

UNIT 6

Cloud Platforms and Applications

(12 Hours) — Cloud Computing (CACS402)

Disclaimer: These notes were prepared as a study aid and may contain errors or omissions. Please cross-check important definitions, figures and product details (especially vendor services such as Google App Engine, Microsoft Azure and Aneka, which change often) against your prescribed textbook, class notes and official documentation before relying on them in an exam.

6.1 Web Services

A **web service** is a software component that exposes functionality over a network using standard Internet protocols so that different applications, written in different languages and running on different platforms, can interoperate. Web services are the building blocks of SOA and of cloud APIs.

Types of web services

Feature	SOAP (Big web services)	REST (RESTful services)
Nature	Protocol with strict standards (WS-*)	Architectural style over HTTP
Message format	XML only (envelope, header, body)	JSON, XML, plain text, etc.
Interface description	WSDL	OpenAPI/Swagger (optional)
Operations	Custom operations in the message	HTTP verbs: GET, POST, PUT, DELETE on resource URIs
State	Can be stateful	Stateless
Overhead	Heavier	Lightweight, easy to cache
Typical use	Enterprise systems, high-security transactions	Web, mobile and cloud APIs

Core standards

- **XML** — data format; **SOAP** — message exchange; **WSDL** — describes the service; **UDDI** — registry for discovery.
- **HTTP/HTTPS** — transport; **JSON** — compact data format used by REST.

Example 6.1 — A REST call

GET /students/101 → returns the record of student 101 in JSON.

POST /students → creates a new student. PUT /students/101 → updates. DELETE /students/101 → removes.

Each request is independent (stateless) and the URI identifies the resource.

In cloud computing, almost every service (storage, VM creation, machine-learning) is exposed as a web-service API, letting developers automate infrastructure and integrate services (mashups).

6.2 AppEngine

Google App Engine (GAE) is Google's Platform-as-a-Service for building and hosting web applications and back-ends on Google's infrastructure. Developers upload code and Google handles servers, load balancing, patching and automatic scaling. GAE was launched in 2008 and is now part of Google Cloud.

Environments

Environment	Description
Standard	Code runs in a sandbox with supported runtimes (e.g., Python, Java, Go, Node.js, PHP, Ruby); very fast scaling, including scale to zero; free daily quota.
Flexible	Applications run in Docker containers on Compute Engine VMs; supports custom runtimes and libraries; scaling is slower and no scale to zero.

Main features

- **Automatic scaling** up and down with traffic; load balancing and health checks.
- **Managed storage:** Cloud Datastore / Firestore (NoSQL), Cloud SQL, Cloud Storage.
- **Services:** Memcache (caching), Task Queues / Cloud Tasks, Cron jobs (scheduling), Mail, user authentication, and URL fetch for calling external services.
- **Versioning and traffic splitting:** deploy several versions and route a percentage of users for testing.
- **Development tools:** local development server, gcloud CLI, monitoring and logging.

Typical workflow

- Write the app (e.g., Python Flask) and an *app.yaml* configuration file.

- Test locally using the development server.
- Deploy with `gcloud app deploy`; App Engine gives a URL and scales automatically.

Advantages	Limitations
No server management; quick deployment.	Runtime and library restrictions (standard).
Automatic scaling and pay-per-use.	Vendor lock-in to Google services.
Integrated with Google Cloud services.	Less low-level control than IaaS.

Note: Google has updated App Engine's runtimes and APIs several times; check the current official documentation for supported languages and legacy APIs.

6.3 Azure Platform

Microsoft Azure is Microsoft's public cloud platform offering IaaS, PaaS and SaaS. It was announced in 2008 and became generally available in 2010 (initially "Windows Azure"). It has a global network of regions and availability zones.

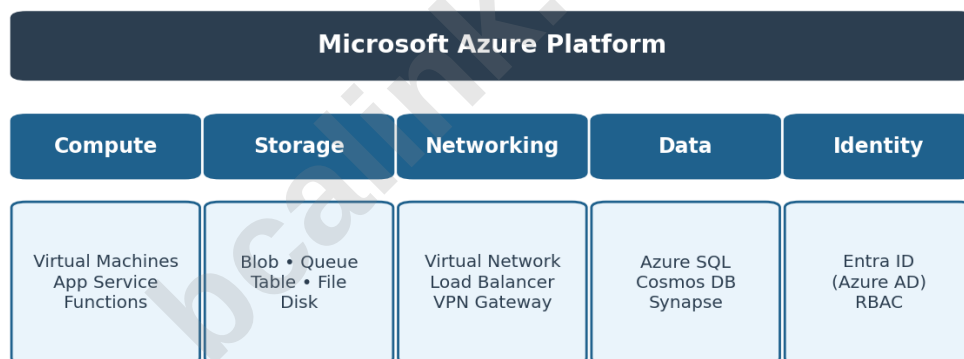


Figure 6.1 — Core service categories of Microsoft Azure

Core services

Category	Services	Purpose
Compute	Virtual Machines, App Service, Azure Functions, AKS (Kubernetes), Cloud Services (legacy)	Run applications, containers and serverless code.
Storage	Blob (objects), Queue (messages), Table (NoSQL), File and Disk storage	Durable, replicated storage.
Networking	Virtual Network, Load Balancer, Application Gateway, VPN Gateway, CDN	Connect, distribute and secure traffic.
Databases	Azure SQL Database, Cosmos DB, Database for MySQL/PostgreSQL	Managed relational and NoSQL data.
Identity and security	Microsoft Entra ID (Azure AD), RBAC, Key Vault, Defender for Cloud	Authentication, authorisation, secrets, monitoring.
Analytics and AI	Synapse, HDInsight, Databricks, Azure Machine Learning, Cognitive Services	Big data and machine learning.
Management/Dev Ops	Azure Monitor, Resource Manager (ARM), Azure DevOps	Deployment, monitoring, automation.

Key concepts

- **Subscription → Resource Group → Resources:** resources are organised and billed in resource groups.
- **Regions and availability zones** for resilience; **SLA-backed** services.
- **Hybrid support:** Azure Arc, Azure Stack, and ExpressRoute private connectivity to on-premises.
- **Pricing:** pay-as-you-go, reserved instances and spot pricing.

Example 6.2 — Hosting a web app on Azure

Deploy a web app with Azure App Service (PaaS), store user files in Blob Storage, keep data in Azure SQL Database, and authenticate users with Entra ID. Enable autoscale to add instances when CPU is high.

6.4 Aneka

Aneka is a .NET-based cloud application platform (PaaS) developed by Manjrasoft (originating from the CLOUDS Laboratory, University of Melbourne). It provides a software platform and programming models for building and deploying distributed applications on a mix of desktops, clusters, data centres and public clouds.

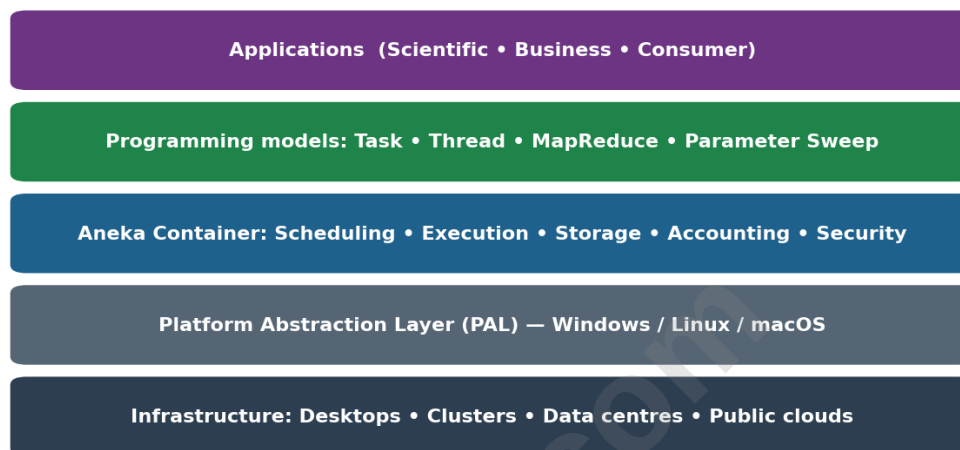


Figure 6.2 — Layered architecture of the Aneka platform

Key components

- **Aneka Container:** the lightweight runtime installed on every node; hosts services such as scheduling, execution, storage, accounting and security.
- **Master and worker nodes:** the master coordinates scheduling; workers execute tasks.
- **Platform Abstraction Layer (PAL):** hides differences between operating systems.
- **Management tools:** Aneka Cloud Management Studio / web console to configure and monitor the cloud.
- **Dynamic provisioning:** can acquire extra resources from public clouds (e.g., Amazon EC2, Azure) when local capacity is insufficient — a hybrid cloud approach.

Programming models

Model	Description	Use
Task programming	Independent tasks, each with its own input/output, executed in parallel.	Bag-of-tasks, simulations
Thread programming	Distributed threads that mimic local multithreading.	Fine-grained parallel apps
MapReduce programming	Map and reduce operations over large data.	Data analysis
Parameter sweep	Same application run over many parameter combinations.	Scientific experiments

Example 6.3 — Aneka use

A genomics lab has 50 idle desktop PCs. Using Aneka's task model, each PC processes a batch of sequences; when a deadline approaches, Aneka acquires extra virtual machines from a public cloud to finish sooner.

Note: Aneka's features and licensing are defined by Manjrasoft; confirm current details with the official documentation.

6.4 (b) Open Challenges

Despite rapid growth, cloud computing still faces several unresolved technical and business challenges:

Challenge	Explanation
Security, privacy and trust	Protecting data in multi-tenant environments; regulatory compliance.
Interoperability and portability	Lack of common standards makes moving between providers difficult (vendor lock-in).
Service Level Agreements (SLAs)	Defining, measuring and enforcing performance, availability and penalties.
Performance and scalability	Variable performance, network latency, bandwidth cost for data-intensive apps.
Energy efficiency	Data centres consume large power; need green computing and efficient scheduling.
Resource management and scheduling	Efficient VM placement, load balancing, autoscaling and cost optimisation.
Reliability and fault tolerance	Outages and cascading failures; disaster recovery.
Legal and ethical issues	Data sovereignty, jurisdiction, liability, e-discovery.

Challenge	Explanation
Standardisation and billing models	Different pricing models and unpredictable costs.
Skills and organisational change	Shortage of cloud expertise; migration of legacy applications.

Emerging directions: edge and fog computing, serverless computing, multi-cloud, confidential computing and AI-driven cloud management.

6.5 Scientific Applications

Cloud platforms offer scientists elastic access to large computing resources without owning clusters. Typical scientific domains and cloud use:

Domain	Application	Cloud benefit
Healthcare and life sciences	Genome sequencing, protein folding, medical imaging analysis, drug discovery	Burst capacity for large batch jobs.
Bioinformatics / ECG analysis	Processing patient signals and sequence databases	Parallel task execution and data storage.
Astronomy and physics	Telescope data processing, particle-physics event analysis (e.g., CERN)	Massive data storage and parallel processing.
Geoscience and climate	Satellite image processing, weather forecasting, earthquake simulation	HPC-style workloads on demand.
Chemistry and materials	Molecular simulations	Parameter sweeps on many nodes.
Engineering	CFD, structural simulation, Monte-Carlo methods	Pay-per-use high-performance computing.

- Common patterns: **bag-of-tasks / parameter sweep**, **MapReduce** data analysis, workflow engines and scientific gateways.
- Advantages: no upfront cluster cost, easy sharing of data and results, reproducibility using VM images.
- Challenges: data-transfer time, costs of long-running jobs, tightly coupled MPI applications may run slower than on dedicated supercomputers.

6.6 Business and Consumer Applications

Business applications

Area	Cloud examples
CRM (Customer Relationship Management)	Salesforce, Zoho CRM, Microsoft Dynamics 365
ERP and finance/accounting	SAP S/4HANA Cloud, Oracle NetSuite, QuickBooks Online
Productivity and collaboration	Microsoft 365, Google Workspace, Slack, Zoom
Human resources and payroll	Workday, BambooHR
E-commerce and marketing	Shopify, Amazon Web Services-hosted stores, HubSpot
Business intelligence and analytics	Power BI, Tableau Cloud, BigQuery
Backup and disaster recovery, hosting	AWS, Azure Site Recovery

- Benefits for business: lower IT cost, faster time to market, remote work support, easy scaling for seasonal demand and global reach.

Consumer applications

Category	Examples
Email and communication	Gmail, Outlook.com, WhatsApp, Zoom
Storage and file sharing	Google Drive, iCloud, Dropbox, OneDrive
Media streaming	Netflix, YouTube, Spotify
Social networking	Facebook, Instagram, X (Twitter), LinkedIn
Online office and education	Google Docs, Canva, Google Classroom
Gaming	Cloud gaming services (e.g., GeForce NOW, Xbox Cloud Gaming)
Personal assistants and photos	Google Photos, Alexa, Siri

Example 6.4 — Netflix on the cloud

Netflix runs much of its service on Amazon Web Services: content is encoded in the cloud, stored in object storage, and delivered through a content-delivery network; compute scales up during evening peak hours and back down at night.

Key Takeaways — Unit 6

- ✓ Web services (SOAP/WSDL/UDDI and REST/JSON) let heterogeneous applications interoperate; cloud APIs are web services.
- ✓ Google App Engine is a PaaS with standard and flexible environments, automatic scaling and managed services.
- ✓ Microsoft Azure offers compute, storage, networking, databases, identity, analytics and AI across global regions.
- ✓ Aneka is a .NET PaaS with a container-based architecture supporting task, thread, MapReduce and parameter-sweep models and hybrid provisioning.
- ✓ Open challenges: security, interoperability, SLAs, energy efficiency, resource management and legal issues.
- ✓ Applications span science (genomics, astronomy, climate), business (CRM, ERP, collaboration) and consumer services (email, storage, streaming).

— *End of Unit 6* —