Lecture 1: Introduction to Cloud Computing

- **Evolution of IT Infrastructure**:
  - From Mainframe/Mini Computers to Cloud/Mobile Computing.
  - The evolution paralleled with the history of electricity provision – from private generation to public supply.
- **Traditional IT vs. Cloud**:
  - In traditional IT, a long process of decision, procurement, implementation, and going live can take 6-12 months, often leading to under-supply or waste of capacities.
  - Cloud offers a better alternative with quick provision of environments, PAYG pricing, and easy capacity adjustments, reducing initial investments and waste.
- **Characteristics of Cloud Computing**:
  - On-demand self-service, resource pooling, rapid elasticity, broad network access, and measured service.
- **Cloud Service Models**:
  - **IaaS (Infrastructure as a Service)**: Consumers provision servers, databases, and networks; manage virtualization layers.
  - **PaaS (Platform as a Service)**: Consumers deploy applications on cloud platforms managed by providers.

    - **SaaS (Software as a Service)**: Consumers use provider-managed applications with no need for infrastructure management.
- **Cloud Deployment Models**:
    - **Public Cloud**: Open for the general public, cost-effective, less secure.
    - **Private Cloud**: Exclusive for one organization, more secure, more expensive.
    - **Hybrid Cloud**: Mix of private and public, balance of security and cost.
    - **Community Cloud**: For specific communities with shared concerns.
- **Cloud Providers and Market Share**:
    - Dominance of Amazon AWS, Microsoft Azure, and Google Cloud Platform in the market.
    - Importance of the cloud in the IT industry, similar to the utility of electricity.
- **NIST Definition of Cloud**: On-demand network access to shared resources, easy provisioning and release with minimal management.
- **Actors in Cloud Computing**:
    - Cloud Consumer, Provider, Auditor, Broker, and Carrier, each with specific roles in service provision, usage, and management.
- **References**:

    - National Institute of Standards and Technology for Cloud definitions and architecture.
    - Various online resources for further information on IT infrastructure evolution.

For a comprehensive understanding, review the comparisons between traditional IT processes and cloud computing's efficiencies, the essential characteristics of cloud, and the roles of different cloud actors as outlined in the NIST framework.

Lecture 2: Core Cloud Services and Global Architecture

- **History of Cloud Computing**:
    - Amazon Web Services (AWS) launched in July 2002.
    - AWS released S3 and EC2 in 2006, RDS and VPC services in 2009.
    - Microsoft announced Azure in 2008, with general availability (GA) of Azure Web Sites VMs for Linux/Windows in 2012.
    - AWS released their first service, SQS, in 2004, and GA of SQL Azure and Windows Azure happened in 2010.
- **Core Cloud Services**:
    - Cloud services are broadly categorized into Compute, Storage, and Network.
    - AWS and Azure offer a variety of services in these categories.
- **Global Architecture**:

    - Datacenter, Availability Zone, Region, and Geography are key structural elements of cloud services.
    - The global network is integral for providing high availability and reliability.
    - AWS and Azure both have their own Shared Responsibility Models outlining the security responsibilities of the cloud provider and the customer.
- **Accessing Resources**:
    - Resources can be accessed through Web, Console, CLI, SDK, and REST API.
- **Selecting a Region**:
    - Considerations include data governance, legal requirements, proximity to users, service availability, and cost variations by region.
- **Account Management**:
    - AWS uses Accounts and Management Accounts to organize resources.
    - Azure uses Subscriptions and Accounts for the same purpose.
- **Pricing**:
    - Cloud services follow a pay-as-you-go (PAYG) model with possible discounts on reservations and volume-based discounts.
    - Providers offer per second/minute billing and provide pricing calculators.

- **Service Limits/Quotas**:
    - There are default quotas for services which are region-specific and can include soft limits (increasable upon request) and hard limits (not increasable).
    - Service limits are to prevent excessive charges and overuse of resources by a single entity.
- **Service and Resource Scopes**:
    - Scopes can be global, regional, or zonal.

Understanding the chronological development of cloud services, the framework of global architecture, and the financial aspects of using cloud services is essential. Remember to pay close attention to the Shared Responsibility Model and service limits as they are critical for security and cost management.

Lecture 3 Part 1: Cloud Compute & Storage

- **Virtualization / Containerization**:
    - Comparing Virtual Machines with Containers.
    - VMs include an entire copy of an OS plus its overhead, while containers share the OS kernel.
- **Software Defined Compute**:
    - It virtualizes and abstracts compute functions from the hardware.
    - Workloads can be dispersed among any number of processing units.

    - Hardware tends to be generic and industry-standard to easily meet demand.
- **Cloud Compute Services**:
    - AWS EC2 and Azure VMs are leading services.
    - Both offer Infrastructure as a Service (IaaS).
    - AWS has per second billing; Azure has per minute billing.
- **Instance Types**:
    - Ranging from General Purpose, Compute Optimized, Memory Optimized, Storage Optimized to Accelerated Computing and Burstable Instances.
- **Tenancy & Purchasing Options**:
    - Default/Shared, Dedicated Instances, and Dedicated Hosts.
    - On-demand, Reserved, Spot Instances, and Savings Plans.
- **Instance Lifecycle**:
    - The process from launch to termination of a cloud compute instance.
- **Placement Groups**:
    - AWS: Cluster, Partition, and Spread Placement Groups.
    - Azure: Proximity Placement Groups.

Understanding the differences between virtualization and containerization, the types of compute instances, and their lifecycle can help optimize cloud compute resources. Additionally, the purchasing options and placement groups have implications for cost and performance.

Lecture 3 Part 2: Cloud Compute & Storage

- **Software Defined Storage (SDS)**:
  - SDS architecture abstracts storage resources from the underlying hardware.
  - Provides flexibility in managing and scaling storage resources.
- **CAP Theorem**:
  - It's impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees: Consistency, Availability, Partition Tolerance.
  - In the event of a network partition, one must choose between consistency and availability.
- **Types of Storage**:
  - **Block Storage**: Manages data as blocks within sectors and tracks, suitable for structured data like databases and logs.
    - Performance measured in IOPS.
    - AWS Elastic Block Storage, Azure Disks are examples.
  - **File Storage**: Manages data as a file hierarchy, commonly used over shared networks.
    - Supports protocols like NFS and SMB.
    - AWS Elastic File System, Azure Files are examples.
  - **Object Storage**: Manages data as objects, each consisting of data, metadata, and a globally unique identifier.

        - Accessible through HTTPS, APIs, or SDKs.
        - AWS S3, Azure Blobs are examples.
- **Object Storage Features**:
    - Versioning, encryption (client-side, server-side, with various key management options), web hosting, and immutability (time-based retention, legal hold).
- **Object Storage Tiers**:
    - Cloud Service Providers (CSPs) offer different access tiers for various use cases:
        - AWS: Standard, Intelligent-Tiering, Infrequent Access, Glacier.
        - Azure: Hot, Cool, Archive.
- **Other Important Aspects of Cloud Storage**:
    - Archiving and backup solutions.
    - Hybrid storage options.
    - Bulk data transfer methods.

Understanding the various types of storage and their specific features is crucial for designing and implementing efficient and secure cloud storage solutions. The CAP theorem is a fundamental principle in distributed systems that affects the design of cloud storage architectures.

Lecture 4: Cloud High Availability & Disaster Recovery

- **Downtime**: This refers to periods when a system is unavailable, which can be planned (scheduled maintenance) or unplanned (failures or breaches).
- **Availability**: Expressed as a percentage, availability reflects the operational uptime of a system. The higher the availability percentage (like "five nines" or 99.999%), the less downtime per year.
- **High Availability (HA)**:
    - HA aims to ensure an agreed level of operational performance, usually uptime, for a higher-than-normal period.
    - Key strategies for HA include the elimination of single points of failure, reliable crossover, redundancy, and failure detection.
- **Redundancy**:
    - Passive Redundancy: Including excess capacity to accommodate performance decline.
    - Active Redundancy: Using complex systems to detect failure and reconfigure automatically.
- **Failover and Failback**:
    - Failover is the process of switching to a standby instance upon system failure.
    - Failback is the restoration of a failed system to its original working state.

- **Replication**:
    - Ensures consistency between redundant resources to improve reliability, fault tolerance, or accessibility.
    - Active Replication processes the same request at every replica.
    - Passive Replication processes requests at one replica, then transfers the result to others.
- **High Availability Clusters**: Groups of computers that support applications to provide continued service during failures, with configurations like Active/Active or Active/Passive.
- **Load Balancing**: Distributes network traffic across a group of backend servers to enhance the efficiency and capacity of service handling.
- **Business Continuity (BC)**:
    - The capability of an organization to continue delivery of products or services at pre-defined acceptable levels following a disruptive incident.
    - Involves Business Continuity Planning (BCP) to create systems of prevention and recovery to deal with potential threats.
- **Disaster Recovery (DR)**:
    - Part of BCP involving policies, tools, and procedures to enable the recovery or continuation of vital technology infrastructure and systems following a disaster.
    - Elements of a DR plan include risk evaluation, critical asset identification, backup strategy, and testing/optimization.

- **Backup Strategies**:
    - Full, Incremental, Differential, Near Continuous Data Protection.
- **DR Tiers**: Range from no off-site data (Tier 0) to highly automated, business-integrated solutions (Tier 6).
- **Recovery Objectives**:
    - Recovery Time Objective (RTO): The targeted period for restoring business processes after a disaster.
    - Recovery Point Objective (RPO): The acceptable period for data loss due to a disaster.
    - Recovery Consistency Objective (RCO): The amount of data inconsistency tolerable in a recovered dataset.

Understanding high availability, disaster recovery, and business continuity are essential for ensuring that cloud services remain operational and resilient to failures or disruptions. Backup strategies and recovery objectives are central to developing effective disaster recovery plans.

Lecture 5: Cloud Networking

- **Networking Recap**:
    - Covers IP addresses, subnets/VLSM, basic routing, and default gateways.
- **Software Defined Networking (SDN)**:

    - A network management approach that enables dynamic, programmatically efficient network configuration.
    - It separates network control and forwarding functions, allowing the network control to be directly programmable.
- **Network Function Virtualization (NFV)**:
    - NFV decouples physical network functions from hardware and runs them as software in VMs or containers.
    - Includes functions like firewalls, traffic control, and virtual routing.
- **Virtual Networks and Subnets**:
    - Virtual networks are isolated logical networks with one or more IP ranges and subnets.
    - AWS VPC and Azure Virtual Networks are examples.
- **Access Controlling**:
    - Firewall functionalities are implemented to control inbound and outbound traffic through ACL rules.
- **Network Gateway Services**:
    - Network gateways like Internet Gateway, NAT Gateway, VPN Gateway, and Network Transit Gateway provide connectivity between cloud resources and outside networks.
- **Network Peering**:
    - Allows for the connection and communication between two virtual networks.

    - Peering is non-transitive, meaning if VNet A is peered with VNet B, and VNet B is peered with VNet C, A and C cannot communicate unless directly peered.
- **Hybrid Connectivity**:
    - Various methods such as direct internet connectivity, site-to-site VPN, and AWS Direct Connect or Azure Express Route are used to connect remote users and networks to the cloud.
- **Private Access to Cloud Services**:
    - Private links provide access to PaaS services from the virtual network without going through the public internet.
    - Azure Service Endpoints and AWS Private Link are examples.
- **Flow Logs**:
    - Captures information about IP traffic going to and from network interfaces.
    - Used for troubleshooting, intrusion detection, and compliance.

Understanding cloud networking fundamentals, along with the nuances of network virtualization, access control, and gateway services, is vital. Knowing how to connect and manage traffic between cloud services and on-premises networks is equally important for designing secure and efficient cloud architectures.

Lecture 6: Cloud Security & Monitoring

- **Software Defined Security (SDS)**:
    - SDS is a security model where information security is implemented, controlled, and managed by software.
    - It's policy-driven and automates security controls like intrusion detection, network segmentation, and access controls.
- **Principal of Least Privilege (PLOP)**:
    - In information security, PLOP dictates that a user or process should only be granted privileges essential for its intended function.
- **Identity and Access Management (IAM)**:
    - IAM is a framework of policies and technologies ensuring appropriate access to technology resources.
    - Components of IAM include logical organization, users, groups, policies, and roles.
    - Multi-Factor Authentication (MFA) is a key security feature within IAM frameworks.
- **Federated Identity & Single Sign-On (SSO)**:
    - Federated Identity allows for a linked electronic identity across multiple identity management systems.
    - SSO enables users to move between multiple systems without re-authentication.
- **Encryption Types**:

    - Symmetric Key, Asymmetric Key, and Hashing are the primary types of encryption.
    - Encryption stages include Data in Use, Data in Transit, and Data at Rest.
- **Cloud Security Risks & Threats**:
    - Challenges include poor access management, data breaches, leaks, misconfiguration, insecure APIs, account hijacking, lack of visibility, and DoS/DDoS attacks.
- **Monitoring in Cloud**:
    - Cloud monitoring is reviewing, observing, and managing the operational workflow in cloud-based IT infrastructure.
    - It can be manual or automated and involves the use of services like Datadog, AppDynamics, Azure Monitor, and Amazon CloudWatch.
- **Metrics, Events & Logs**:
    - Metrics: Raw data on resource usage or behavior.
    - Events: Generated by systems to capture what, where, and when something happens.
    - Logs: Provide detailed information on what systems have been doing and are crucial for troubleshooting.
- **Alerts**:
    - Alerts can be reactive elements of the monitoring system, triggering actions based on changes in metrics, events, or logs.

    - Types of alerts include threshold-based, anomaly detection-based, and heartbeat alerts.

It's important to understand how SDS, IAM, and encryption contribute to a robust security posture in the cloud. Additionally, effective monitoring and alerting are crucial for maintaining operational health and security compliance.

Lecture 7: Cloud Application Deployment & Management

- **Continuous Integration and Continuous Deployment (CI/CD)**:
    - CI/CD is a method to frequently deliver apps to customers by introducing automation into the stages of app development.
    - The main concepts attributed to CI/CD are continuous integration, continuous deployment, and continuous delivery.
- **Deployment Strategies**:
    - **Canary Releases**: Roll out the change to a small subset of users before rolling it out to the entire infrastructure.
    - **Blue-Green Deployment**: Reduce downtime and risk by running two identical production environments.
- **Infrastructure as Code (IaC)**:
    - IaC is the management of infrastructure (networks, virtual machines, load balancers, and connection topology) in a descriptive model, using the same versioning as DevOps team uses for source code.
- **Configuration Management vs. Provisioning**:

- Configuration management tools, like Ansible, Puppet, Chef, and SaltStack, are designed to install and manage software on existing servers.
- Provisioning tools, like Terraform and CloudFormation, are designed to provision the servers themselves.

Lecture 8: Cloud Governance & Compliance

- **Cloud Governance**:
  - Cloud governance is a set of rules and principles that guide IT decision-making regarding the purchase and operation of cloud services.
- **Compliance**:
  - Compliance in the cloud ensures that cloud services and deployments follow all relevant compliance requirements, which may be internal policies or external regulations and laws.
- **Cost Management**:
  - It involves understanding and controlling where the money is spent in the cloud, identifying wasted resources, and selecting the right size of resources.
- **Security & Identity**:
  - Cloud governance should ensure that security and identity policies are followed, access is controlled, and data is protected.
- **Resource Consistency**:

    - Ensure consistent setup and configuration of resources to meet organizational standards.
- **Tagging & Resource Organization**:
    - Using tags to organize resources can simplify management, cost tracking, and compliance.
    -

Lecture 9: Cloud Native Applications

- **Monolithic vs. Microservices Architecture**:
    - Monolithic architectures are traditional ways of building applications as a single unit, whereas microservices architecture breaks the application into smaller, independent units that work together.
- **12 Factor App Methodology**:
    - This is a methodology for building software-as-a-service apps that obey twelve key rules, such as keeping the development, staging, and production as similar as possible and using declarative formats for setup automation to minimize time and cost for new developers joining the project.
- **Cloud Native Computing Foundation (CNCF)**:
    - CNCF is focused on making cloud-native computing universal and sustainable. It champions the cloud-native approach and oversees a number of open-source projects that are foundational to this method of building and running applications.

- **Site Reliability Engineering (SRE)**:
    - Google's SRE is a discipline that incorporates aspects of software engineering and applies them to infrastructure and operations problems, with the goal of creating scalable and highly reliable software systems.
- **DevOps & NoOps**:
    - DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) to shorten the development life cycle. NoOps is the concept that an IT environment can be automated and abstracted from the underlying infrastructure to the extent that there is no need for a dedicated team to manage software in-house.
- **Application Modernization**:
    - This involves updating legacy software applications to scalable, cloud-native models to improve features and maintainability. The modernization journey includes rehosting, replatforming, refactoring, rearchitecting, and rebuilding applications.

Summary for All the Lectures for IT4090

1. **Introduction to Cloud Computing**: Covered the evolution of IT infrastructure, comparison between traditional IT and Cloud, Cloud characteristics, service models, and deployment models.

2. **Core Cloud Services and Global Architecture**: Discussed the development of cloud services, core service categories, global architecture, accessing resources, and considerations for selecting a region.
3. **Cloud Compute & Storage (Part 1)**: Focused on virtualization versus containerization, software-defined compute, cloud compute services, instance types, tenancy, purchasing options, and instance lifecycle.
4. **Cloud Compute & Storage (Part 2)**: Continued with software-defined storage, CAP theorem, types of storage, object storage features, storage tiers, and other aspects of cloud storage.
5. **Cloud High Availability & Disaster Recovery**: Explained downtime, availability, high availability strategies, redundancy, failover and failback, replication, HA clusters, load balancing, business continuity, disaster recovery, backup strategies, DR tiers, and recovery objectives.
6. **Cloud Networking**: Recapped basic networking concepts, software-defined networking, network function virtualization, virtual networks, access controlling, network gateway services, network peering, hybrid connectivity, private access to services, and flow logs.
7. **Cloud Security & Monitoring**: Covered software-defined security, the principle of least privilege, identity and access management, federated identity, single sign-on, encryption types, cloud security risks, monitoring, metrics, events, logs, and alerts.
8. **Cloud Application Deployment & Management**: Talked about continuous integration/deployment, deployment strategies,

infrastructure as code, configuration management versus provisioning.

9. **Cloud Governance & Compliance**: Addressed cloud governance, compliance, cost management, security & identity, resource consistency, tagging, and resource organization.
10. **Cloud Native Applications**: Discussed the difference between monolithic and microservices architecture, the 12-factor app methodology, Cloud Native Computing Foundation, Site Reliability Engineering, DevOps & NoOps, and application modernization.