Server Environments

1. **Development Environment**: The development environment is where developers write and test their code, and create new features for the application. This environment is typically hosted on local machines or a dedicated development server.

2. **Test Environment**: The test environment is used to validate the functionality and quality of the application before it is deployed to production. It mimics the production environment as closely as possible and can be hosted on dedicated servers or virtual machines.

3. **Staging Environment**: The staging environment is a replica of the production environment and is used to test the application in a production-like environment before it is released to production.

4. **Production Environment**: The production environment is where the application is deployed and made available to end-users. It is critical that the production environment is stable, secure, and highly available to ensure the application is always accessible.

RAGHU

Server Types / Categories

1. **Web Server**: A web server is used to host websites and web applications. It receives and processes HTTP requests from clients and sends back the corresponding HTML pages or other web resources.

2. **Application Server**: An application server is used to host and run business applications, such as Enterprise Resource Planning (ERP) software, Customer Relationship Management (CRM) software, and other applications.

3. **Database Server**: A database server is used to host databases and store data. It provides access to the stored data to client applications.

4. **Mail Server**: A mail server is used to host and manage email accounts for an organization. It receives and sends email messages to and from clients and other mail servers.

5. **File Server**: A file server is used to host and manage files and folders that are shared within an organization. It enables users to store, access, and share files and documents.

6. **Print Server**: A print server is used to manage and control printing tasks within an organization. It enables users to print documents and manage print jobs from a central location.

7. **DNS Server**: A DNS server is used to translate domain names into IP addresses. It provides a way for clients to locate and connect to servers and other network devices using friendly domain names.

8. **Proxy Server**: A proxy server is used to provide security, privacy, and performance improvements for client requests. It acts as an intermediary between clients and servers, intercepting and processing requests on behalf of the clients.

9. **Backup Server**: A backup server is used to store and manage backups of important data and applications. It provides a way to recover data in case of data loss or system failure.

Testing Types

1. **Unit Testing**: It is the process of testing individual units or components of the software to ensure that they work as expected. It is usually done by developers during the development phase.

2. **Integration Testing**: It is the process of testing the integration of different components of the software to ensure that they work together as expected. It is usually done after unit testing.

3. **System Testing**: It is the process of testing the entire software system as a whole to ensure that it meets the specified requirements. It is usually done after integration testing. System testing can involve various types of tests such as functional testing, performance testing, security testing, usability testing, and compatibility testing.

4. **Acceptance Testing**: It is the process of testing the software by end-users or customers to ensure that it meets their requirements and is fit for purpose. It is usually done after system testing.

5. **Regression Testing**: It is the process of testing the software after making changes or updates to ensure that the existing functionality is not impacted and no new issues are introduced.

6. **Performance Testing**: It is the process of testing the software to ensure that it performs well under various loads and conditions.

RAGHU

7. **Security Testing**: It is the process of testing the software to identify vulnerabilities and weaknesses that could be exploited by malicious users or hackers.

8. **Usability Testing**: It is the process of testing the software to ensure that it is easy to use and understand by end-users.

10. **Compatibility Testing**: It is the process of testing the software to ensure that it works as expected on different platforms, browsers, and devices.

Cloud Terminologies

1. Public Cloud: A cloud computing environment in which the infrastructure and services are owned and operated by a third-party provider, such as AWS, Microsoft Azure, or Google Cloud.

2. Private Cloud: A cloud computing environment in which the infrastructure and services are owned and operated by a single organization.

3. Hybrid Cloud: A cloud computing environment that combines both public and private cloud infrastructures, allowing workloads to be moved between them as needed.

4. Infrastructure as a Service (IaaS): A cloud computing model that provides virtualized computing resources, such as virtual machines, storage, and networking, on a pay-per-use basis.

5. Platform as a Service (PaaS): A cloud computing model that provides a platform for developing, testing, and deploying applications, without the need to manage the underlying infrastructure.

6. Software as a Service (SaaS): A cloud computing model that delivers software applications over the internet on a subscription basis, without the need for local installation or management.

7. Cloud Native: An approach to software development and deployment that utilizes cloud computing principles, such as containerization, microservices, and automation.

8. Virtualization: A technique for creating virtual versions of computing resources, such as servers, storage, and networking, enabling multiple workloads to share a single physical infrastructure.

RAGHU

9. Containerization: A lightweight form of virtualization that encapsulates an application and its dependencies in a container, making it portable and easier to deploy across different environments.

10. Serverless Computing: A cloud computing model in which the cloud provider manages the infrastructure and automatically scales resources in response to application demands, enabling developers to focus on code rather than infrastructure management.

11. Multi-Cloud: A strategy that involves using multiple cloud providers to avoid vendor lock-in and increase resilience and flexibility.

12. Cloud Migration: The process of moving applications, data, and other workloads from on-premises infrastructure to the cloud.

13. Cloud Security: The set of practices, technologies, and policies used to protect cloud computing environments and data from threats, such as unauthorized access, data breaches, and cyber attacks.

14. Auto Scaling: A feature of cloud computing platforms that automatically adjusts the number of computing resources based on application demands, enabling cost savings and improved performance.

15. Elasticity: The ability of cloud computing environments to dynamically allocate and release computing resources in response to changing application demands.

RAGHU

UNIX Terminologies

1. Shell: A command-line interpreter that provides a way to interact with the UNIX system.

2. Terminal: A graphical or text-based interface that allows users to enter commands and interact with the Unix system.

3. Process: An instance of a program that is being executed by the Unix system.

4. File system: The structure used by Unix to organize and store files and directories.

5. User: A person who is authorized to access the Unix system.

6. Root: The superuser account on a Unix system with complete control over the system.

7. Permissions: The access control mechanism used by Unix to restrict access to files and directories.

8. Environment variables: Variables that are set and used by the Unix system to configure various aspects of the system.

9. Pipes: A mechanism for connecting the output of one command to the input of another command.

10. Redirection: A mechanism for directing the input or output of a command to or from a file or device.

11. Shell scripts: A script written in a Unix shell programming language that can be executed on the Unix system.

12. Daemon: A background process that runs continuously on the Unix system, providing services to other processes or users.

13. Kernel: The core of the Unix operating system that provides low-level services to other parts of the system.

Networking Terminologies

1. IP Address: Internet Protocol Address is a unique numerical identifier assigned to each device connected to a network that uses the Internet Protocol for communication.

2. DNS: Domain Name System is a hierarchical decentralized naming system that translates domain names to IP addresses.

3. Router: A device that connects two or more networks together and routes network traffic between them.

4. Firewall: A security system that controls the incoming and outgoing network traffic based on predefined security rules.

RAGHU

5. VPN: Virtual Private Network is a secure connection between two networks or a remote user and a network over the internet.

6. Switch: A network switch is a device that connects devices together on a local area network (LAN), enabling communication between them.

7. Gateway: A gateway is a networking device that provides a bridge between different networks, allowing data to be transferred from one network to another.

8. Subnet: A subnet is a logical subdivision of an IP network that allows multiple networks to share a single IP address.

9. LAN: Local Area Network is a computer network that covers a small area such as an office, building, or campus.

10. WAN: Wide Area Network is a network that covers a large geographical area such as a city, country or even the whole world.

11. DHCP: Dynamic Host Configuration Protocol is a network management protocol that automatically assigns IP addresses and other network configuration settings to devices on a network.

12. NAT: Network Address Translation is a method of remapping one IP address space into another by modifying network address information in the IP header while in transit.

13. VLAN: Virtual Local Area Network is a group of devices on one or more LANs that are configured to communicate as if they are on the same physical LAN.

14. OSI Model: The Open Systems Interconnection Model is a conceptual model that characterizes and standardizes the communication functions of a telecommunication or computing system.

AWS Terminology

1. Regions: AWS operates in multiple geographically distributed regions around the world, each consisting of multiple Availability Zones. AWS customers can select the region closest to their users to improve latency and performance.

RAGHU

2. Availability Zones: Availability Zones are data centers located within a region that are physically separate from each other and connected by high-speed, low-latency networks. Each Availability Zone is designed to be isolated from failures in other zones, providing customers with high availability and fault tolerance.

3. Edge Locations: Edge Locations are points of presence for Amazon's CloudFront content delivery network (CDN). They are used to cache content closer to users to reduce latency and improve performance.

4. Local Zones: Local Zones are a new type of AWS infrastructure deployment that provides low-latency access to AWS services in select metropolitan areas.

5. EC2 (Elastic Compute Cloud): EC2 provides scalable compute capacity in the cloud. It enables customers to launch and manage virtual machines, or instances, on-demand.

6. AMI (Amazon Machine Image): An AMI is a pre-configured virtual machine image used to create EC2 instances. It contains the operating system, application server, and any additional software needed to run an application.

7. S3 (Simple Storage Service): S3 provides object storage for files and data. It enables customers to store and retrieve data from anywhere on the web, and at any time.

8. EBS (Elastic Block Store): EBS provides block-level storage volumes for use with EC2 instances. It enables customers to store and retrieve data persistently from their instances.

9. EFS (Elastic File System): EFS provides scalable file storage for use with EC2 instances. It enables customers to create and mount file systems to their instances.

10. RDS (Relational Database Service): RDS provides managed database services. It enables customers to deploy, manage, and scale relational databases in the cloud.

11. DynamoDB: DynamoDB is a NoSQL database service. It enables customers to store and retrieve any amount of data, at any level of throughput.

RAGHU

12. EKS (Elastic Kubernetes Service): EKS provides managed Kubernetes clusters for use with containerized applications. It enables customers to easily deploy, manage, and scale containerized applications on AWS.

13. IAM (Identity and Access Management): IAM enables customers to manage access to AWS resources and services. It enables customers to create and manage users, groups, and roles to control who can access resources and services.

14. SNS (Simple Notification Service): SNS is a messaging service that sends notifications to subscribed endpoints or clients. It enables customers to send messages to multiple recipients or clients at once.

15. SQS (Simple Queue Service): SQS is a message queuing service that enables decoupling of application components. It enables customers to send, store, and receive messages between software components, without losing messages or requiring other components to be available.

16. SES (Simple Email Service): SES is a cloud-based email sending service. It enables customers to send and receive email using AWS infrastructure.

17. CloudWatch: CloudWatch is a monitoring service for AWS resources and applications. It enables customers to monitor and collect metrics, collect and track log files, and set alarms.

18. VPC (Virtual Private Cloud): VPC enables customers to create a virtual network within AWS. It enables customers to launch Amazon resources into a virtual network that they define.

19. Elastic Beanstalk: Elastic Beanstalk is a platform as a service (PaaS) that enables developers to deploy and manage applications in the cloud. It enables customers to quickly deploy and manage applications without having to worry about the infrastructure.

20. Lambda: Lambda runs code in response to events and automatically manages compute resources. It enables customers to run code without

21. ECS (Elastic Container Service) and EKS (Elastic Kubernetes Service) are both container orchestration services offered by AWS.

RAGHU

DevOps Terminologies

1. Continuous Integration (CI): A practice of frequently merging code changes to a central repository, which is then automatically built and tested.
2. Continuous Delivery (CD): A practice of automatically deploying code changes to production or a production-like environment after passing automated tests.
3. Continuous Deployment: A practice of automatically deploying code changes to production without any human intervention.
4. Configuration Management: A process of managing and automating the configuration of software and infrastructure components.
5. Infrastructure as Code (IaC): A practice of using code to manage and automate the provisioning of infrastructure.
6. DevOps Culture: A culture that emphasizes collaboration, communication, and integration between development and operations teams.
7. Microservices: A software architecture style that structures an application as a collection of small, independent services that communicate through APIs.
8. Agile: An iterative approach to software development that emphasizes flexibility, collaboration, and customer satisfaction.
9. Version Control: A system for tracking changes to code and other files over time, enabling collaboration among multiple developers.
10. Release Management: A process of planning, scheduling, and controlling the deployment of software changes to production or other environments.

RAGHU

Deployment Servers

1. Apache Tomcat: A widely-used Java application server that can be used for deploying Java web applications.

2. GlassFish: An open-source Java EE application server that can be used for deploying Java web applications.

3. JBoss: A popular Java EE application server that can be used for deploying Java web applications.

4. WildFly: A lightweight, fast, and flexible Java EE application server that can be used for deploying Java web applications.

5. Node.js: A JavaScript runtime built on the V8 JavaScript engine that allows developers to run JavaScript on the server-side. It can be used to deploy Node.js applications.

6. Nginx: A high-performance open-source web server used for serving static and dynamic content over the internet. It can also be used as a reverse proxy and load balancer. Nginx is often used with dynamic languages such as PHP, Python, Ruby, and Node.js, as well as with web frameworks like Flask, Django, Ruby on Rails, and Express.js

7. Apache HTTP Server: A widely-used open-source web server that is used to serve static and dynamic content over the internet. It can also be used as a reverse proxy and load balancer.

8. Microsoft IIS: A web server specifically designed for Microsoft Windows operating systems, commonly used for ASP.NET and .NET Core applications.

9. **AWS Elastic Beanstalk**: A fully managed service for deploying web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, and Docker on AWS infrastructure.

10. **Heroku**: A cloud-based platform that allows developers to deploy, manage, and scale applications built with Ruby, Node.js, Java, Python, PHP, and Go.

11. Google App Engine: A fully managed platform for deploying and scaling web applications and services developed with Java, Python, PHP, Node.js, Ruby, and Go.

RAGHU

DevOps Tools

1. Jenkins: Jenkins is an open-source automation server that helps automate various stages of the software development lifecycle such as building, testing, and deploying code. It is written in Java and can be used to create pipelines of activities that automate the entire software development process.

2. Git: Git is a version control system that allows developers to manage source code and track changes to it over time. It is an open-source tool that is widely used in software development and is particularly useful for teams working on large and complex projects. With Git, developers can collaborate on code, track changes, and maintain different versions of the codebase.

3. Docker: Docker is a platform that allows developers to build, deploy, and run applications in containers. Containers are lightweight and portable, making it easier to deploy applications across different environments.

4. Ansible: Ansible is an open-source automation tool that helps automate the configuration and management of servers and applications. It uses a simple language called YAML to define tasks and playbooks, making it easier for developers to manage infrastructure as code.

5. Kubernetes: Kubernetes is an open-source container orchestration platform that helps manage and scale containerized applications. It automates tasks such as deployment, scaling, and management of containerized applications, making it easier to deploy and manage applications at scale.

6. Sonarqube: Sonarqube is an open-source tool that helps automate code quality checks, code review, and code analysis. It can be used to detect bugs, vulnerabilities, and security issues in code and provides insights into code complexity, maintainability, and test coverage.

7. Nagios: Nagios is an open-source monitoring tool that helps monitor infrastructure and applications for issues and sends alerts when problems arise. It can monitor various aspects of infrastructure, including servers, network devices, and applications.

8. Splunk: Splunk is a log management and analysis tool that helps analyze logs and identify issues and trends in application and infrastructure performance. It can be used to monitor logs, metrics, and other data sources and provides insights into application and infrastructure performance.

9. Terraform: Terraform is an open-source infrastructure as code tool that allows developers to define and manage infrastructure as code, making it easier to automate infrastructure deployment and management.

10. ELK: ELK is an open-source stack that combines Elasticsearch, Logstash, and Kibana. It is used for log management and analysis and provides a centralized platform for searching, analyzing, and visualizing logs and other data sources.

RAGHU