

Assignment - 2

07/2/2024
7

- ① Compare and Contrast Diversified Services provided by Microsoft Azure and Amazon AWS with their pricing models.

Service Category	Microsoft Azure	Description	Amazon AWS	Description
Compute Service	Azure Virtual Machines, AKS	Provides Scalable VMs & Container Orchestration	Amazon EC2	Elastic Compute Cloud offering VMs
Storage Services	Blob Storage, File Storage, Disk Storage	Object Storage, File Shares, and block storage	Amazon S3, EFS, EBS	Object Storage - S3, File System, block storage
Database Services	SQL Database, Cosmos DB, Database for MySQL	Managed Relational and NoSQL database	Amazon RDS, DynamoDB	Managed relational & NoSQL database
Networking Services	Virtual Network, Load balancer, VPN Gateway	Networking Infrastructure & Security	Amazon VPC, ELB, AWS VPN	Networking & load balancing services
AI & ML Services	Azure Machine Learning, Cognitive Services	AI & ML Capabilities	Amazon SageMaker, AWS AI/ML Services	ML & AI services
Pricing Models	Pay-as-you-go, Reserved Instances	Flexible pricing with reserved capacity	Pay-as-you-go, Spot Instances	Flexible pricing

Hybrid cloud	Azure Arc	Extend Azure services to on-premises & other cloud	AWS outposts	Extend AWS infrastructure to on-premises
Global reach	Global data center with multiple regions	Provides service in various geographical locations	Global data center with multiple region	Services available globally.

9 Explain Docker and microservices in detail and explain their deployment procedures.

Docker

* Docker is a containerization platform that allows developers to package and automate the deployment of applications inside lightweight, portable containers. Containers are isolated environments that encapsulate an application and its dependencies, ensuring consistency across different environments. Docker uses containerization technology to package & distribute applications along with their dependencies as containers.

Key Concepts

* Image - An image is a lightweight, standardized & executable package that includes everything needed to run a piece of software.

* Container - A Container is an instance, of a docker image.

* Dockerfile - It's a script that contains instructions for building docker image.

Advantages

- * Isolation
- * Portability
- * Resource Efficiency.

Deployment procedure

1. Write dockerfile - Create a dockerfile that defines the image, dependencies and configuration for your application.
2. Build docker image - Use the dockerfile to build a docker image using the docker build command.
3. Push image to Registry - Upload the build image to a container registry.
4. Pull image on Target environment - On the target machine or environment, pull the docker image using the docker pull command.
5. Run Container - Start the application by running a container from the pulled image using the docker run command.

Microservices.

10

Microservices is an architectural style that structures an application as a collection of small, independent services, each focused on a specific business capability. These services communicate with each other through APIs, enabling the development, deployment and scaling of individual components independently.

Key Concepts

1. Services - Microservices are small, self-contained services that are independently deployable and scalable.
2. APIs - Microservices communicate with each other through well-defined APIs.

Advantages

- * Scalability
- * Flexibility
- * Fault Isolation

Deployment procedure

11

to, ~~with~~ ~~microservice~~

1. Containerize microservices - package each microservice into a docker container following the docker deployment procedure.
2. orchestration - Use Container orchestration tools like Kubernetes and Docker Swarm to manage and deploy multiple containers.
3. Service discovery - Implement a service discovery mechanism to allow microservices to locate and communicate with each other.
4. Continuous Integration / Continuous deployment : set up CI/CD pipelines to automate the testing, building and deployment of microservices.
5. Monitoring and Scaling - Implement monitoring tools to track the performance of microservices.

③ Explain how Service based Companies are indirectly promoting / consuming services from big cloud under Azure, AWS, IBM. 12

Service based Companies often rely on cloud computing services from major providers like Azure, AWS, IBM and others. These cloud platforms offer a range of services, including infrastructure as a service, platform as a service and software as a service. Here how service based Companies indirectly promote and consume services from these big cloud providers

① Infrastructure as Service (IaaS) - Many service-based companies leverage the infrastructure service provided by cloud providers. This includes virtual machines, storage, and networking resources. By using cloud infrastructure, companies can scale their operations without need to invest heavily in physical hardware. This directly promotes the cloud providers IaaS offerings.

② Platform as Service (PaaS) - Cloud providers offer platforms that facilitate application development, deployment, and management. Service-based companies often use these platforms to build & deploy their applications more efficiently.

③ Software as Service (SaaS) - Service-based companies may consume various S/W services hosted on the cloud. Cloud providers offer a wide range of SaaS solutions such as Microsoft 365 on Azure, AWS SaaS applications solutions, IBM Cloud Pak for Applications.

- ④ Data Storage & Analytics - Cloud providers offer robust data storage & analytics services.¹³ Service-based companies can store and process large volumes of data using solutions like AWS S3, Azure Blob Storage or IBM Cloud Object Storage.
- ⑤ Machine Learning and AI services.
- ⑥ Security and Compliance
- ⑦ Global Reach and Scalability.

④ Explain how DevOps teams are using cloud and what is the reason for using cloud.

DevOps teams leverage cloud computing for various reasons, and the adoption of cloud services has become a key enabler of DevOps practices. Here are some key ways

① Infrastructure as Code (IaC):

- Usage - DevOps teams utilize cloud services to implement Infrastructure as Code practices. This involves defining and managing infrastructure configurations through code, allowing for automated and consistent provisioning of resources.

- Reason - Cloud platforms provide APIs and services that make it easier to automate the creation, modification and deletion of infrastructure components.

② Scalability & Elasticity -

Usage - Cloud platforms offer scalable & elastic¹⁴ resources that can be adjusted based on demand. DevOps teams design applications & infrastructure to scale horizontally.

Reason - Cloud services enable automatic scaling of resources, ensuring that applications can handle varying workloads efficiently.

③ Collaboration and Communication

Usage - cloud-based collaboration tools and communication platforms are often used by DevOps teams to enhance communication and collaboration among team members, especially in distributed or remote environments.

Reason - cloud-based collaboration tools such as Slack, Microsoft Teams or Google Workspace, provide real-time communication, file sharing and collaboration features.

④ Cost optimization

Usage - Cloud platforms provide tools for tracking and optimizing costs associated with infrastructure and services. DevOps teams use these tools to manage and optimize resource usage efficiently.

Reason - Cloud cost management tools allow teams to monitor spending, identify unused resources, and implement cost-saving strategies.

⑤ Monitoring and Logging

⑥ Continuous Integration and Continuous Deployment (CI/CD)