net
- Jscript
- J#

ASP.NET is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages.

#### ASP.NET Web Forms Model

ASP.NET web forms extend the event-driven model of interaction to the web applications. The browser submits a web form to the web server and the server returns a full markup page or HTML page in response.

All client side user activities are forwarded to the server for stateful processing. The server processes the output of the client actions and triggers the reactions.

Now, HTTP is a stateless protocol. ASP.NET framework helps in storing the information regarding the state of the application, which consists of:

- Page state
- Session state

The page state is the state of the client, i.e., the content of various input fields in the web form. The session state is the collective information obtained from various pages the user visited and worked with, i.e., the overall session state. To clear the concept, let us take an example of a shopping cart.

User adds items to a shopping cart. Items are selected from a page, say the items page, and the total collected items and price are shown on a different page, say the cart page. Only HTTP cannot keep track of all the information coming from various pages. ASP.NET session state and server side infrastructure keeps track of the information collected globally over a session.

The ASP.NET runtime carries the page state to and from the server across page requests while generating ASP.NET runtime codes, and incorporates the state of the server side components in hidden fields.

This way, the server becomes aware of the overall application state and operates in a two-tiered connected way.

### The ASP.NET Component Model

The ASP.NET component model provides various building blocks of ASP.NET pages. Basically it is an object model, which describes:

- Server side counterparts of almost all HTML elements or tags, such as <form> and <input>.
- Server controls, which help in developing complex user-interface. For example, the Calendar control or the Gridview control.

ASP.NET is a technology, which works on the .Net framework that contains all web-related functionalities. The .Net framework is made of an object-oriented hierarchy. An ASP.NET web application is made of pages. When a user requests an ASP.NET page, the IIS delegates the processing of the page to the ASP.NET runtime system.

The ASP.NET runtime transforms the .aspx page into an instance of a class, which inherits from the base class page of the .Net framework. Therefore, each ASP.NET page is an object and all its components i.e., the server-side controls are also objects.

### Components of .Net Framework 3.5

Before going to the next session on Visual Studio.Net, let us go through at the various components of the .Net framework 3.5. The following table describes the components of the .Net framework 3.5 and the job they perform:

#### Components and their Description

##### (1) Common Language Runtime or CLR

It performs memory management, exception handling, debugging, security checking, thread

execution, code execution, code safety, verification, and compilation. The code that is directly managed by the CLR is called the managed code. When the managed code is compiled, the compiler converts the source code into a CPU independent intermediate language (IL) code. A Just In Time(JIT) compiler compiles the IL code into native code, which is CPU specific.

## **(2) .Net Framework Class Library**

It contains a huge library of reusable types. classes, interfaces, structures, and enumerated values, which are collectively called types.

## **(3) Common Language Specification**

It contains the specifications for the .Net supported languages and implementation of language integration.

## **(4) Common Type System**

It provides guidelines for declaring, using, and managing types at runtime, and cross-language communication.

## **(5) Metadata and Assemblies**

Metadata is the binary information describing the program, which is either stored in a portable executable file (PE) or in the memory. Assembly is a logical unit consisting of the assembly manifest, type metadata, IL code, and a set of resources like image files.

## **(6) Windows Forms**

Windows Forms contain the graphical representation of any window displayed in the application.

**(7) ASP.NET and ASP.NET AJAX**

ASP.NET is the web development model and AJAX is an extension of ASP.NET for developing and implementing AJAX functionality. ASP.NET AJAX contains the components that allow the developer to update data on a website without a complete reload of the page.

**(8) ADO.NET**

It is the technology used for working with data and databases. It provides access to data sources like SQL server, OLE DB, XML etc. The ADO.NET allows connection to data sources for retrieving, manipulating, and updating data.

**(9) Windows Workflow Foundation (WF)**

It helps in building workflow-based applications in Windows. It contains activities, workflow runtime, workflow designer, and a rules engine.

**(10) Windows Presentation Foundation**

It provides a separation between the user interface and the business logic. It helps in developing visually stunning interfaces using documents, media, two and three dimensional graphics, animations, and more.

**(11) Windows Communication Foundation (WCF)**

It is the technology used for building and executing connected systems.

**(12) Windows CardSpace**

It provides safety for accessing resources and sharing personal information on the internet.

### (13) LINQ

It imparts data querying capabilities to .Net languages using a syntax which is similar to the traditional query language SQL.

### Create a web page using ASP .Net

Tasks illustrated in this walkthrough include:

- Creating a file system Web Forms application project.
- Familiarizing yourself with Visual Studio.
- Creating an ASP.NET page.
- Adding controls.
- Adding event handlers.
- Running and testing a page from Visual Studio.

### Prerequisites

In order to complete this walkthrough, you will need:

- Microsoft Visual Studio 2013 or Microsoft Visual Studio Express 2013 for Web. The .NET Framework is installed automatically.

#### **Note**

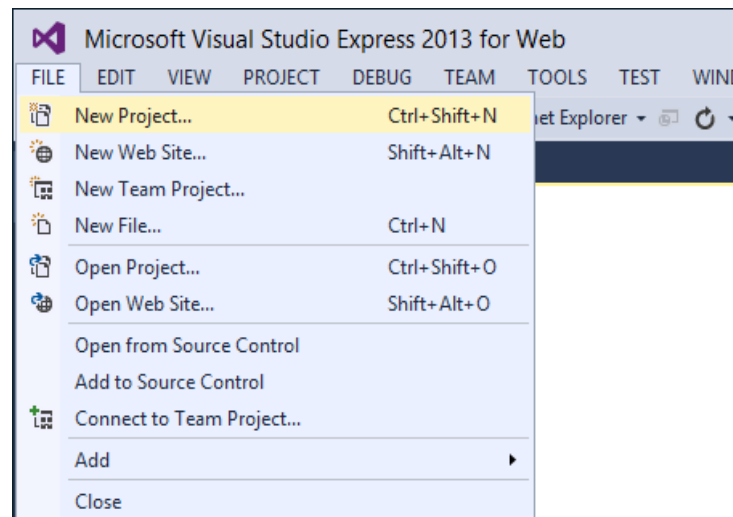
Microsoft Visual Studio 2013 and Microsoft Visual Studio Express 2013 for Web will often be referred to as Visual Studio throughout this tutorial series.

If you are using Visual Studio, this walkthrough assumes that you selected the **Web Development** collection of settings the first time that you started Visual Studio. For more information.

In this part of the walkthrough, you will create a Web application project and add a new page to it. You will also add HTML text and run the page in your browser.

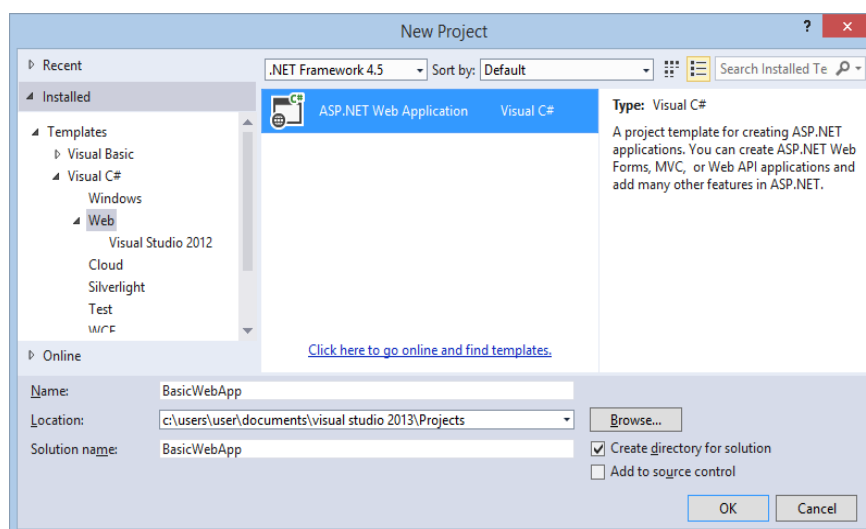
### To create a Web application project

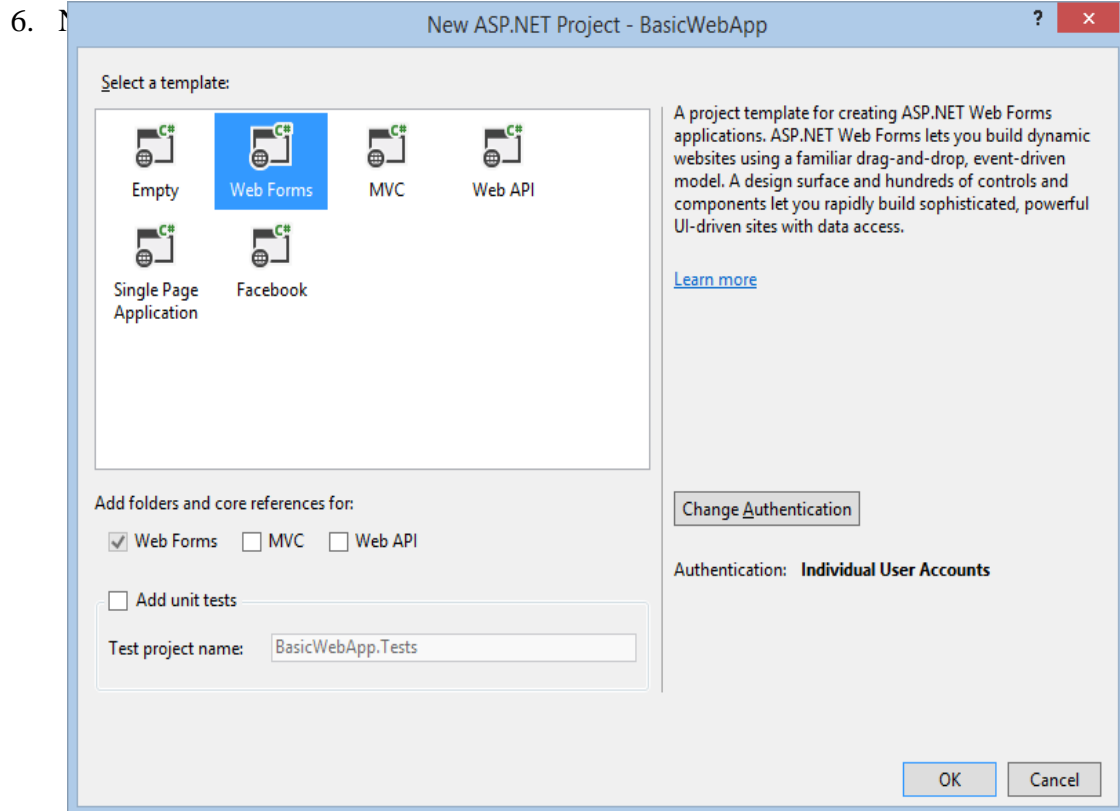
1. Open Microsoft Visual Studio.
2. On the **File** menu, select **New Project**.



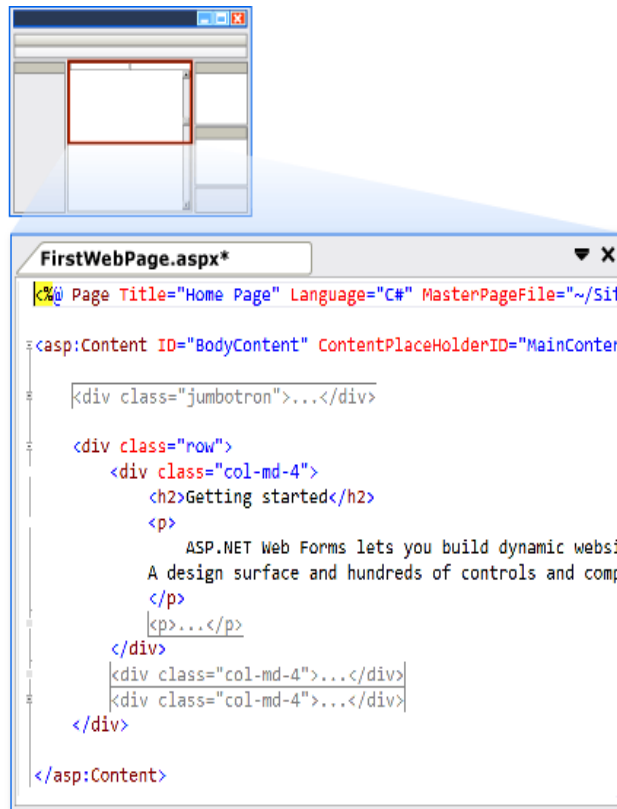
The **New Project** dialog box appears.

3. Select the **Templates -> Visual C# -> Web** templates group on the left.
4. Choose the **ASP.NET Web Application** template in the center column.
5. Name your project **BasicWebApp** and click the **OK** button.





Visual Studio creates a new project that includes prebuilt functionality based on the Web Forms template. It not only provides you with a *Home.aspx* page, an *About.aspx* page, a *Contact.aspx* page, but also includes membership functionality that registers users and saves their credentials so that they can log in to your website. When a new page is created, by default Visual Studio displays the page in **Source** view, where you can see the page's HTML elements. The following illustration shows what you would see in **Source** view if you created a new Web page named *BasicWebApp.aspx*.

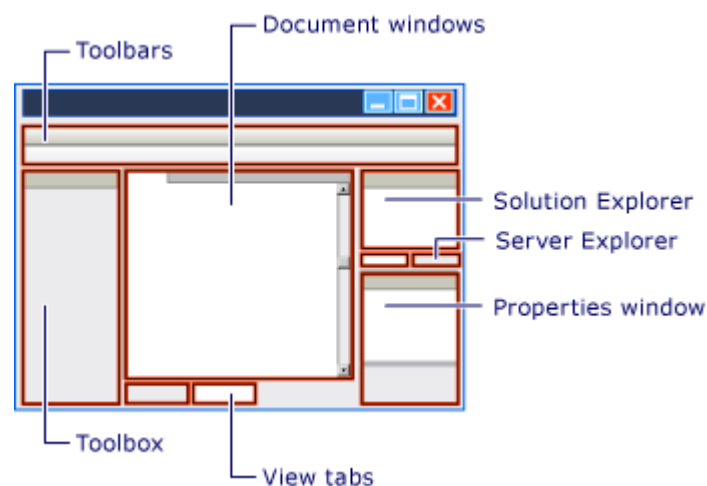


## A Tour of the Visual Studio Web Development Environment

Before you proceed by modifying the page, it is useful to familiarize yourself with the Visual Studio development environment. The following illustration shows you the windows and tools that are available in Visual Studio and Visual Studio Express for Web.

This diagram shows default windows and window locations. The **View** menu allows you to display additional windows, and to rearrange and resize windows to suit your preferences. If changes have already been made to the window arrangement, what you see will not match the illustration.

The Visual Studio environment



**Familiarize yourself with the Web designer**



Examine the above illustration and match the text to the following list, which describes the most commonly used windows and tools. (Not all windows and tools that you see are listed here, only those marked in the preceding illustration.)

- Toolbars. Provide commands for formatting text, finding text, and so on. Some toolbars are available only when you are working in **Design** view.
- **Solution Explorer** window. Displays the files and folders in your Web application.
- Document window. Displays the documents you are working on in tabbed windows. You can switch between documents by clicking tabs.
- **Properties** window. Allows you to change settings for the page, HTML elements, controls, and other objects.
- View tabs. Present you with different views of the same document. **Design** view is a near-WYSIWYG editing surface. **Source** view is the HTML editor for the page. **Split** view displays both the **Design** view and the **Source** view for the document. You will work with the **Design** and **Source** views later in this walkthrough. If you prefer to open Web pages in **Design** view, on the **Tools** menu, click **Options**, select the **HTML Designer** node, and change the **Start Pages In** option.
- **ToolBox**. Provides controls and HTML elements that you can drag onto your page. **Toolbox** elements are grouped by common function.
- **Server Explorer**. Displays database connections. If Server Explorer is not visible, on the View menu, click Server Explorer.

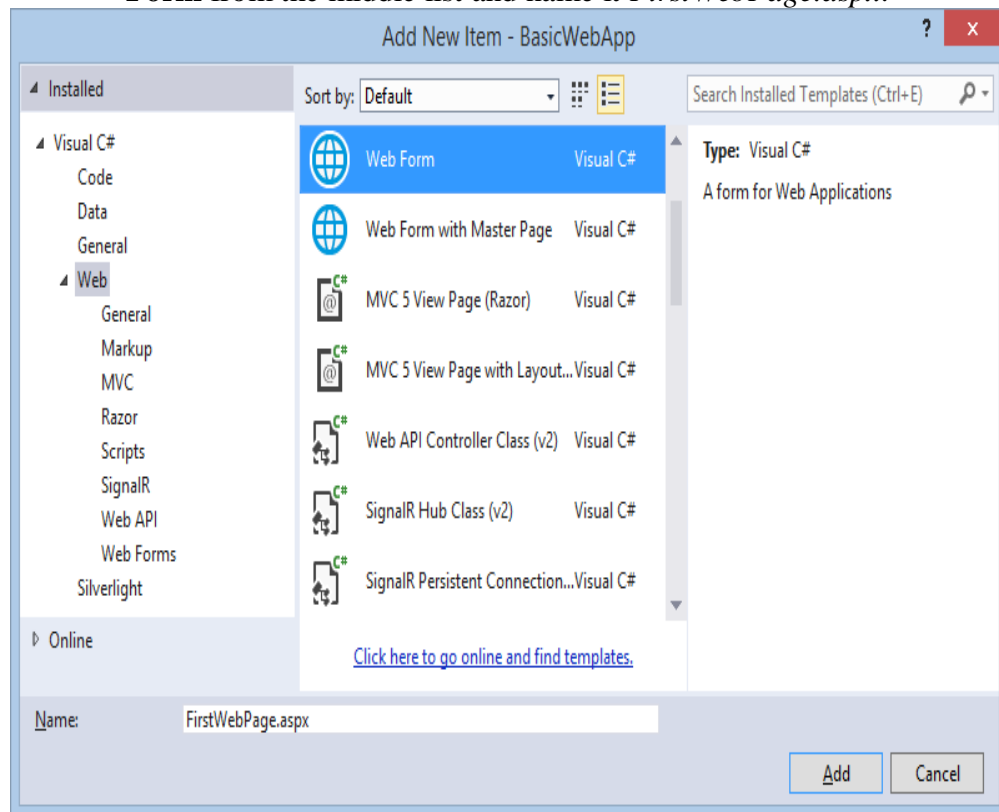
### Creating a new ASP.NET Web Forms Page

When you create a new Web Forms application using the **ASP.NET Web Application** project template, Visual Studio adds an ASP.NET page (Web Forms page) named *Default.aspx*, as well as several other files and folders. You can use the *Default.aspx* page as the home page for your Web application. However, for this walkthrough, you will create and work with a new page.

#### To add a page to the Web application

1. Close the *Default.aspx* page. To do this, click the tab that displays the file name and then click the close option.
2. In **Solution Explorer**, right-click the Web application name (in this tutorial the application name is **BasicWebSite**), and then click **Add -> New Item**. The **Add New Item** dialog box is displayed.

3. Select the **Visual C# -> Web** templates group on the left. Then, select **Web Form** from the middle list and name it *FirstWebPage.aspx*.



4. Click **Add** to add the web page to your project. Visual Studio creates the new page and opens it.

### Adding HTML to the Page

In this part of the walkthrough, you will add some static text to the page.

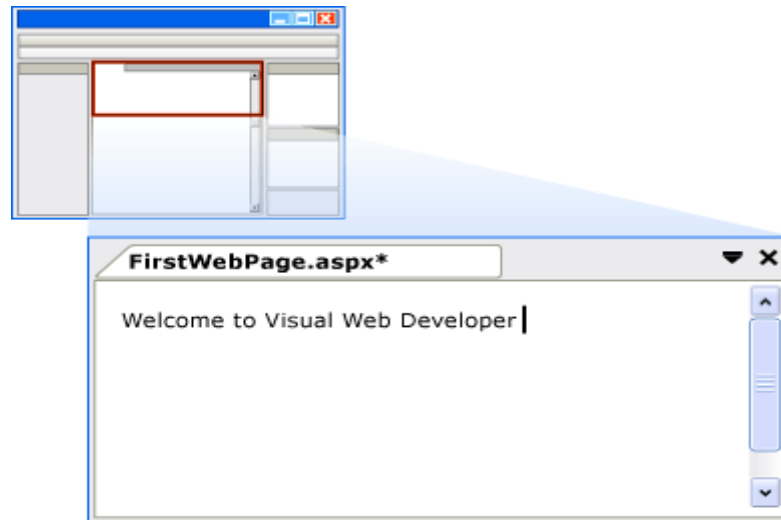
#### To add text to the page

1. At the bottom of the document window, click the **Design** tab to switch to **Design** view.

Design view displays the current page in a WYSIWYG-like way. At this point, you do not have any text or controls on the page, so the page is blank except for a dashed line that outlines a rectangle. This rectangle represents a **div** element on the page.

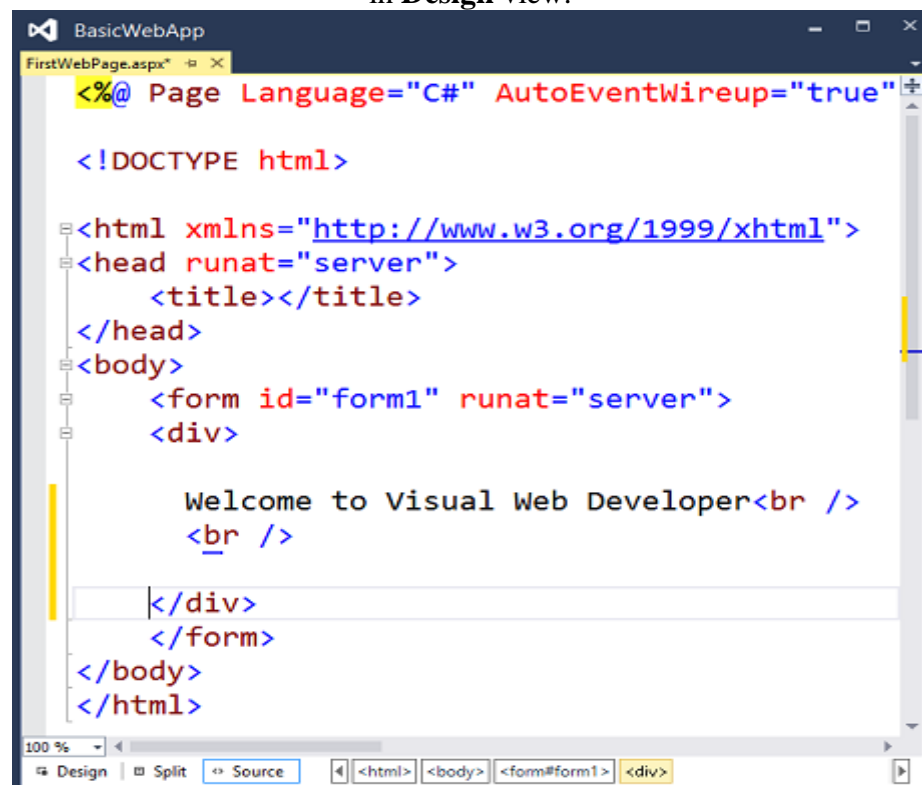
2. Click inside the rectangle that is outlined by a dashed line.
3. Type **Welcome to Visual Web Developer** and press **ENTER** twice.

The following illustration shows the text you typed in **Design** view.



4. Switch to **Source** view.

You can see the HTML in **Source** view that you created when you typed in **Design** view.



## Running the Page

Before you proceed by adding controls to the page, you can first run it.

### To run the page

1. In **Solution Explorer**, right-click *FirstWebPage.aspx* and select **Set as Start Page**.
2. Press **CTRL+F5** to run the page.

The page is displayed in the browser. Although the page you created has a file-name extension of *.aspx*, it currently runs like any HTML page.

To display a page in the browser you can also right-click the page in **Solution Explorer** and select **View in Browser**.

3. Close the browser to stop the Web application.

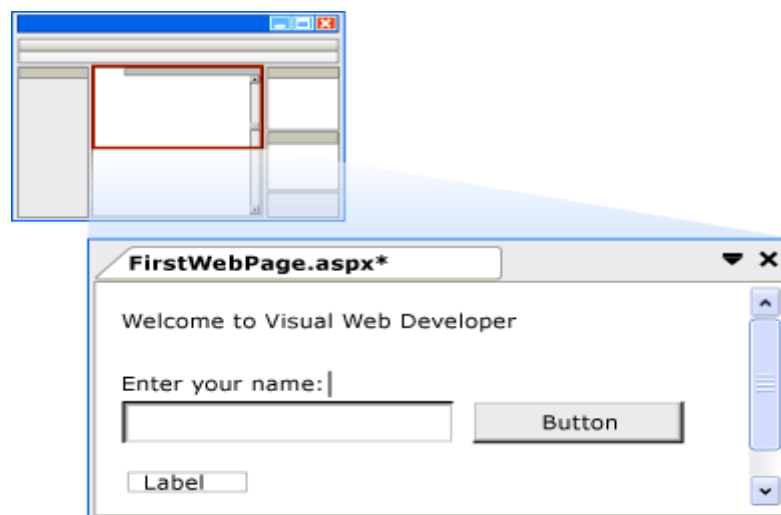
### Adding and Programming Controls

You will now add server controls to the page. Server controls, such as buttons, labels, text boxes, and other familiar controls, provide typical form-processing capabilities for your Web Forms pages. However, you can program the controls with code that runs on the server, rather than the client.

#### To add controls to the page

1. Click the **Design** tab to switch to **Design** view.
2. Put the insertion point at the end of the **Welcome to Visual Web Developer** text and press **ENTER** five or more times to make some room in the **div** element box.
3. In the **Toolbox**, expand the **Standard** group if it is not already expanded. Note that you may need to expand the **Toolbox** window on the left to view it.
4. Drag a **TextBox** control onto the page and drop it in the middle of the **div** element box that has **Welcome to Visual Web Developer** in the first line.
5. Drag a **Button** control onto the page and drop it to the right of the **TextBox** control.
6. Drag a **Label** control onto the page and drop it on a separate line below the **Button** control.
7. Put the insertion point above the **TextBox** control, and then type **Enter your name: .**

This static HTML text is the caption for the **TextBox** control. You can mix static HTML and server controls on the same page. The following illustration shows how the three controls appear in **Design** view.



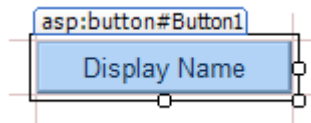
### Setting Control Properties

Visual Studio offers you various ways to set the properties of controls on the page. In this part of the walkthrough, you will set properties in both **Design** view and **Source** view.

#### To set control properties

1. First, display the **Properties** windows by selecting from the **View** menu -> **Other Windows -> Properties Window**. You could alternatively select **F4** to display the **Properties** window.

2. Select the Button control, and then in the **Properties** window, set the value of **Text** to **Display Name**. The text you entered appears on the button in the designer, as shown in the following illustration.



3. Switch to **Source** view.

**Source** view displays the HTML for the page, including the elements that Visual Studio has created for the server controls. Controls are declared using HTML-like syntax, except that the tags use the prefix **asp:** and include the attribute **runat="server"**.

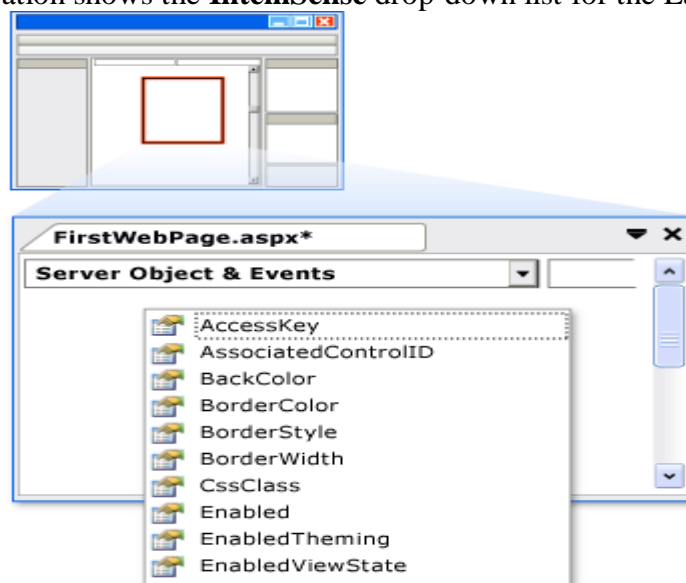
Control properties are declared as attributes. For example, when you set the **Text** property for the Button control, in step 1, you were actually setting the **Text** attribute in the control's markup.

#### Note

All the controls are inside a **form** element, which also has the attribute **runat="server"**. The **runat="server"** attribute and the **asp:** prefix for control tags mark the controls so that they are processed by ASP.NET on the server when the page runs. Code outside of **<form runat="server">** and **<script runat="server">** elements is sent unchanged to the browser, which is why the ASP.NET code must be inside an element whose opening tag contains the **runat="server"** attribute.

4. Next, you will add an additional property to the Label control. Put the insertion point directly after **asp:Label** in the **<asp:Label>** tag, and then press **SPACEBAR**.

A drop-down list appears that displays the list of available properties you can set for a Label control. This feature, referred to as **IntelliSense**, helps you in **Source** view with the syntax of server controls, HTML elements, and other items on the page. The following illustration shows the **IntelliSense** drop-down list for the Label control.



5. Select **ForeColor** and then type an equal sign.

IntelliSense displays a list of colors.

**Note**

You can display an **IntelliSense** drop-down list at any time by pressing **CTRL+J** when viewing code.

6. Select a color for the **Label** control's text. Make sure you select a color that is dark enough to read against a white background.

The **ForeColor** attribute is completed with the color that you have selected, including the closing quotation mark.

### Programming the Button Control

For this walkthrough, you will write code that reads the name that the user enters into the text box and then displays the name in the Label control.

#### Add a default button event handler

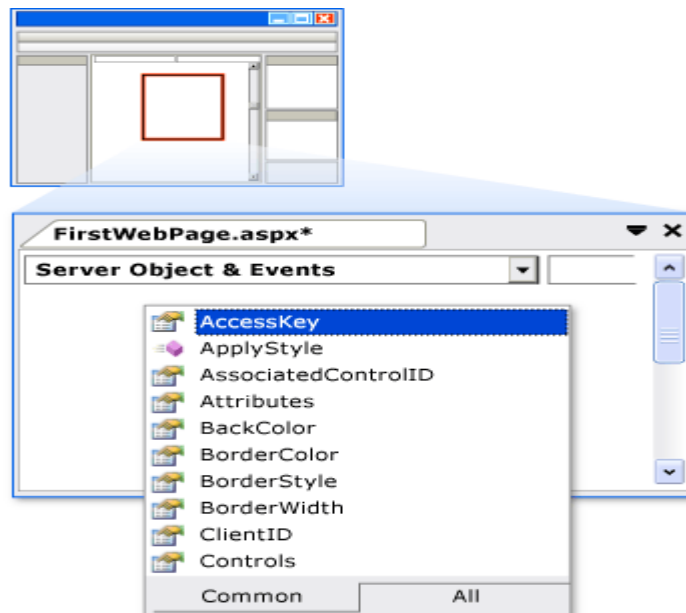
1. Switch to **Design** view.
2. Double-click the Button control.

By default, Visual Studio switches to a code-behind file and creates a skeleton event handler for the Button control's default event, the Click event. The code-behind file separates your UI markup (such as HTML) from your server code (such as C#). The cursor is positioned to added code for this event handler.

**Note**

Double-clicking a control in **Design** view is just one of several ways you can create event handlers.

3. Inside the **Button1\_Click** event handler, type **Label1** followed by a period (.). When you type the period after the **ID** of the label (**Label1**), Visual Studio displays a list of available members for the Label control, as shown in the following illustration. A member commonly a property, method, or event.



4. Finish the **Click** event handler for the button so that it reads as shown in the following code example.

C#Copy

```
protected void Button1_Click(object sender, System.EventArgs e)
{
    Label1.Text = TextBox1.Text + ", welcome to Visual Studio!";
}
```

VBCopy

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    Label1.Text = Textbox1.Text & ", welcome to Visual Studio!"
End Sub
```

5. Switch back to viewing the **Source** view of your HTML markup by right-clicking *FirstWebPage.aspx* in the **Solution Explorer** and selecting **View Markup**.
6. Scroll to the **<asp:Button>** element. Note that the **<asp:Button>** element now has the attribute **onclick="Button1\_Click"**.

Event handler methods can have any name; the name you see is the default name created by Visual Studio. The important point is that the name used for the **OnClick** attribute in the HTML must match the name of a method defined in the code-behind.

### Running the Page

You can now test the server controls on the page.

#### To run the page

1. Press **CTRL+F5** to run the page in the browser. If an error occurs, recheck the steps above.
2. Enter a name into the text box and click the **Display Name** button.

The name you entered is displayed in the Label control. Note that when you click the button, the page is posted to the Web server. ASP.NET then recreates the page, runs your code (in this case, the Button control's Click event handler runs), and then sends the new page to the browser. If you watch the status bar in the browser, you can see that the page is making a round trip to the Web server each time you click the button.

3. In the browser, view the source of the page you are running by right-clicking on the page and selecting **View source**.

In the page source code, you see HTML without any server code. Specifically, you do not see the **<asp:>** elements that you were working with in **Source** view. When the page runs, ASP.NET processes the server controls and renders HTML elements to the page that perform the functions that represent the control. For example, the **<asp:Button>** control is rendered as the HTML **<input type="submit">** element.

4. Close the browser.

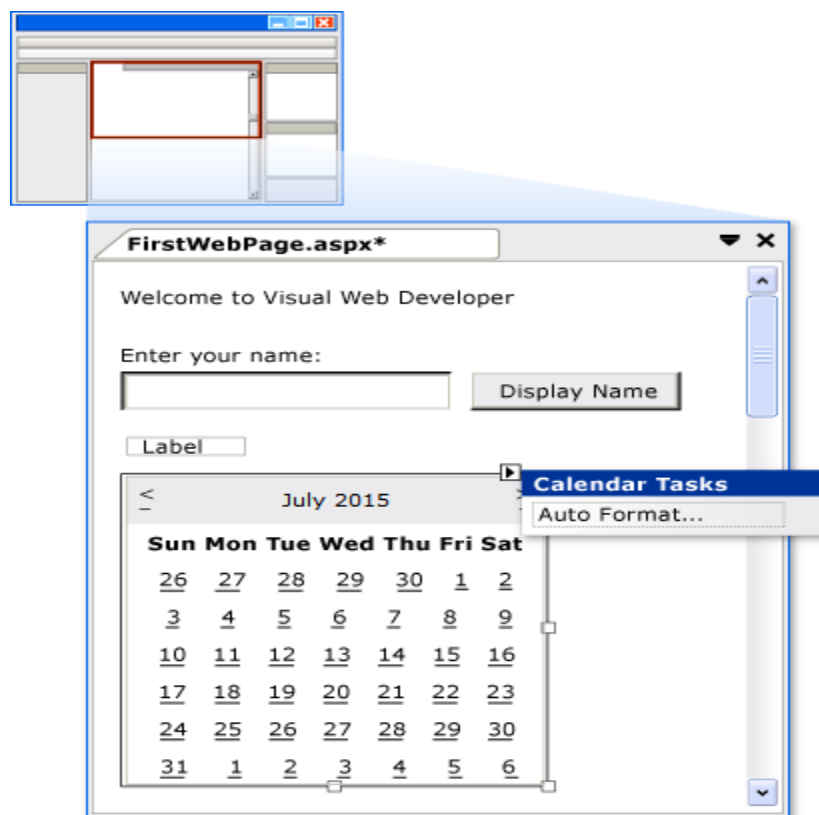
## Working with Additional Controls

In this part of the walkthrough, you will work with the **Calendar control**, which displays dates a month at a time. The **Calendar control** is a more complex control than the button, text box, and label you have been working with and illustrates some further capabilities of server controls.

### To add a Calendar control

1. In Visual Studio, switch to **Design** view.
2. From the **Standard** section of the **Toolbox**, drag a Calendar control onto the page and drop it below the **div** element that contains the other controls.

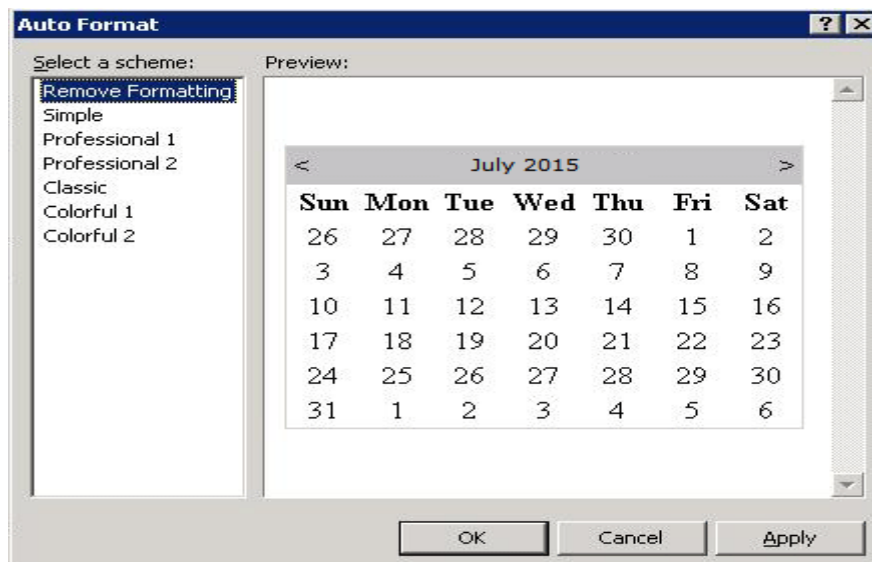
The calendar's smart tag panel is displayed. The panel displays commands that make it easy for you to perform the most common tasks for the selected control. The following illustration shows the Calendar control as rendered in **Design** view.



3. In the smart tag panel, choose **Auto Format**.

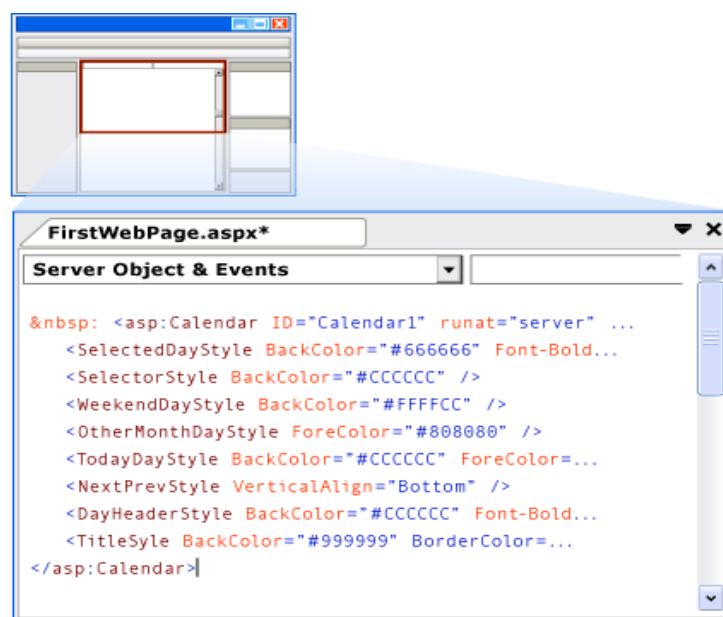
The **Auto Format** dialog box is displayed, which allows you to select a formatting scheme for the calendar. The following illustration shows the **Auto Format** dialog box for the Calendar control.





4. From the **Select a scheme** list, select **Simple** and then click **OK**.
5. Switch to **Source** view.

You can see the `<asp:Calendar>` element. This element is much longer than the elements for the simple controls you created earlier. It also includes subelements, such as `<WeekEndDayStyle>`, which represent various formatting settings. The following illustration shows the Calendar control in **Source** view. (The exact markup that you see in **Source** view might differ slightly from the illustration.)



## Programming the Calendar Control

In this section, you will program the Calendar control to display the currently selected date.

### To program the Calendar control

1. In **Design** view, double-click the Calendar control.

A new event handler is created and displayed in the code-behind file named *FirstWebPage.aspx.cs*.

2. Finish the SelectionChanged event handler with the following code.

C#Copy

```
protected void Calendar1_SelectionChanged(object sender, System.EventArgs e)
{
    Label1.Text = Calendar1.SelectedDate.ToLongDateString();
}
```

VBCopy

```
Protected Sub Calendar1_SelectionChanged(ByVal sender As Object, ByVal e As
System.EventArgs)
    Label1.Text = Calendar1.SelectedDate.ToLongDateString()
End Sub
```

The above code sets the text of the label control to the selected date of the calendar control.

### Running the Page

You can now test the calendar.

#### To run the page

1. Press **CTRL+F5** to run the page in the browser.
2. Click a date in the calendar.

The date you clicked is displayed in the Label control.

3. In the browser, view the source code for the page.

Note that the Calendar control has been rendered to the page as a **table**, with each day as a **td** element.

4. Close the browser.

### Q.7) Why there is need of ADO.Net? Explain how to use ADO.Net in any web application.(8M)(W-17)

ADO.NET provides a comprehensive caching data model for marshalling data between applications and services with facilities to optimistically update the original data sources. This enables developer to begin with XML while leveraging existing skills with SQL and the relational model.

Although the ADO.NET model is different from the existing ADO model, the same basic concepts include provider, connection and command objects. By combining the continued use of SQL with similar basic concepts, current ADO developers should be able to migrate to ADO.NET over a reasonable period of time.

## Create a simple data application by using ADO.NET

When you create an application that manipulates data in a database, stored procedures. By following this topic, you can discover how to interact with a database from within a simple Windows Forms "forms over data" application by using Visual C# or Visual Basic and ADO.NET. All .NET data technologies—including datasets, LINQ to SQL, and Entity Framework—ultimately perform steps that are very similar to those shown in this article.

This article demonstrates a simple way to get data out of a database in a fast manner. If your application needs to modify data in non-trivial ways and update the database, you should consider using Entity Framework and using data binding to automatically sync user interface controls to changes in the underlying data.

### Important

you perform basic tasks such as defining connection strings, inserting data, and running. To keep the code simple, it doesn't include production-ready exception handling.

### Prerequisites

To create the application, you'll need:

- Visual Studio.
- SQL Server Express LocalDB. If you don't have SQL Server Express LocalDB,

Create the sample database by following these steps:

1. In Visual Studio, open the **Server Explorer** window.
2. Right-click on **Data Connections** and choose **Create New SQL Server Database**.
3. In the **Server name** text box, enter **(localdb)\mssqllocaldb**.
4. In the **New database name** text box, enter **Sales**, then choose **OK**.

The empty **Sales** database is created and added to the Data Connections node in Server Explorer.

5. Right-click on the **Sales** data connection and select **New Query**.

A query editor window opens.

6. Copy the Sales Transact-SQL script to your clipboard.
7. Paste the T-SQL script into the query editor, and then choose the **Execute** button.

After a short time, the query finishes running and the database objects are created. The database contains two tables: Customer and Orders. These tables contain no data initially, but you can add data when you run the application that you'll create. The database also contains four simple stored procedures.

### Create the forms and add controls

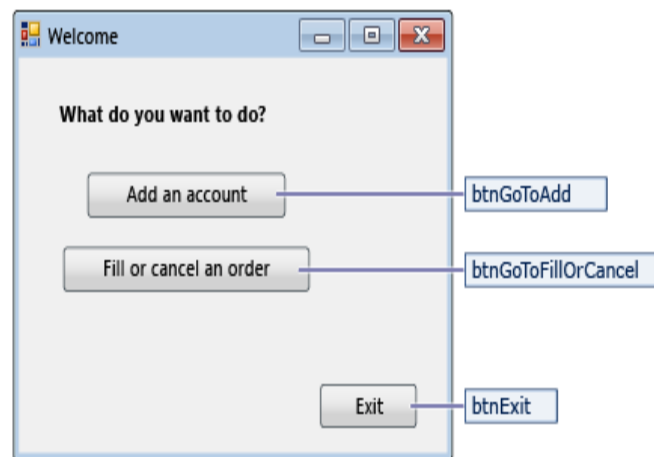
1. Create a project for a Windows Forms application, and then name it **SimpleDataApp**. Visual Studio creates the project and several files, including an empty Windows form that's named **Form1**.
2. Add two Windows forms to your project so that it has three forms, and then give them the following names:
  - **Navigation**
  - **NewCustomer**
  - **FillOrCancel**

For each form, add the text boxes, buttons, and other controls that appear in the following illustrations. For each control, set the properties that the tables describe.

### Note

The group box and the label controls add clarity but aren't used in the code.

## Navigation form



### Controls for the Navigation form

### Properties

Button

Name = btnGoToAdd

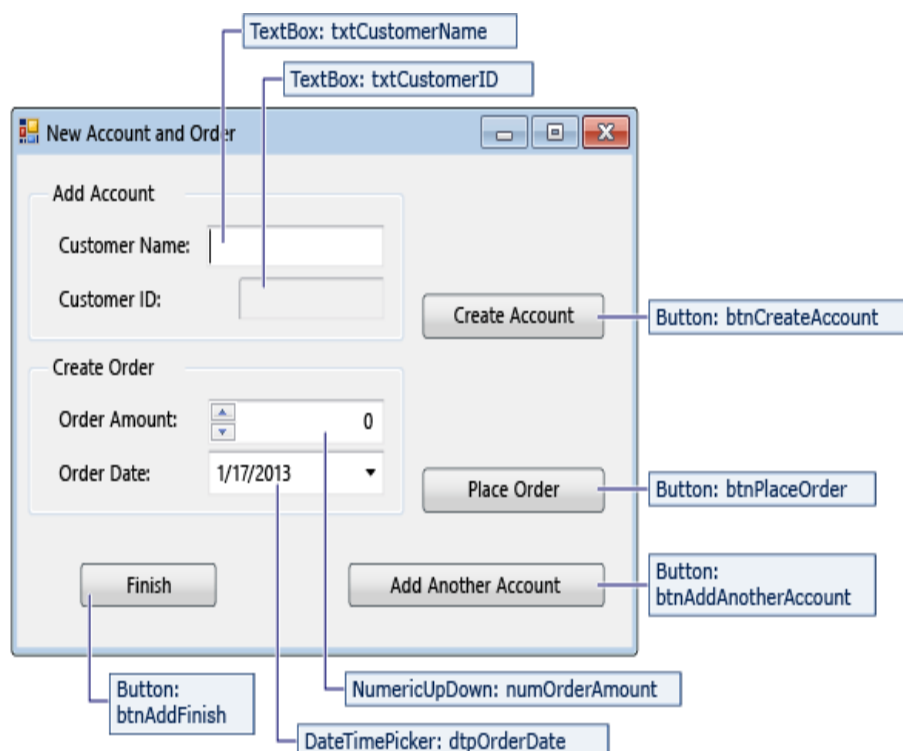
Button

Name = btnGoToFillOrCancel

Button

Name = btnExit

## NewCustomer form



<b>Controls for the NewCustomer form</b>	<b>Properties</b>
TextBox	Name = txtCustomerName
TextBox	Name = txtCustomerID  Readonly = True
Button	Name = btnCreateAccount
NumericUpDown	DecimalPlaces = 0  Maximum = 5000  Name = numOrderAmount
DateTimePicker	Format = Short  Name = dtpOrderDate
Button	Name = btnPlaceOrder
Button	Name = btnAddAnotherAccount
Button	Name = btnAddFinish

**FillOrCancel form**

TextBox: txtOrderID

Order ID:  Find Order Button: btnFindByOrderID

If filling an order, specify filled date 1/17/2013 DateTimePicker: dtpFillDate

DataGridView: dgvCustomerOrders

Cancel Order Fill Order Finish Button: btnFinishUpdates

Button: btnCancelOrder Button: btnFillOrder

Controls for the FillOrCancel form	Properties
TextBox	Name = txtOrderID
Button	Name = btnFindByOrderID
DateTimePicker	Format = Short Name = dtpFillDate
DataGridView	Name = dgvCustomerOrders ReadOnly = True RowHeadersVisible = False
Button	Name = btnCancelOrder
Button	Name = btnFillOrder

Controls for the FillOrCancel form	Properties
Button	Name = btnFinishUpdates

### Store the connection string

When your application tries to open a connection to the database, your application must have access to the connection string. To avoid entering the string manually on each form, store the string in the *App.config* file in your project, and create a method that returns the string when the method is called from any form in your application.

You can find the connection string by right-clicking on the **Sales** data connection in **Server Explorer** and choosing **Properties**. Locate the **ConnectionString** property, then use **Ctrl+A**, **Ctrl+C** to select and copy the string to the clipboard.

1. If you're using C#, in **Solution Explorer**, expand the **Properties** node under the project, and then open the **Settings.settings** file. If you're using Visual Basic, in **Solution Explorer**, click **Show All Files**, expand the **My Project** node, and then open the **Settings.settings** file.
2. In the **Name** column, enter `connString`.
3. In the **Type** list, select **(Connection String)**.
4. In the **Scope** list, select **Application**.
5. In the **Value** column, enter your connection string (without any outside quotes), and then save your changes.

### Note

In a real application, you should store the connection string securely, as described in **Connection strings and configuration files**.

### Write the code for the forms

This section contains brief overviews of what each form does. It also provides the code that defines the underlying logic when a button on the form is clicked.

### Navigation form

The Navigation form opens when you run the application. The **Add an account** button opens the NewCustomer form. The **Fill or cancel orders** button opens the FillOrCancel form. The **Exit** button closes the application.

### *Make the Navigation form the startup form*

If you're using C#, in **Solution Explorer**, open **Program.cs**, and then change the `Application.Run` line to this: `Application.Run(new Navigation());`

If you're using Visual Basic, in **Solution Explorer**, open the **Properties** window, select the **Application** tab, and then select **SimpleDataApp.Navigation** in the **Startup form** list.

**Create auto-generated event handlers**

Double-click the three buttons on the Navigation form to create empty event handler methods. Double-clicking the buttons also adds auto-generated code in the Designer code file that enables a button click to raise an event.

**Add code for the Navigation form logic**

In the code page for the Navigation form, complete the method bodies for the three button click event handlers as shown in the following code.

C#Copy

```

/// <summary>
/// Opens the NewCustomer form as a dialog box,
/// which returns focus to the calling form when it is closed.
/// </summary>
private void btnGoToAdd_Click(object sender, EventArgs e)
{
    Form frm = new NewCustomer();
    frm.Show();
}

/// <summary>
/// Opens the FillOrCancel form as a dialog box.
/// </summary>
private void btnGoToFillOrCancel_Click(object sender, EventArgs e)
{
    Form frm = new FillOrCancel();
    frm.ShowDialog();
}

/// <summary>
/// Closes the application (not just the Navigation form).
/// </summary>
private void btnExit_Click(object sender, EventArgs e)
{
    this.Close();
}

```

**NewCustomer form**

When you enter a customer name and then select the **Create Account** button, the NewCustomer form creates a customer account, and SQL Server returns an IDENTITY value as the new customer ID. You can then place an order for the new account by specifying an amount and an order date and selecting the **Place Order** button.

**Create auto-generated event handlers**

Create an empty Click event handler for each button on the NewCustomer form by double-clicking on each of the four buttons. Double-clicking the buttons also adds auto-generated code in the Designer code file that enables a button click to raise an event.



***Add code for the NewCustomer form logic***

To complete the NewCustomer form logic, follow these steps.

1. Bring the `System.Data.SqlClient` namespace into scope so that you don't have to fully qualify the names of its members.

C#Copy

```
using System.Data.SqlClient;
```

2. Add some variables and helper methods to the class as shown in the following code.

C#Copy

```
// Storage for IDENTITY values returned from database.
```

```
private int parsedCustomerID;
```

```
private int orderID;
```

```
/// <summary>
```

```
/// Verifies that the customer name text box is not empty.
```

```
/// </summary>
```

```
private bool IsCustomerNameValid()
```

```
{
```

```
    if (txtCustomerName.Text == "")
```

```
    {
```

```
        MessageBox.Show("Please enter a name.");
```

```
        return false;
```

```
    }
```

```
    else
```

```
    {
```

```
        return true;
```

```
    }
```

```
}
```

```
/// <summary>
```

```
/// Verifies that a customer ID and order amount have been provided.
```

```
/// </summary>
```

```
private bool IsOrderDataValid()
```

```
{
```

```
    // Verify that CustomerID is present.
```

```
    if (txtCustomerID.Text == "")
```

```
    {
```

```
        MessageBox.Show("Please create customer account before placing order.");
```

```
        return false;
```

```
    }
```

```
    // Verify that Amount isn't 0.
```

```
    else if ((numOrderAmount.Value < 1))
```

```

    {
        MessageBox.Show("Please specify an order amount.");
        return false;
    }
    else
    {
        // Order can be submitted.
        return true;
    }
}

```

```

/// <summary>
/// Clears the form data.
/// </summary>
private void ClearForm()
{
    txtCustomerName.Clear();
    txtCustomerID.Clear();
    dtpOrderDate.Value = DateTime.Now;
    numOrderAmount.Value = 0;
    this.parsedCustomerID = 0;
}

```

3. Complete the method bodies for the four button click event handlers as shown in the following code.

C#Copy

```

/// <summary>
/// Creates a new customer by calling the Sales.uspNewCustomer stored procedure.
/// </summary>
private void btnCreateAccount_Click(object sender, EventArgs e)
{
    if (IsCustomerNameValid())
    {
        // Create the connection.
        using (SqlConnection connection = new
SqlConnection(Properties.Settings.Default.connString))
        {
            // Create a SqlCommand, and identify it as a stored procedure.
            using (SqlCommand sqlCommand = new
SqlCommand("Sales.uspNewCustomer", connection))
            {
                sqlCommand.CommandType = CommandType.StoredProcedure;

                // Add input parameter for the stored procedure and specify what to use as its
value.

```

```

        sqlCommand.Parameters.Add(new SqlParameter("@CustomerName",
SqlDbType.NVarChar, 40));
        sqlCommand.Parameters["@CustomerName"].Value =
txtCustomerName.Text;

        // Add the output parameter.
        sqlCommand.Parameters.Add(new SqlParameter("@CustomerID",
SqlDbType.Int));
        sqlCommand.Parameters["@CustomerID"].Direction =
ParameterDirection.Output;

        try
        {
            connection.Open();

            // Run the stored procedure.
            sqlCommand.ExecuteNonQuery();

            // Customer ID is an IDENTITY value from the database.
            this.parsedCustomerID =
(int)sqlCommand.Parameters["@CustomerID"].Value;

            // Put the Customer ID value into the read-only text box.
            this.txtCustomerID.Text = Convert.ToString(parsedCustomerID);
        }
        catch
        {
            MessageBox.Show("Customer ID was not returned. Account could not be
created.");
        }
        finally
        {
            connection.Close();
        }
    }
}

/// <summary>
/// Calls the Sales.uspPlaceNewOrder stored procedure to place an order.
/// </summary>
private void btnPlaceOrder_Click(object sender, EventArgs e)
{
    // Ensure the required input is present.
    if (IsOrderDataValid())
    {

```

```

// Create the connection.
using (SqlConnection connection = new
SqlConnection(Properties.Settings.Default.connString))
{
    // Create SqlCommand and identify it as a stored procedure.
    using (SqlCommand sqlCommand = new
SqlCommand("Sales.uspPlaceNewOrder", connection))
    {
        sqlCommand.CommandType = CommandType.StoredProcedure;

        // Add the @CustomerID input parameter, which was obtained from
uspNewCustomer.
        sqlCommand.Parameters.Add(new SqlParameter("@CustomerID",
SqlDbType.Int));
        sqlCommand.Parameters["@CustomerID"].Value = this.parsedCustomerID;

        // Add the @OrderDate input parameter.
        sqlCommand.Parameters.Add(new SqlParameter("@OrderDate",
SqlDbType.DateTime, 8));
        sqlCommand.Parameters["@OrderDate"].Value = dtpOrderDate.Value;

        // Add the @Amount order amount input parameter.
        sqlCommand.Parameters.Add(new SqlParameter("@Amount",
SqlDbType.Int));
        sqlCommand.Parameters["@Amount"].Value = numOrderAmount.Value;

        // Add the @Status order status input parameter.
        // For a new order, the status is always O (open).
        sqlCommand.Parameters.Add(new SqlParameter("@Status",
SqlDbType.Char, 1));
        sqlCommand.Parameters["@Status"].Value = "O";

        // Add the return value for the stored procedure, which is the order ID.
        sqlCommand.Parameters.Add(new SqlParameter("@RC", SqlDbType.Int));
        sqlCommand.Parameters["@RC"].Direction =
ParameterDirection.ReturnValue;

        try
        {
            //Open connection.
            connection.Open();

            // Run the stored procedure.
            sqlCommand.ExecuteNonQuery();

            // Display the order number.
            this.orderID = (int)sqlCommand.Parameters["@RC"].Value;

```

```

        MessageBox.Show("Order number " + this.orderID + " has been
submitted.");
    }
    catch
    {
        MessageBox.Show("Order could not be placed.");
    }
    finally
    {
        connection.Close();
    }
}
}
}

/// <summary>
/// Clears the form data so another new account can be created.
/// </summary>
private void btnAddAnotherAccount_Click(object sender, EventArgs e)
{
    this.ClearForm();
}

/// <summary>
/// Closes the form/dialog box.
/// </summary>
private void btnAddFinish_Click(object sender, EventArgs e)
{
    this.Close();
}

```

### FillOrCancel form

The FillOrCancel form runs a query to return an order when you enter an order ID and then click the **Find Order** button. The returned row appears in a read-only data grid. You can mark the order as canceled (X) if you select the **Cancel Order** button, or you can mark the order as filled (F) if you select the **Fill Order** button. If you select the **Find Order** button again, the updated row appears.

#### *Create auto-generated event handlers*

Create empty Click event handlers for the four buttons on the FillOrCancel form by double-clicking the buttons. Double-clicking the buttons also adds auto-generated code in the Designer code file that enables a button click to raise an event.

#### *Add code for the FillOrCancel form logic*

To complete the FillOrCancel form logic, follow these steps.

1. Bring the following two namespaces into scope so that you don't have to fully qualify the names of their members.

C#Copy

```
using System.Data.SqlClient;
using System.Text.RegularExpressions;
```

2. Add a variable and helper method to the class as shown in the following code.

C#Copy

```
// Storage for the order ID value.
private int parsedOrderID;

/// <summary>
/// Verifies that an order ID is present and contains valid characters.
/// </summary>
private bool IsOrderIDValid()
{
    // Check for input in the Order ID text box.
    if (txtOrderID.Text == "")
    {
        MessageBox.Show("Please specify the Order ID.");
        return false;
    }

    // Check for characters other than integers.
    else if (Regex.IsMatch(txtOrderID.Text, @"^\d*$"))
    {
        // Show message and clear input.
        MessageBox.Show("Customer ID must contain only numbers.");
        txtOrderID.Clear();
        return false;
    }
    else
    {
        // Convert the text in the text box to an integer to send to the database.
        parsedOrderID = Int32.Parse(txtOrderID.Text);
        return true;
    }
}
```

3. Complete the method bodies for the four button click event handlers as shown in the following code.

C#Copy

```

/// <summary>
/// Executes a t-SQL SELECT statement to obtain order data for a specified
/// order ID, then displays it in the DataGridView on the form.
/// </summary>
private void btnFindByOrderID_Click(object sender, EventArgs e)
{
    if (IsOrderIDValid())
    {
        using (SqlConnection connection = new
SqlConnection(Properties.Settings.Default.connString))
        {
            // Define a t-SQL query string that has a parameter for orderID.
            const string sql = "SELECT * FROM Sales.Orders WHERE orderID =
@orderID";

            // Create a SqlCommand object.
            using (SqlCommand sqlCommand = new SqlCommand(sql, connection))
            {
                // Define the @orderID parameter and set its value.
                sqlCommand.Parameters.Add(new SqlParameter("@orderID",
SqlDbType.Int));
                sqlCommand.Parameters["@orderID"].Value = parsedOrderID;

                try
                {
                    connection.Open();

                    // Run the query by calling ExecuteReader().
                    using (SqlDataReader dataReader = sqlCommand.ExecuteReader())
                    {
                        // Create a data table to hold the retrieved data.
                        DataTable dataTable = new DataTable();

                        // Load the data from SqlDataReader into the data table.
                        dataTable.Load(dataReader);

                        // Display the data from the data table in the data grid view.
                        this.dgvCustomerOrders.DataSource = dataTable;

                        // Close the SqlDataReader.
                        dataReader.Close();
                    }
                }
                catch
                {
                    MessageBox.Show("The requested order could not be loaded into the
form.");

```

```

    }
    finally
    {
        // Close the connection.
        connection.Close();
    }
}
}
}

/// <summary>
/// Cancels an order by calling the Sales.uspCancelOrder
/// stored procedure on the database.
/// </summary>
private void btnCancelOrder_Click(object sender, EventArgs e)
{
    if (IsOrderIDValid())
    {
        // Create the connection.
        using (SqlConnection connection = new
SqlConnection(Properties.Settings.Default.connString))
        {
            // Create the SqlCommand object and identify it as a stored procedure.
            using (SqlCommand sqlCommand = new
SqlCommand("Sales.uspCancelOrder", connection))
            {
                sqlCommand.CommandType = CommandType.StoredProcedure;

                // Add the order ID input parameter for the stored procedure.
                sqlCommand.Parameters.Add(new SqlParameter("@orderID",
SqlDbType.Int));
                sqlCommand.Parameters["@orderID"].Value = parsedOrderID;

                try
                {
                    // Open the connection.
                    connection.Open();

                    // Run the command to execute the stored procedure.
                    sqlCommand.ExecuteNonQuery();
                }
                catch
                {
                    MessageBox.Show("The cancel operation was not completed.");
                }
            }
        }
    }
}

```



```

        {
            // Close connection.
            connection.Close();
        }
    }
}

/// <summary>
/// Fills an order by calling the Sales.uspFillOrder stored
/// procedure on the database.
/// </summary>
private void btnFillOrder_Click(object sender, EventArgs e)
{
    if (IsOrderIDValid())
    {
        // Create the connection.
        using (SqlConnection connection = new
SqlConnection(Properties.Settings.Default.connString))
        {
            // Create command and identify it as a stored procedure.
            using (SqlCommand sqlCommand = new SqlCommand("Sales.uspFillOrder",
connection))
            {
                sqlCommand.CommandType = CommandType.StoredProcedure;

                // Add the order ID input parameter for the stored procedure.
                sqlCommand.Parameters.Add(new SqlParameter("@orderID",
SqlDbType.Int));
                sqlCommand.Parameters["@orderID"].Value = parsedOrderID;

                // Add the filled date input parameter for the stored procedure.
                sqlCommand.Parameters.Add(new SqlParameter("@FilledDate",
SqlDbType.DateTime, 8));
                sqlCommand.Parameters["@FilledDate"].Value = dtpFillDate.Value;

                try
                {
                    connection.Open();

                    // Execute the stored procedure.
                    sqlCommand.ExecuteNonQuery();
                }
                catch
                {
                    MessageBox.Show("The fill operation was not completed.");
                }
            }
        }
    }
}

```

```

    }
    finally
    {
        // Close the connection.
        connection.Close();
    }
}
}
}

/// <summary>
/// Closes the form.
/// </summary>
private void btnFinishUpdates_Click(object sender, EventArgs e)
{
    this.Close();
}

```

---

**Q.8) Explain step by step how to create console application using ADO, NET? Consider any example. (7M)(W-16)**

```

class Program
{
    static void Main(string[] args)
    {
        int num1;
        int num2;
        string operand;
        float answer;

        Console.WriteLine("Please enter the first integer: ");
        num1 = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Please enter an operand (+, -, /, *): ");
        operand = Console.ReadLine();

        Console.WriteLine("Please enter the second integer: ");
        num2 = Convert.ToInt32(Console.ReadLine());

        switch (operand)
        {
            case "-":
                answer = num1 - num2;
                break;
            case "+":
                answer = num1 + num2;

```

```

        break;
    case "/":
        answer = num1 / num2;
        break;
    case "*":
        answer = num1 * num2;
        break;
    default:
        answer = 0;
        break;
    }

    Console.WriteLine(num1.ToString() + " " + operand + " " + num2.ToString() + " = " + answer.ToString());

    Console.ReadLine();
}
}

```

---

**Q.9) Write a program in C# to design calculator as console based application. (6M) (W-18)**

```

class Program
{
    static void Main(string[] args)
    {
        int num1;
        int num2;
        string operand;
        float answer;

        Console.Write("Please enter the first integer: ");
        num1 = Convert.ToInt32(Console.ReadLine());

        Console.Write("Please enter an operand (+, -, /, *): ");
        operand = Console.ReadLine();

        Console.Write("Please enter the second integer: ");
        num2 = Convert.ToInt32(Console.ReadLine());

        switch (operand)
        {
            case "-":
                answer = num1 - num2;
                break;
            case "+":
                answer = num1 + num2;
                break;
            case "/":

```

```

        answer = num1 / num2;
        break;
    case "*":
        answer = num1 * num2;
        break;
    default:
        answer = 0;
        break;
    }

    Console.WriteLine(num1.ToString() + " " + operand + " " + num2.ToString() + " = " + answer.ToString());

    Console.ReadLine();
}
}

```

---



**Tulsiramji Gaikwad-Patil College of Engineering and Technology**  
 Wardha Road, Nagpur-441 108  
 NAAC Accredited

---

*Department of Computer Science & Engineering*

---

**Semester: B.E. Eighth Semester (CBS)**  
**Subject: Clustering & Cloud Computing**  
**Unit-6 Solution**

---

**Q.1) How the cloud application deploy on to the windows Azure cloud(7M)(S-17)**

**Ans:**

#### **Deploying Application On Windows Azure Portal**

To deploy application on Microsoft Data Center you need to have a Windows Azure Account. Windows Azure is a paid service however you can start with free trial. To register for free account follow the below steps.

Register for Free Account

Step 1

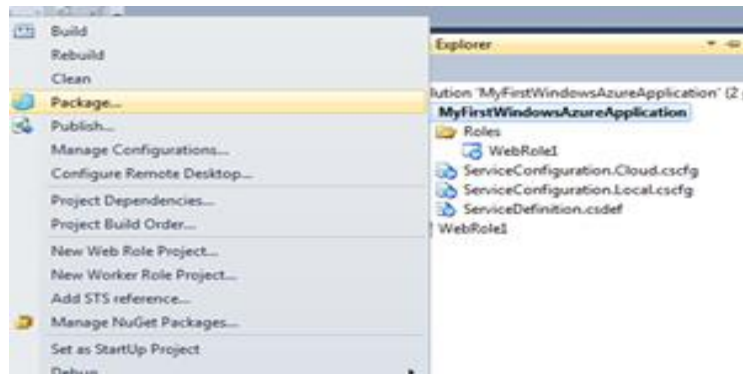
You will be asked to login using Live ID. Provide your live id and login. If you don't have live ID create one to work with Windows Azure Free Trail



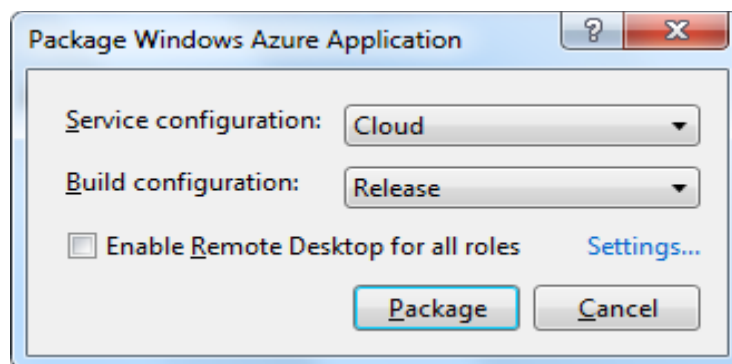
Next proceed through the screen to create free account.



After successful registration you will be getting a success registration message. After registration go back to visual studio and right click on Windows Azure Project and select Package.



Next choose Service Configuration as Cloud and Build Configuration as Release and click Package



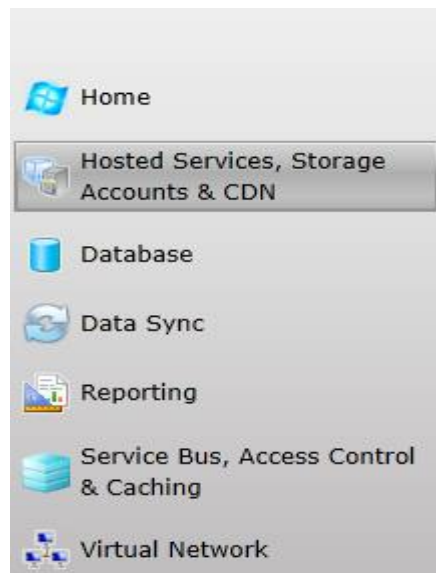
After successful package you can see Service Package File and Cloud Service Configuration file in the folder explorer. We need to upload these two files to deploy application on Microsoft Data Center.

---

SUPPORT	ACCOUNT
Contact	Subscriptions
Forums	Profile
Service Dashboard	Management Portal
FAQ	
SLA	
Legal	

You will be navigated to live login page. Provide same live id and password you used to create Free Trial. After successful authenticating you will be navigated to Management Portal.

To deploy on Microsoft Data Center, first you need to create Hosted Service. To create Hosted Service from left tab select Hosted Service, Storage, Account and CDN



In top you will get three options. Their purpose is very much clear with their name.



Click on New Hosted Service to create a Hosted service. Provide information as below to create hosted service.

**Create a New Hosted Service**

**Choose a subscription**

debugmode

**Enter a name for your service**

MyFirstService

**Enter a URL prefix for your service**

debugmodemyfirstservice .cloudapp.net

**Choose a region or affinity group**

☒ Choose a Region ☐ Create or choose an affinity group

**Deployment options**

☐ Deploy to stage environment

☒ Deploy to production environment

☐ Do not deploy

☒ Start after successful deployment

**Deployment name**

d1

Choose Subscription Name. It should be the same as your registered subscription of previous step.

Enter name of the service

Enter URL of the service. This URL need to be unique. On this URL you will be accessing the application. So this application will be used at URL debugmodemyfirstservice.cloudapp.net

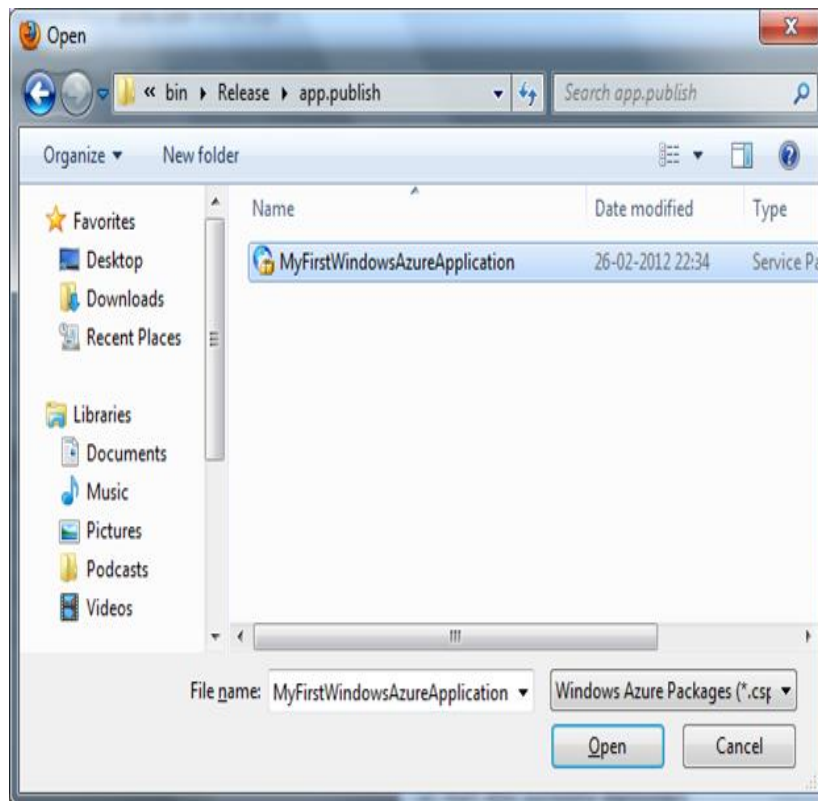
Choose a region from the drop down. In further post we will get into details of affinity group.

In Deployment option choose, Deploy to production environment.

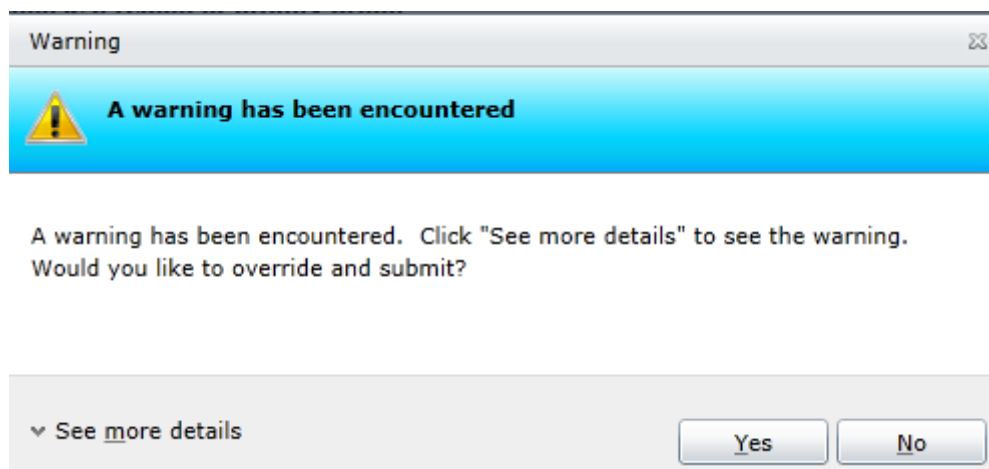
Give a deployment name.

Next to upload package select browse locally. On browsing navigate to folder YourFolderNameMyFirstWindowsAzureApplicationMyFirstWindowsAzureApplicationbinReleaseapp.publish and choose files.





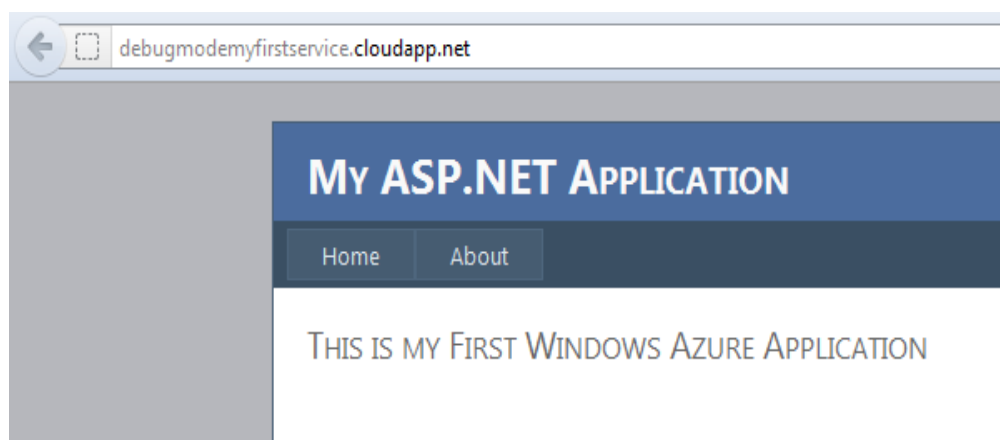
As of now for simplicity don't add any Certificate and click on Ok to create a hosted service with package of application created in last step. You will get a warning message. Click Yes on warning and proceed.



Now you need to wait for 5 to 10 minutes to get your application ready to use. Once service is ready you can see ready status for the Web Role.

Choose Columns		Filter hosted services	
Name	Type	Status	Environment
debugmode	Subscription	Active	
SecondApplication	Hosted Service	Created	
Certificates			
xyzDemo	Service Certificate	Created	
MyFirstService	Hosted Service	Created	
Certificates			
d1	Deployment	Ready	Production
WebRole1	Role	Ready	Production
WebRole1_IN_	Instance	Ready	Production
PDC08 CTP	Subscription	Disabled	

After stats are ready, you are successfully created and deployed first web application in Windows



=====

**Q.2) What is provisioning in cloud computing. How Virtual machine can be provision in Azure cloud.(6M)(S-17)**

**Ans:**

### **1. Provisioning VMM**

VMM 2012 R2 must be deployed in order to provision VM's

VMM requirements can be found at this link.

VMM step by step deployment guide can be found here.

### **2. Configure VMM with Hosts**

Configure Host Groups as per your resources and Add Hosts to the appropriate host groups. Information can be found here.

### **3. Configure VMM Networking**

Deploy Logical Networks and IP Pools / Network Sites, Deploy VLANs / NVGRE where appropriate and Deploy Virtual Networks. Information can be found at this link.

#### 4. Configure VMM Templates

Configure Hardware Profiles, Configure Guest OS Profile and Deploy VMM Templates.

#### 5. Configure SPF

Configure Service Account, Deploy SPF, Ensure SPF Account is a VMM Admin! And is a member of all the appropriate groups

---

### Q.3) Explain how window Azure maximize data availability and minimize security risks.(7M)

**Ans:**

Downtime is a fundamental metric for measuring productivity in a data warehouse, but this number does little to help you understand the basis of a system's availability. Focusing too much on the end-of-month number can perpetuate a bias toward a reactive view of availability. Root-cause analysis is important for preventing specific past problems from recurring, but it doesn't prevent new issues from causing future downtime.

Minimize risk, maximize availability

Potentially more dangerous is the false sense of security encouraged by historically high availability. Even perfect availability in the past provides no assurance that you are prepared to handle the risks that may lie just ahead or to keep pace with the changing needs of your system and users.

So how can you shift your perspective to a progressive view of providing for availability needs on a continual basis? The answer is availability management—a proactive approach to availability that applies risk management concepts to minimize the chance of downtime and prolonged outages. Teradata recommends four steps for successful availability management.

#### #1: Understand the risks

Effective availability management begins with understanding the nature of risk. "There are a variety of occurrences that negatively impact the site, system or data, which can reduce the availability experienced by end users. We refer to these as risk events," explains Kevin Lewis, director of Teradata Customer Services Offer Management.

The features of effective availability management:

- > Improves system productivity and quality of support
- > Encourages partnering to meet strategic and tactical availability needs
- > Recognizes all sources and impacts of availability risk
- > Applies a simple, holistic approach to risk mitigation
- > Facilitates communication between operations and management
- > Includes benchmarking using an objective, best-practice assessment

- > Establishes a clear improvement roadmap to meet evolving needs

The more vulnerable a system is to risk events, the greater the potential for extended outages or reduced availability and, consequently, lost business productivity.

Data warehousing risk events can range from the barely detectable to the inconvenient to the catastrophic. Risk events can be sorted into three familiar categories of downtime based on their type of impact:

Planned downtime is a scheduled system outage, usually during low-usage or non-critical periods (e.g., upgrades/updates, planned maintenance, testing).

Unplanned downtime is an unanticipated loss of system, data or application access (e.g., utility outages, human error, planned downtime overruns).

Degraded downtime is "low quality" availability in which the system is available, but performance is slow and inefficient (e.g., poor workload management, capacity exhaustion).

Although unplanned downtime is usually the most painful, companies have a growing need to reduce degraded and planned downtime as well. Given the variety of risk causes and impacts, follow the next step to reduce your system's vulnerability to risk events.

## **#2: Assess and strategize**

Although the occurrences of risk events to the Teradata system are often uncontrollable, applying a good availability management framework mitigates their impact. To meet strategic and tactical availability objectives, Teradata advocates a holistic system of seven attributes to address all areas that affect system availability. These availability management attributes are the tangible real-world IT assets, tools, people and processes that can be budgeted, assigned, administered and supervised to support system availability. They are:

**Environment.** The equipment layout and physical conditions within the data center that houses the infrastructure, including temperature, airflow, power quality and data center cleanliness

**Infrastructure.** The IT assets, the network architecture and configuration connecting them, and their compatibility with one another. These assets include the production system; dual systems; backup, archive and restore (BAR) hardware and software; test and development systems; and disaster recovery systems

**Technology.** The design of each system, including hardware and software versions, enabled utilities and tools, and remote connectivity Support level. Maintenance coverage hours, response times, proactive processes, support tools employed and the accompanying availability reports Operations. Operational procedures and support personnel used in the daily administration of the system and database Data protection. Processes and product features that minimize or eliminate data loss, corruption and theft; this includes system security, fallback, hot standby nodes, hot standby disks and large cliques Recoverability. Strategies and processes to regularly back up and archive data and to restore data and functionality in case of data loss or disaster

As evident in this list of attributes, supporting availability goes beyond maintenance service level agreements and downtime reporting. These attributes incorporate multiple technologies, service providers, support functions and management areas. This span necessitates an active partnership between Teradata and the customer to ensure all areas are adequately addressed. In addition to being

comprehensive, these attributes provide the benefit of a common language for communicating, identifying and addressing availability management needs.

Answer the sample best-practice questions for each attribute. A "no" response to any yes/no question represents an availability management gap. Other questions will help you assess your system's overall availability management.

Dan Odette, Teradata Availability Center of Expertise leader, explains: "Discussing these attributes with customers makes it easier for them to understand their system availability gaps and plan an improvement roadmap. This approach helps customers who are unfamiliar with the technical details of the Teradata system or IT support best practices such as the Information Technology Infrastructure Library [ITIL]."

### **#3: Weigh the odds**

To reduce the risk of downtime and/or prolonged outages, your availability management capabilities must be sufficient to meet your usage needs. (See figure 1, left.)

According to Chris Bowman, Teradata Technical Solutions architect, "Teradata encourages customers to obtain a more holistic view of their system availability and take appropriate action based on benchmarking across all of the attributes." In order to help customers accomplish this, Teradata offers an Availability Assessment service. "We apply Teradata technological and ITIL service management best practices to examine the people, processes, tools and architectural solutions across the seven attributes to identify system availability risks," Bowman says.

**Collect.** Data is collected across all attributes, including environmental measurements, current hardware/software configurations, historic incident data and best-practice conformity by all personnel that support and administer the Teradata system. This includes customer management and staff, Teradata support services, and possibly other external service providers. Much of this data can be collected remotely by Teradata, though an assigned liaison within the customer organization is requested to facilitate access to the system and coordinate any personnel interviews.

**Analyze.** Data is consolidated and analyzed by an availability management expert who has a strong understanding of the technical details within each attribute and their collective impact on availability. During this stage, the goal is to uncover gaps that may not be apparent because of a lack of best-practice knowledge or organizational "silos." Silos are characterized by a lack of cross-functional coordination due to separate decision-making hierarchies or competing organizational objectives.

**Recommend.** The key deliverable of an assessment is a clear list of practical recommendations for availability management improvements. To have the maximum positive impact, recommendations must include:

An unbiased, expert perspective of the customer's specific availability management situation  
Mitigation suggestions to prevent the recurrence of historical outages  
Quantified benchmarking across all attributes to pinpoint the areas of greatest vulnerability to risk events  
Corrective actions provided for every best-practice shortfall  
Operations-level improvement actions with technical details to facilitate tactical implementation  
Management-level guidance in the form of a less technical, executive scorecard to facilitate decision making and budget prioritization  
Teradata collects data

across all attributes and analyzes the current effectiveness of your availability management. The result is quantified benchmarking and actionable recommendations.

#### **#4: Plan the next move**

The recommendations from the assessment provide the basis for an availability management improvement roadmap "Cross-functional participation by both operations and management levels is crucial for maximizing the knowledge transfer of the assessment findings and ensuring follow-through," Odette says. Typically, not all of the recommendations can be implemented at once because of resource and budget constraints, so it's common to take a phased approach. Priorities are based on the assessment benchmarks, the customer's business objectives, the planned evolution for use of the Teradata system and cost-to-benefit considerations. Many improvements can be effectively cost-free to implement but still have a big impact. For example, adjusting equipment layout can improve airflow, which in turn can reduce heat-related equipment failures. Or, having the system/database administrators leverage the full capabilities of tools already built into Teradata can prevent or reduce outages. Lewis adds, "More significant improvements such as a disaster recovery capability or dual active systems may require greater investment and effort, but incremental steps can be planned and enacted over time to ensure availability management keeps pace with the customer's evolving needs." An effective availability management strategy requires a partnership between you, as the customer, and Teradata. Together, we can apply a comprehensive framework of best practices to proactively manage risk and provide for your ongoing availability needs.

---

#### **Q.4) Give the steps to create virtual machine. (7M) (W-17)**

Azure storage is one of the cloud computing PAAS (Platform as a service) service provided by the Microsoft Azure team. The storage option is one of the best computing services provided by Azure as it supports both legacy application development using Azure SQL and modern application development using Azure No-SQL table storage. Storage in Azure can be broadly classified into two categories based on the type of data that we are going to save.

1. Relational data Storage

2. Non-Relational data storage

#### **Relational Data Storage:**

Relational data can be saved in the cloud using Azure SQL storage.

#### **Azure SQL Storage:**

This kind of storage option is used when we want to store relational data in the cloud. This is one of the PAAS offerings from Azure built based on the SQL server relational database technology. Quick scalability and Pay as you Use options of SQL Azure encourages an organization to store their relational data into the cloud. This type of storage option

enables the developers/organizations to migrate on-premise SQL data to Azure SQL and vice versa for greater availability, reliability, durability, NoSqlscalability and data protection.

### **Non-Relational Data Storage:**

This kind of cloud storage option enables the users to store their documents, media filesNoSQLdata over the cloud that can be accessed using REST APIs. In order to work with this kind of data, we should have Storage account in the azure. Storage account structure can be shown below. Storage account wraps all the storage options provided by the azure like Blob storage, Queue storage, file storage, NoSQL storage. Access keys are used to authenticate storage account.

Azure provides four types of storage options based on the data type.

- 1.Blob storage
2. Queue storage
3. Table Storage
4. File storage

Blob Storage is used to store unstructured data such as text or binary data that can be accessed using HTTP or HTTPS from anywhere in the world.

Common usage of blob storage are as follows:

- For streaming audio and video
- For serving documents and images directly to the browser
- For storing data for big data analysis
- For storing data for backup, restore and disaster recovery.

Queue storage is used to transfer a large amount of data between cloud apps asynchronously. This kind of storage is mainly used to transfer the data between apps for asynchronous communication between cloud components. File storage is used when we want to store and share files using smb protocol. With Azure File storage, applications running in Azure virtual machines or cloud services can mount a file share in the cloud, just as a desktop application mounts a typical SMB share. Azure Table storage is not Azure SQL relational data storage, Table storage is the Microsoft's No-SQL database which stores data in a key-value pair. This kind of storage is used to store a large amount of data for future analysis using Hadoop

support. Following advantages makes azure storage popular in the market.

### **Scalability:**

We can start with small size blob and we can increase the size as per the demand to an infinite number without affecting production environment.

### **Secure and Reliable:**

Security for azure storage data can be provided in two ways, By using Storage Account access keys and by server level and client level encryption.

### **Durability & High availability:**

Replication concept has been used for azure storage in order to give high availability(99.99 % uptime) and durability. This replication concept maintains different copies of your data to different location or region based on the replication option [Locally redundant storage, Zone-redundant storage, Geo-redundant storage, Read-access geo-redundant storage] at the time of creating a storage account.

### **How is cloud storage different from on-premise data center?**

Simplicity, Scalability, Maintenance and Accessibility of data are the features which we expect from any public cloud storage and these are main assets of azure cloud storage and which is very difficult to get in on-premise datacenters. Simplicity: We can easily create and set up storage objects in azure. Scalability: Storage capacity is highly scalable and elastic.

Accessibility: Data in azure storage is easily searchable and accessible through the latest web technologies like HTTP and REST APIs. Multiprotocol (HTTP, TCP, etc) data access for modern applications makes azure to stand in the crowd. Maintenance and Backup of data: Not required to bother about maintenance of datacenter and backup of data everything will be taken care of the azure team. Azure's replication concept is used to maintain the different copies of data at different geo location. Using this we can protect our data even if a natural disaster occurs. High availability and disaster recovery are one of the good feature provided by azure storage which we cannot see in on-premise datacenters.

---

**Q.5) Explain how to deploy application using windows Azure subscription.(9M)(W-17)(w-16)**



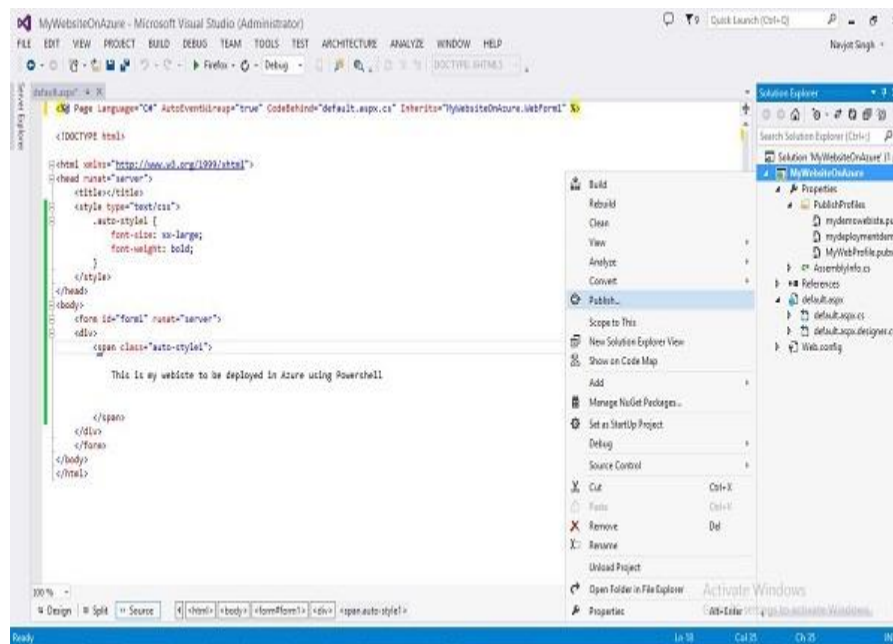
## Deploying a Web App from PowerShell

To get started with the PowerShell, refer to 'PowerShell' chapter in the tutorial. In order to deploy a website from PowerShell you will need the deployment package. You can get this from your website developers or you if you are into web deployment you would know about creating a deployment package. In the following sections, first you will learn how to create a deployment package in Visual Studio and then using PowerShell cmdlets, you will deploy the package on Azure.

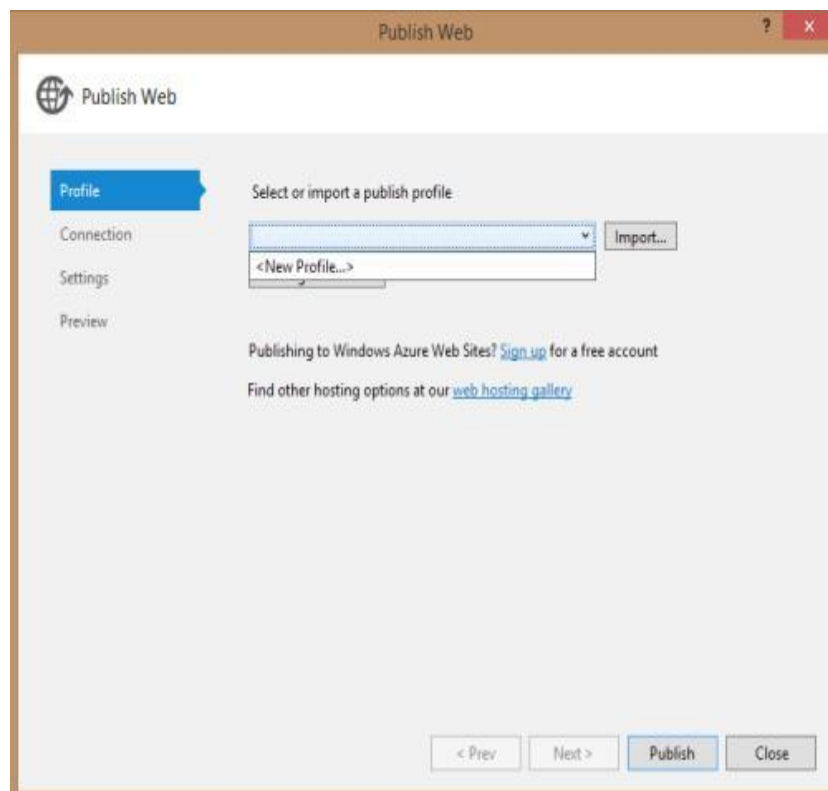
### Create a Deployment Package

**Step 1** – Go to your website in Visual Studio.

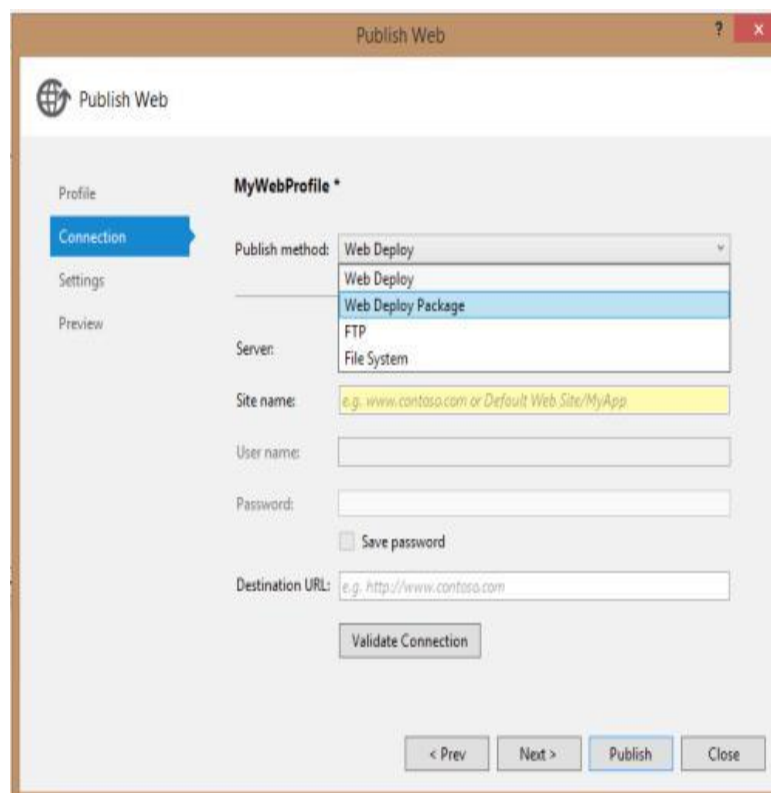
**Step 2** – Right-click on the name of the application in the solution explorer. Select 'Publish'.



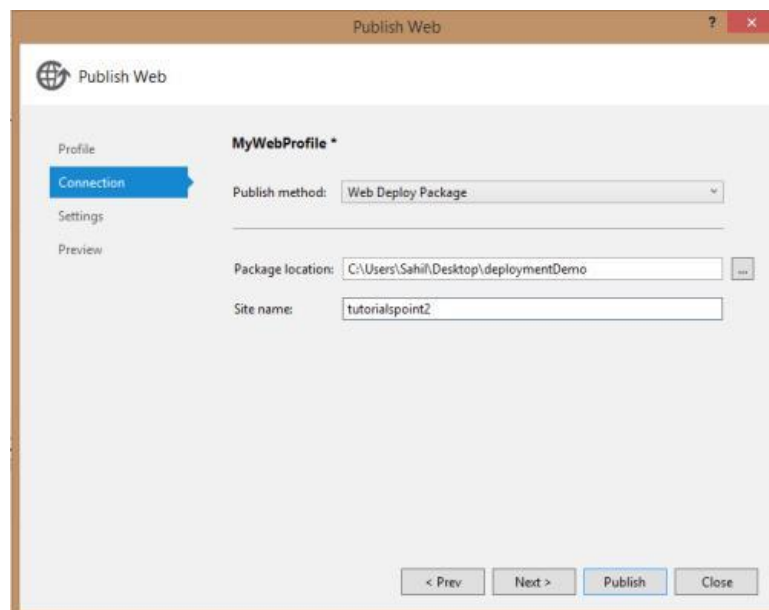
**Step 3** – Create a new profile by selecting 'New Profile' from the dropdown. Enter the name of the profile. There might be different options in dropdown depending on if the websites are published before from the same computer.



**Step 4** – On the next screen, choose ‘Web Deploy Package’ in Publish Method.



**Step 5** – Choose a path to store the deployment package. Enter the name of site and click Next.



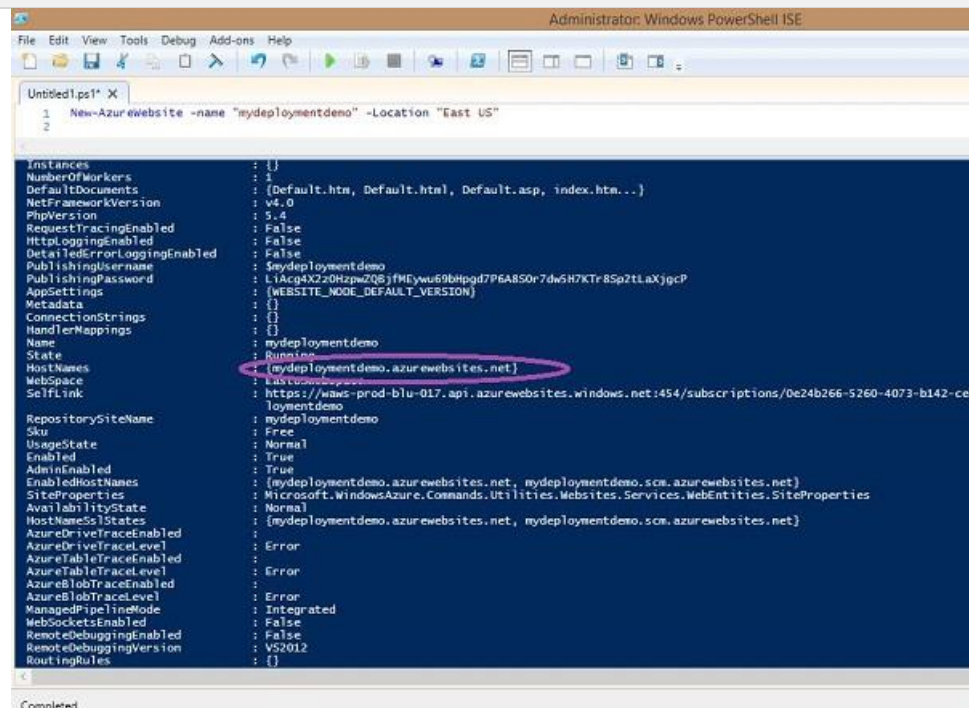
**Step 6** – On the next screen, leave the defaults on and select ‘publish’.

After it's done, inside the folder in your chosen location, you will find a zip file which is what you need during deployment.

Create a Website in Azure using PowerShell

**Step 1** – Enter the following cmdlets to create a website. Replace the highlighted part. This command is going to create a website in free subscription. You can change the subscription after the website is created.

New-AzureWebsite -name "mydeploymentdemo" -location "East US"



If cmdlet is successful, you will see all the information as shown in the above image. You can see the URL of your website as in this example it is mydeploymentdemo.azurewebsites.net.

**Step 2** – You can visit the URL to make sure everything has gone right.

#### Deploy Website using Deployment Package

Once the website is created in Azure, you just need to copy your website's code. Create the zip folder (deployment package) in your local computer.

**Step 1** – Enter the following cmdlets to deploy your website.

Here in above commandlet, the name of the website just created is given and the path of the zip file on the computer.

**Step 2** – Go to your website's URL. You can see the website as shown in the following image.




---

#### **Q.6) Write about worker role & web role while configuring an application in windows Azure.(5M)(W-17) (W-16)**

In Azure, a Cloud Service Role is a collection of managed, load-balanced, Platform-as-a-Service virtual machines that work together to perform common tasks. Cloud Service Roles are managed by Azure fabric controller and provide the ultimate combination of scalability, control, and customization

#### **What is a Web Role?**

Web Role is a Cloud Service role in Azure that is configured and customized to run web applications developed on programming languages/technologies that are supported by Internet Information Services (IIS), such as ASP.NET, PHP, Windows Communication Foundation and Fast CGI.

What is a Worker Role?

Worker Role is any role in Azure that runs applications and services level tasks, which generally do not require IIS. In Worker Roles, IIS is not installed by default. They are mainly used to perform supporting background processes along with Web Roles and do tasks such as automatically compressing uploaded images, run scripts when something changes in the database, get new messages from queue and process and more.

Differences between the Web and Worker Roles

The main difference between the two is that:

- a Web Role automatically deploys and hosts your app through IIS
- a Worker Role does not use IIS and runs your app standalone

Being deployed and delivered through the Azure Service Platform, both can be managed in the same way and can be deployed on a same Azure Instance.

In most scenarios, Web Role and Worker Role instances work together and are often used by an application simultaneously. For example, a web role instance might accept requests from users, then pass them to a worker role instance for processing.

### Worker Roles

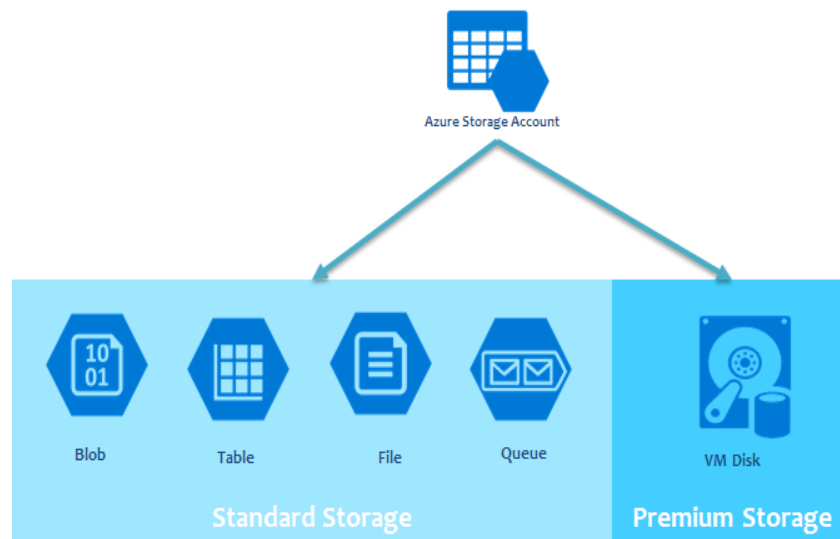
Azure Portal provides basic monitoring for Azure Web and Worker Roles. Users that require advanced monitoring, auto-scaling or self-healing features for their cloud role instances, should learn more about [CloudMonix](#). Along with advanced features designed to keep Cloud Services stable, CloudMonix also provides powerful dashboards, historical reporting, various integrations to popular ITSM and other IT tools and much more. Check out this [table](#) for a detailed comparison of CloudMonix vs native Azure monitoring features.

---

### Q.7) Explain types of storage in windows azure. (7M)(w-16)(W-18)

Azure Storage Types

With an Azure Storage account, you can choose from two kinds of storage services: **Standard Storage** which includes Blob, Table, Queue, and File storage types, and **Premium Storage** – Azure VM disks.



### *Standard Storage account*

With a Standard Storage Account, a user gets access to Blob Storage, Table Storage, File Storage, and Queue storage. Let's explain those just a bit better.

### **Azure Blob Storage**

Blob Storage is basically storage for unstructured data that can include pictures, videos, music files, documents, raw data, and log data...along with their meta-data. Blobs are stored in a directory-like structure called a “**container**”. If you are familiar with AWS S3, containers work much the same way as S3 buckets. You can store any number of blob files up to a total size of 500 TB and, like S3, you can also apply security policies. Blob storage can also be used for data or device backup.

Blob Storage service comes with three types of blobs: **block blobs**, **append blobs** and **page blobs**. You can use block blobs for documents, image files, and video file storage. Append blobs are similar to block blobs, but are more often used for append operations like logging. Page blobs are used for objects meant for frequent read-write operations. Page blobs are therefore used in Azure VMs to store OS and data disks.

To access a blob from storage, the URI should be:

`http://<storage-account-name>.blob.core.windows.net/<container-name>/<blob-name>`

For example, to access a movie called RIO from the bluesky container of an account called carlos, request:

`http://carlos.blob.core.windows.net/ bluesky/RIO.avi`

Note that container names are always in lower case.

### **Azure Table Storage**

Table storage, as the name indicates, is preferred for tabular data, which is ideal for key-value NoSQL data storage. Table Storage is massively scalable and extremely easy to use. Like other NoSQL data stores, it is schema-less and accessed via a REST API. A query to table storage might look like this:

`http://<storage account>.table.core.windows.net/<table>`

### **Azure File Storage**

Azure File Storage is meant for legacy applications. Azure VMs and services share their data via mounted file shares, while on-premise applications access the files using the File Service REST API. Azure File Storage offers file shares in the cloud using the standard SMB protocol and supports both SMB 3.0 and SMB 2.1.

### **Azure Queue Storage**

The Queue Storage service is used to exchange messages between components either in the cloud or on-premise (compare to Amazon's SQS). You can store large numbers of messages to be shared between independent components of applications and communicated asynchronously via HTTP or HTTPS. Typical use cases of Queue Storage include processing backlog messages or exchanging messages between Azure Web roles and Worker roles.

A query to Queue Storage might look like this:

`http://<account>.queue.core.windows.net/<file_to_download>`

### ***Premium Storage account:***

The Azure Premium Storage service is the most recent storage offering from Microsoft, in which data are stored in Solid State Drives (SSDs) for better IO and throughput. Premium storage only supports Page Blobs.

=====

### **Q.8) Explain the complete Azure life cycle in detail.(6M)(W-18)**

The development lifecycle of software that uses the Azure platform mainly follows two processes:

#### **Application Development**

During the application development stage the code for Azure applications is most commonly built locally on a developer's machine. Microsoft has recently added additional services to Azure Apps named Azure Functions. They are a representation of 'serverless' computing and allow developers to build application code directly through the Azure portal using references to a number of different Azure services.

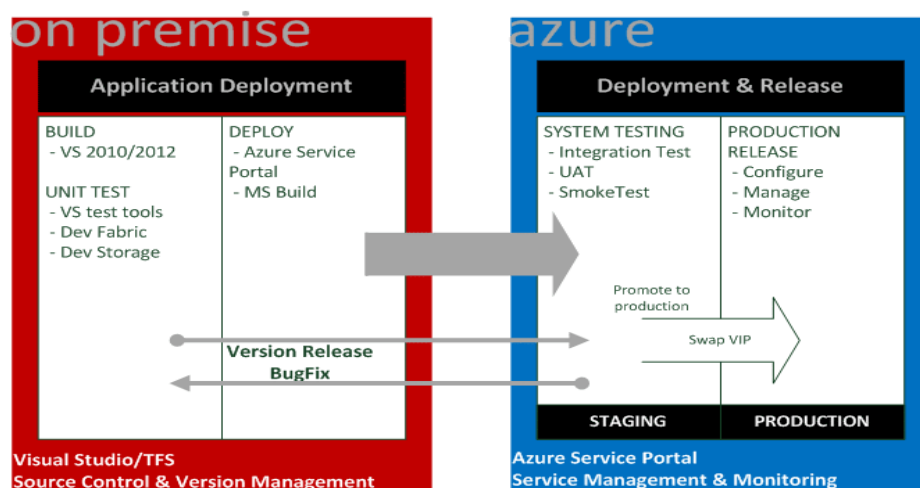
The application development process includes two phases: 1) Construct + Test and 2) Deploy + Monitor.

### Construct & Test

In the development and testing phase, a Windows Azure application is built in the Visual Studio IDE (2010 or above). Developers working on non-Microsoft applications who want to start using Azure services can certainly do so by using their existing development platform. Community-built libraries such as Eclipse Plugins, SDKs for Java, PHP or Ruby are available and make this possible.

Visual Studio Code is a tool that was created as a part of Microsoft efforts to better serve developers and recognize their needs for lighter and yet powerful/highly-configurable tools. This source code editor is available for Windows, Mac and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js. It also has a rich ecosystem of extensions and runtimes for other languages such as C++, C#, Python, PHP and Go.

That said, Visual Studio provides developers with the best development platform to build Windows Azure applications or consume Azure services.



Visual Studio and the Azure SDK provide the ability to create and deploy project infrastructure and code to Azure directly from the IDE. A developer can define the web host, website and database for an app and deploy them along with the code without ever leaving Visual Studio.

Microsoft also proposed a specialized Azure Resource Group deployment project template in Visual Studio that provides all the needed resources to make a deployment in a single, repeatable operation. Azure Resource Group projects work with preconfigured and customized JSON templates, which contain all the information needed for the resources to be deployed on Azure. In most scenarios, where multiple developers or development teams work simultaneously on the same Azure solution, configuration management is an essential part of the development lifecycle.



**Q.9) Write down the steps involved in deployment of an application to windows Azure cloud.(7M)(W-18)**

Downtime is a fundamental metric for measuring productivity in a data warehouse, but this number does little to help you understand the basis of a system's availability. Focusing too much on the end-of-month number can perpetuate a bias toward a reactive view of availability. Root-cause analysis is important for preventing specific past problems from recurring, but it doesn't prevent new issues from causing future downtime.

Minimize risk, maximize availability

Potentially more dangerous is the false sense of security encouraged by historically high availability. Even perfect availability in the past provides no assurance that you are prepared to handle the risks that may lie just ahead or to keep pace with the changing needs of your system and users.

So how can you shift your perspective to a progressive view of providing for availability needs on a continual basis? The answer is availability management—a proactive approach to availability that applies risk management concepts to minimize the chance of downtime and prolonged outages. Teradata recommends four steps for successful availability management.

**#1: Understand the risks**

Effective availability management begins with understanding the nature of risk. "There are a variety of occurrences that negatively impact the site, system or data, which can reduce the availability experienced by end users. We refer to these as risk events," explains Kevin Lewis, director of Teradata Customer Services Offer Management.

The features of effective availability management:

- > Improves system productivity and quality of support
- > Encourages partnering to meet strategic and tactical availability needs
- > Recognizes all sources and impacts of availability risk
- > Applies a simple, holistic approach to risk mitigation
- > Facilitates communication between operations and management
- > Includes benchmarking using an objective, best-practice assessment
- > Establishes a clear improvement roadmap to meet evolving needs

The more vulnerable a system is to risk events, the greater the potential for extended outages or reduced availability and, consequently, lost business productivity.

Data warehousing risk events can range from the barely detectable to the inconvenient to the catastrophic. Risk events can be sorted into three familiar categories of downtime based on their type of impact:

Planned downtime is a scheduled system outage, usually during low-usage or non-critical periods (e.g., upgrades/updates, planned maintenance, testing).

Unplanned downtime is an unanticipated loss of system, data or application access (e.g.,

utility outages, human error, planned downtime overruns).

Degraded downtime is "low quality" availability in which the system is available, but performance is slow and inefficient (e.g., poor workload management, capacity exhaustion).

Although unplanned downtime is usually the most painful, companies have a growing need to reduce degraded and planned downtime as well. Given the variety of risk causes and impacts, follow the next step to reduce your system's vulnerability to risk events.

## **#2: Assess and strategize**

Although the occurrences of risk events to the Teradata system are often uncontrollable, applying a good availability management framework mitigates their impact. To meet strategic and tactical availability objectives, Teradata advocates a holistic system of seven attributes to address all areas that affect system availability. These availability management attributes are the tangible real-world IT assets, tools, people and processes that can be budgeted, assigned, administered and supervised to support system availability. They are:

**Environment.** The equipment layout and physical conditions within the data center that houses the infrastructure, including temperature, airflow, power quality and data center cleanliness

**Infrastructure.** The IT assets, the network architecture and configuration connecting them, and their compatibility with one another. These assets include the production system; dual systems; backup, archive and restore (BAR) hardware and software; test and development systems; and disaster recovery systems

**Technology.** The design of each system, including hardware and software versions, enabled utilities and tools, and remote connectivity

**Support level.** Maintenance coverage hours, response times, proactive processes, support tools employed and the accompanying availability reports

**Operations.** Operational procedures and support personnel used in the daily administration of the system and database

**Data protection.** Processes and product features that minimize or eliminate data loss, corruption and theft; this includes system security, fallback, hot standby nodes, hot standby disks and large cliques

**Recoverability.** Strategies and processes to regularly back up and archive data and to restore data and functionality in case of data loss or disaster

As evident in this list of attributes, supporting availability goes beyond maintenance service level agreements and downtime reporting. These attributes incorporate multiple technologies, service providers, support functions and management areas. This span necessitates an active partnership between Teradata and the customer to ensure all areas are adequately addressed. In addition to being comprehensive, these attributes provide the benefit of a common language for communicating, identifying and addressing availability management needs.

Answer the sample best-practice questions for each attribute. A "no" response to any yes/no question represents an availability management gap. Other questions will help you assess your system's overall availability management.

Dan Odette, Teradata Availability Center of Expertise leader, explains: "Discussing these attributes with customers makes it easier for them to understand their system availability gaps and plan an improvement roadmap. This approach helps customers who are unfamiliar with the technical details of the Teradata system or IT support best practices such as the Information Technology Infrastructure Library [ITIL]."

### **#3: Weigh the odds**

To reduce the risk of downtime and/or prolonged outages, your availability management capabilities must be sufficient to meet your usage needs. (See figure 1, left.)

According to Chris Bowman, Teradata Technical Solutions architect, "Teradata encourages customers to obtain a more holistic view of their system availability and take appropriate action based on benchmarking across all of the attributes." In order to help customers accomplish this, Teradata offers an Availability Assessment service. "We apply Teradata technological and ITIL service management best practices to examine the people, processes, tools and architectural solutions across the seven attributes to identify system availability risks," Bowman says.

**Collect.** Data is collected across all attributes, including environmental measurements, current hardware/software configurations, historic incident data and best-practice conformity by all personnel that support and administer the Teradata system. This includes customer management and staff, Teradata support services, and possibly other external service providers. Much of this data can be collected remotely by Teradata, though an assigned liaison within the customer organization is requested to facilitate access to the system and coordinate any personnel interviews.

**Analyze.** Data is consolidated and analyzed by an availability management expert who has a strong understanding of the technical details within each attribute and their collective impact on availability. During this stage, the goal is to uncover gaps that may not be apparent because of a lack of best-practice knowledge or organizational "silos." Silos are characterized by a lack of cross-functional coordination due to separate decision-making hierarchies or competing organizational objectives.

**Recommend.** The key deliverable of an assessment is a clear list of practical recommendations for availability management improvements. To have the maximum positive impact, recommendations must include:

An unbiased, expert perspective of the customer's specific availability management situation

Mitigation suggestions to prevent the recurrence of historical outages

Quantified benchmarking across all attributes to pinpoint the areas of greatest vulnerability to risk events

Corrective actions provided for every best-practice shortfall

Operations-level improvement actions with technical details to facilitate tactical implementation

Management-level guidance in the form of a less technical, executive scorecard to facilitate decision making and budget prioritization

Teradata collects data across all attributes and analyzes the current effectiveness of your availability management. The result is quantified benchmarking and actionable recommendations.

#### #4: Plan the next move

The recommendations from the assessment provide the basis for an availability management improvement roadmap.

"Cross-functional participation by both operations and management levels is crucial for maximizing the knowledge transfer of the assessment findings and ensuring follow-through," Odette says.

Typically, not all of the recommendations can be implemented at once because of resource and budget constraints, so it's common to take a phased approach. Priorities are based on the assessment benchmarks, the customer's business objectives, the planned evolution for use of the Teradata system and cost-to-benefit considerations.

Many improvements can be effectively cost-free to implement but still have a big impact. For example, adjusting equipment layout can improve airflow, which in turn can reduce heat-related equipment failures. Or, having the system/database administrators leverage the full capabilities of tools already built into Teradata can prevent or reduce outages. Lewis adds, "More significant improvements such as a disaster recovery capability or dual active systems may require greater investment and effort, but incremental steps can be planned and enacted over time to ensure availability management keeps pace with the customer's evolving needs."

An effective availability management strategy requires a partnership between you, as the customer, and Teradata. Together, we can apply a comprehensive framework of best practices to proactively manage risk and provide for your ongoing availability needs.

---

#### **Q.10) What are the steps for creating a simple cloud application using Azure. Explain with the help of an example.(6M)(W-18)**

1. Installation of Windows Azure SDK
2. Developing First Windows Azure Web Application
3. Deploying application locally in Development Storage Fabric
4. Registration for free Windows Azure Trial
5. Deployment of the Application in Microsoft Data Center

I will start fresh with installation of Windows Azure SDK and I will conclude this post with deployment of simple application in Windows Azure Hosted Service. I am not going to create a complex application since purpose of this post is to walkthrough with all the steps from installation, development, debugging to deployment. In further post we will get into more complex applications. Proceed through rest of the post to create your first application for Windows Azure.

---