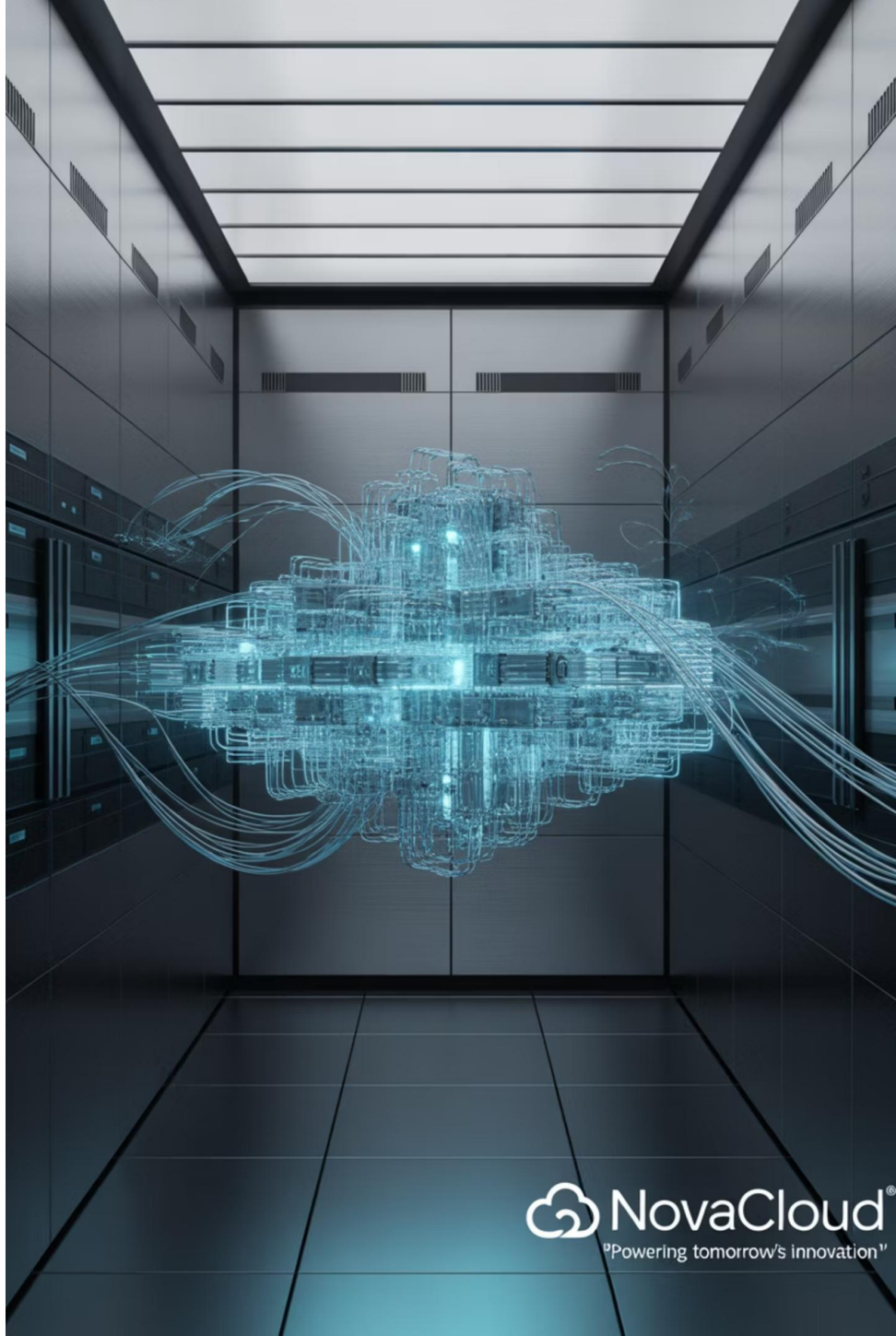
Foundations of Cloud Computing for Technical Project Managers

A comprehensive guide to cloud architecture, services, and implementation strategies for TPMs



Module 1: Cloud 101 – Core Concepts

Duration: 1 Hour

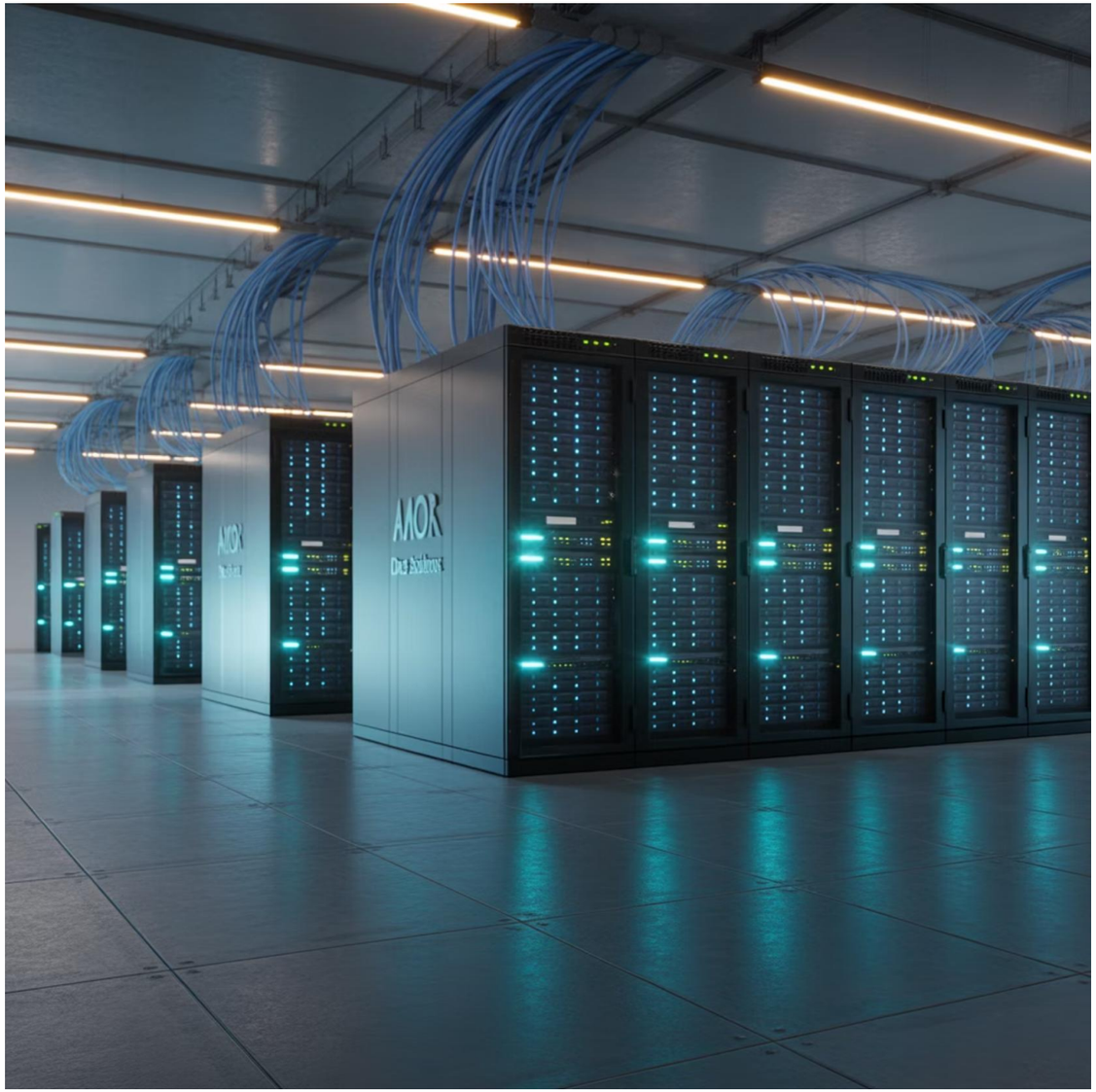
A foundational overview of cloud computing principles, service models, and infrastructure elements essential for technical project managers.

What is Cloud Computing?

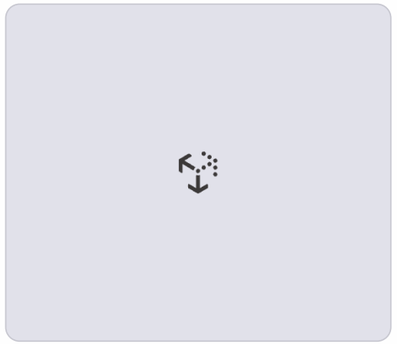
Cloud computing is the on-demand delivery of computing resources—applications, storage, databases, networking—over the internet with pay-as-you-go pricing.

Key Characteristics:

- On-demand self-service
- Broad network access
- Resource pooling
- Rapid elasticity
- Measured service



Service Models: The Cloud Computing Stack

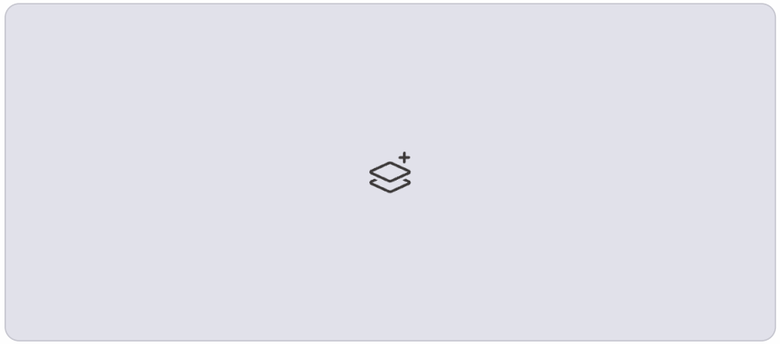


Software as a Service (SaaS)

Complete applications delivered over the web

Examples: Microsoft 365, Salesforce, Google Workspace

Benefit: Zero infrastructure management

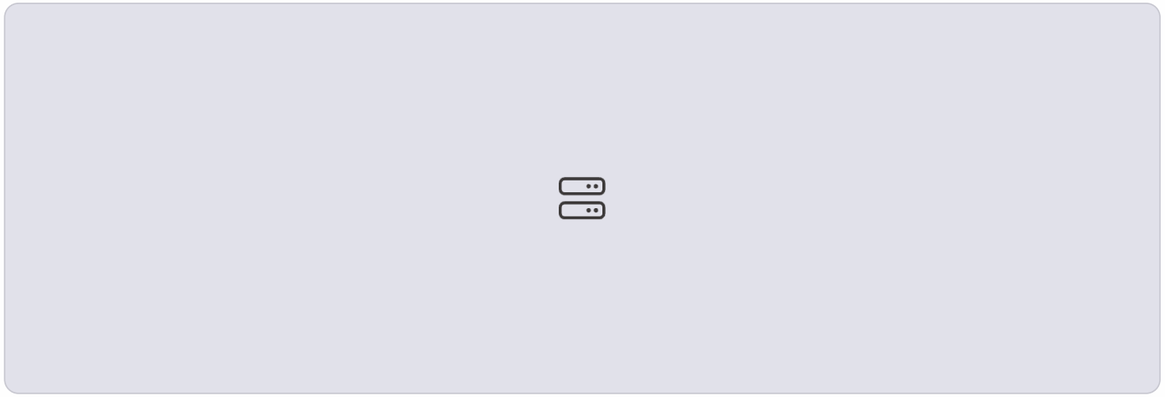


Platform as a Service (PaaS)

Development and deployment environment in the cloud

Examples: Google App Engine, Azure App Services, Heroku

Benefit: Focus on application development without managing infrastructure



Infrastructure as a Service (IaaS)

Virtualized computing resources over the internet

Examples: AWS EC2, Google Compute Engine, Azure VMs

Benefit: Complete control over infrastructure without physical maintenance

Major Cloud Providers: Azure & GCP

Microsoft Azure

- Strong enterprise integration with Microsoft products
- Comprehensive hybrid cloud capabilities
- 60+ regions worldwide
- Integrated DevOps and AI services
- Strong in government and enterprise sectors

Google Cloud Platform (GCP)

- Superior data analytics and machine learning
- Pioneer in Kubernetes and containers
- Global network with 35+ regions
- Strength in big data processing
- Advanced security and privacy controls

Cloud Deployment Models

Public Cloud

Infrastructure owned and operated by third-party providers, delivered over the internet

- Low initial investment
- Highly scalable
- Provider handles maintenance
- Shared resources

Private Cloud

Cloud infrastructure dedicated to a single organization, either on-premises or provider-hosted

- Enhanced control and security
- Customizable architecture
- Compliance-friendly
- Higher initial investment

Hybrid Cloud

Combination of public and private clouds with orchestration between platforms

- Flexibility and agility
- Workload optimization
- Data sovereignty control
- Complex implementation

Global Infrastructure: Building Blocks

Regions

Geographic areas containing multiple data centers

- Data sovereignty compliance
- Disaster recovery planning
- Lower latency for local users

Zones

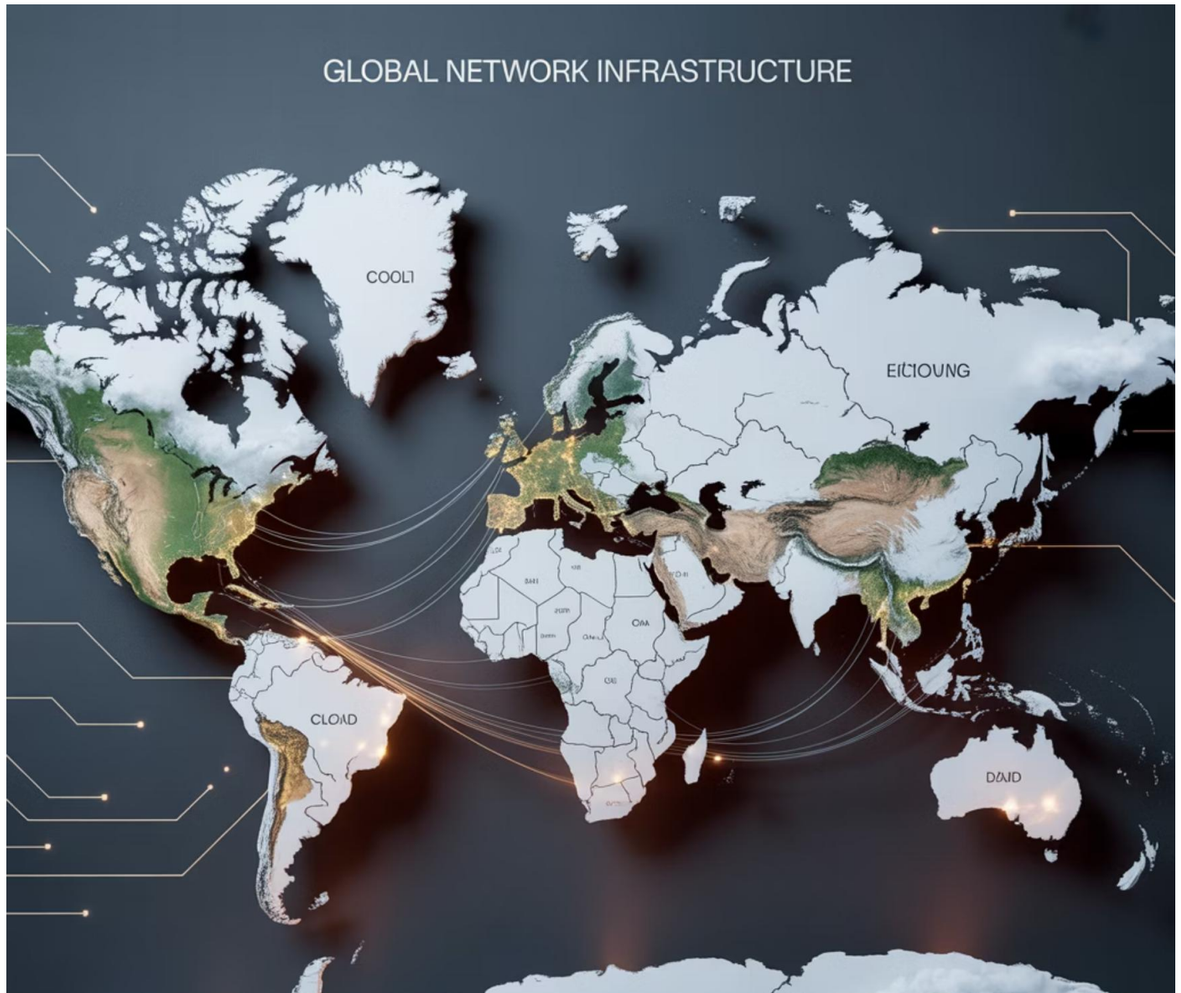
Isolated locations within a region with independent power, cooling, and networking

- High availability within region
- Protection from infrastructure failures

Edge Locations

Mini data centers closer to end users

- Content delivery acceleration
- Reduced latency for applications
- IoT data processing

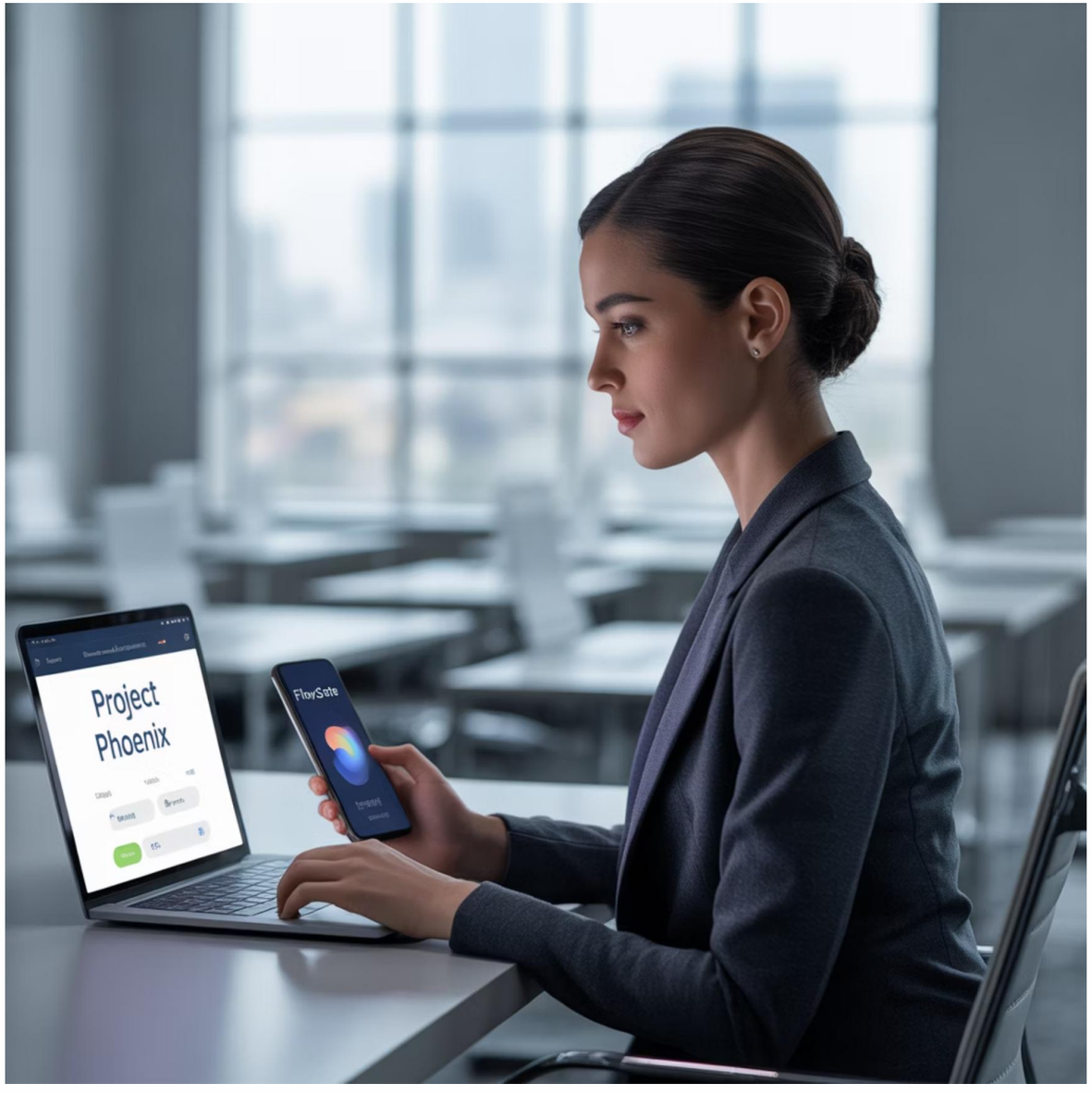


Activity: "Cloud in Your Life"

List 5 applications or tools you use daily. Let's classify each under IaaS, PaaS, or SaaS.

Common Examples:

Gmail/Outlook	SaaS
Slack/Teams	SaaS
Salesforce	SaaS
Company Website	PaaS/IaaS
Netflix/Spotify	SaaS (built on IaaS)



Case Study: Netflix Cloud Adoption Journey

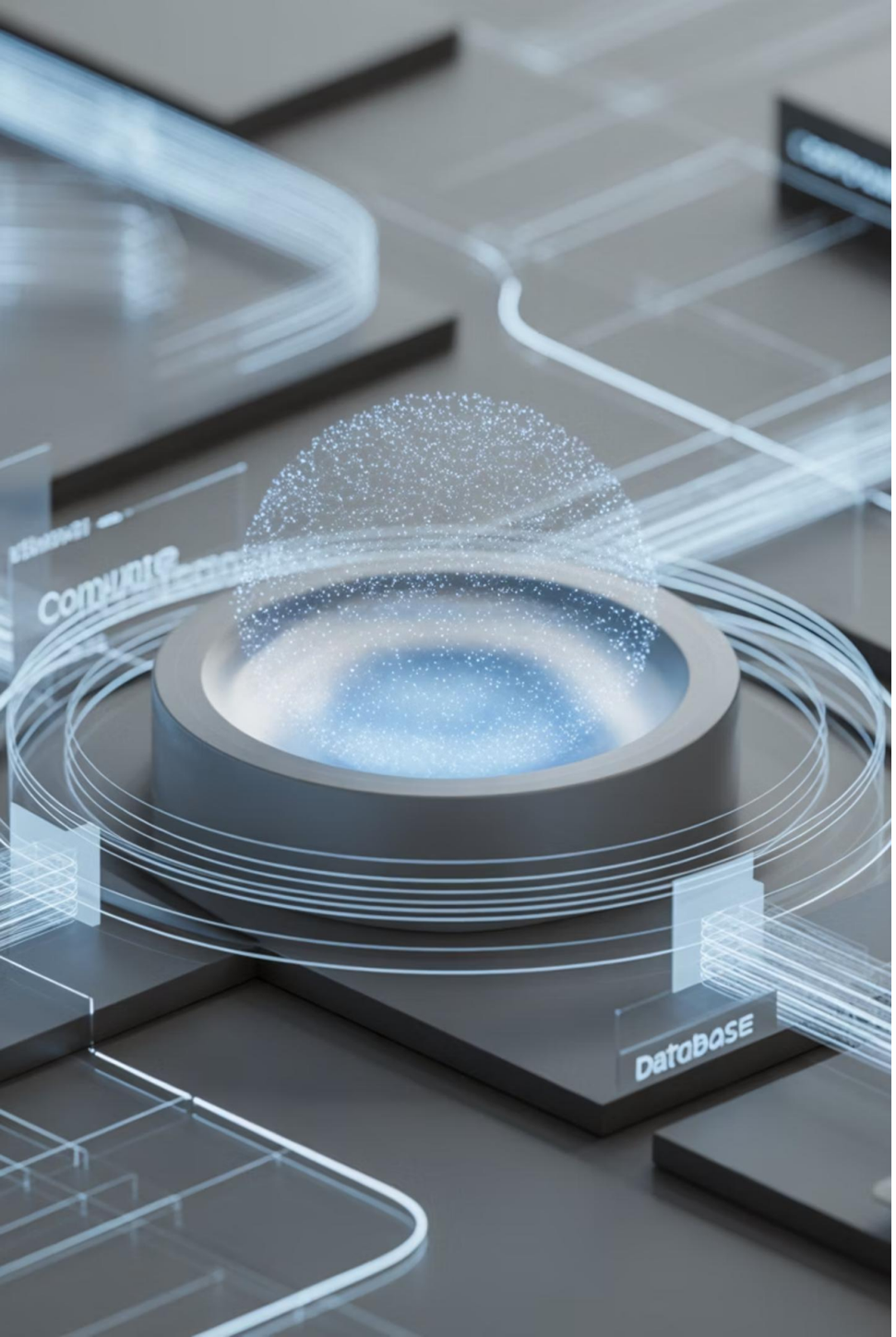
Netflix migrated from traditional data centers to AWS, becoming one of the most prominent cloud-native success stories.

Key Elements:

- Complete migration to AWS by 2016
- Leverages multiple AWS regions for global reach
- Uses availability zones for fault tolerance
- Implements chaos engineering for resilience
- Processes 100+ billion events daily

Service Model Usage:

- **IaaS:** EC2 for compute workloads
- **PaaS:** DynamoDB for metadata
- **SaaS:** Various monitoring tools



Module 2: Key Cloud Services

Duration: 1.5 Hours

An exploration of essential cloud services across major providers, focusing on compute, storage, and database offerings that form the backbone of cloud applications.

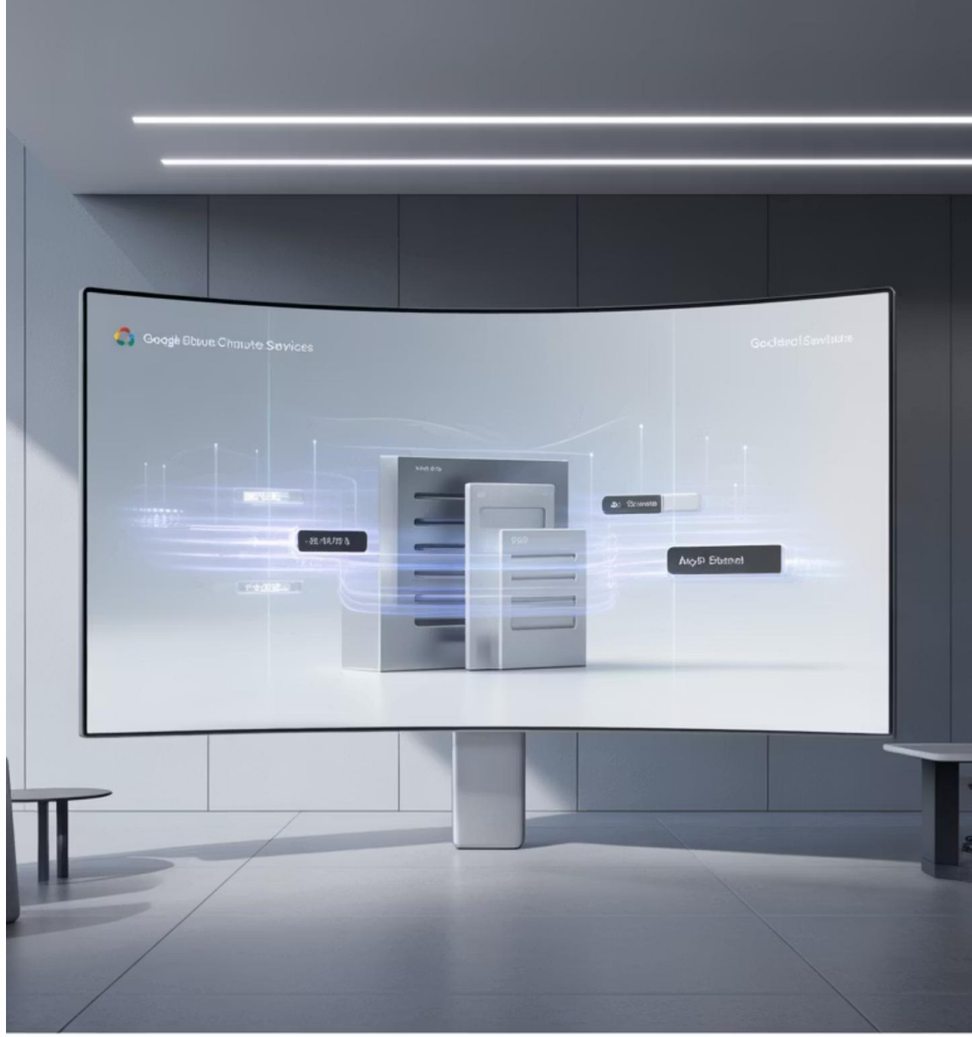
Compute Services: GCP

Google Compute Engine (GCE) - IaaS

- Virtual machines in Google's data centers
- Custom machine types for specific workloads
- Per-second billing with sustained use discounts
- Live migration of VMs during maintenance
- Global load balancing capabilities

Google App Engine (GAE) - PaaS

- Fully managed serverless platform
- Automatic scaling and load balancing
- Support for multiple programming languages
- Versioning and traffic splitting
- Integration with GCP services



Powering Tomorrow's Innovation

Google Cloud

Compute Services: Azure

Azure Virtual Machines - IaaS

- On-demand scalable computing resources
- Windows and Linux VM options
- Extensive range of VM sizes for different workloads
- Integration with Azure Active Directory
- Hybrid connectivity options

Azure App Services - PaaS

- Build and host web apps, mobile backends, RESTful APIs
- Auto-scale based on demand or schedule
- CI/CD integration with GitHub, Azure DevOps
- Managed production environment
- Authentication and authorization capabilities

Storage Services

Azure Blob Storage

- Unstructured data storage (images, documents, videos)
- Three access tiers: Hot, Cool, Archive
- Data redundancy options
- Content delivery network integration
- Security features: encryption, access control

Google Cloud Storage (GCS)

- Object storage for any data type
- Four storage classes: Standard, Nearline, Coldline, Archive
- Global edge caching
- Strong consistency model
- Lifecycle management policies

Database Services

AWS RDS

Managed relational database service supporting multiple database engines

- MySQL, PostgreSQL, Oracle, SQL Server
- Automated backups and patching
- Read replicas for performance

AWS DynamoDB

Fully managed NoSQL database service

- Single-digit millisecond performance
- Automatic scaling
- Multi-region, multi-master capability

GCP Cloud SQL

Fully managed relational database service

- MySQL, PostgreSQL, SQL Server
- Automatic replication
- Integrated with GCP services

Additional Cloud Services

Load Balancers

Distribute incoming traffic across multiple resources

- Azure Load Balancer/Application Gateway
- Google Cloud Load Balancing
- Health checks and automatic failover

Autoscaling

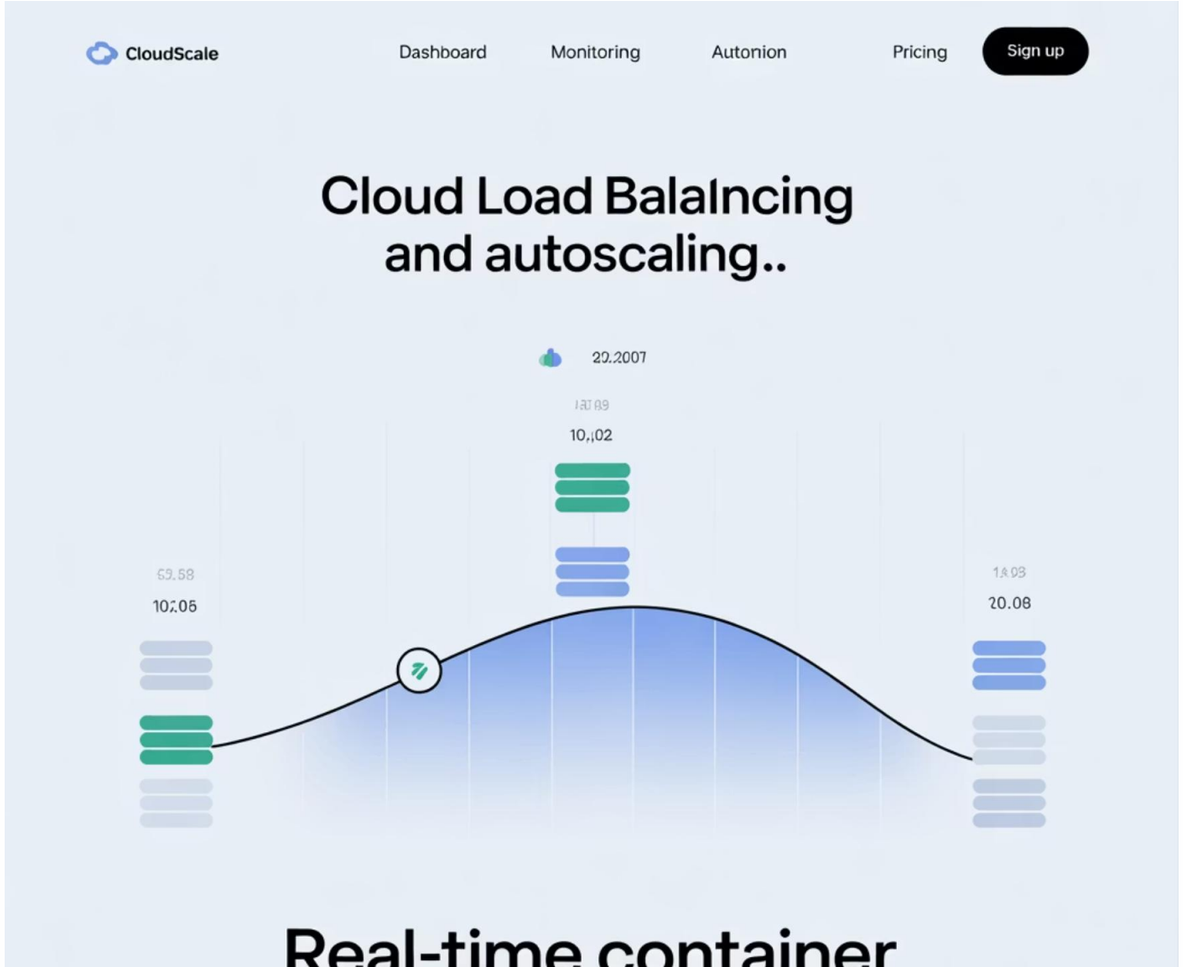
Automatically adjust resource capacity based on demand

- Azure Autoscale
- Google Cloud Autoscaler

Container Services

Orchestration platforms for containerized applications

- Google Kubernetes Engine (GKE)
- Azure Kubernetes Service (AKS)
- Automated deployment, scaling, management
- Integration with CI/CD pipelines



Module 3: Cloud-Native vs On-Prem vs Hybrid

Duration: 1.5 Hours

Comparing deployment strategies, understanding migration pathways, and evaluating the financial implications of different cloud models.

Characteristics of Cloud-Native Applications

Microservices Architecture

Decomposed into small, independent services that communicate via APIs

Containerization

Packaged with dependencies for consistent deployment across environments

DevOps Automation

CI/CD pipelines for continuous integration and deployment

Dynamic Scaling

Elastic resources that scale up or down based on demand

API-First Design

Services communicate through well-defined interfaces

Resilience & Self-Healing

Designed to handle failures and recover automatically

Deployment Model Comparison

On-Premises

Strengths:

- Complete control over infrastructure
- Data locality and physical security
- Predictable long-term costs
- No internet dependency

Weaknesses:

- High capital expenditure
- Limited scalability
- Maintenance overhead
- Disaster recovery challenges

Cloud-Native

Strengths:

- Rapid provisioning and scaling
- Pay-as-you-go pricing
- Global reach and redundancy
- Access to managed services

Weaknesses:

- Potential vendor lock-in
- Less control over infrastructure
- Internet dependency
- Compliance challenges

Hybrid

Strengths:

- Flexibility to optimize workloads
- Balance between control and agility
- Gradual migration path
- Data sovereignty control

Weaknesses:

- Complex infrastructure management
- Security integration challenges
- Skills gap across environments
- Potential network bottlenecks

Use Cases for Each Deployment Model

On-Premises Best For:

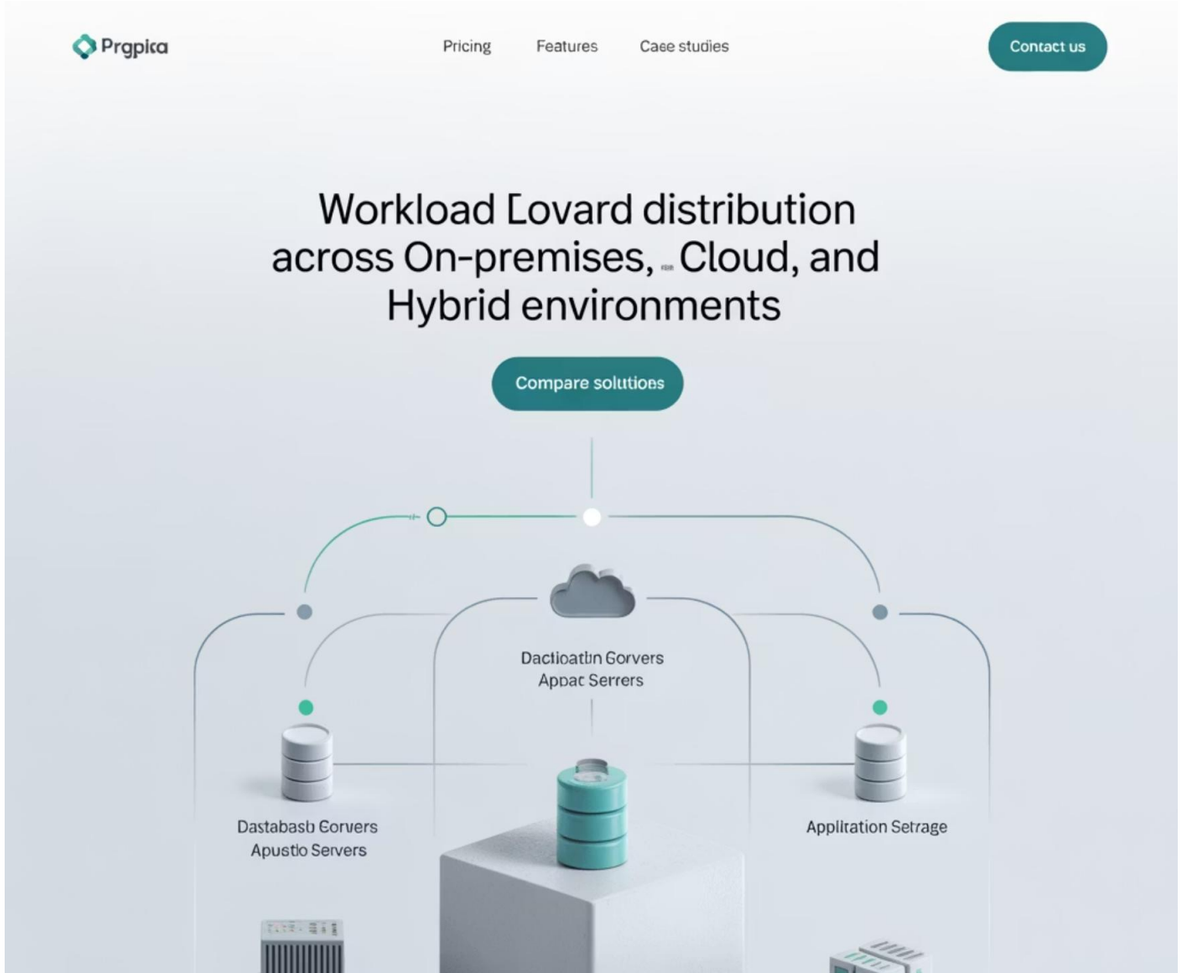
- Highly regulated industries with strict data residency requirements
- Legacy applications difficult to refactor
- Specialized hardware dependencies
- Predictable, stable workloads

Cloud-Native Best For:

- Startups with limited capital
- Variable or unpredictable workloads
- Global customer base requiring low latency
- Development and test environments

Hybrid Best For:

- Enterprises in transition to cloud
- Organizations with mixed workload requirements
- Disaster recovery strategies
- Bursting capacity during peak periods
- Edge computing needs with central processing



TCO vs ROI: The TPM Perspective

Total Cost of Ownership (TCO)

Complete assessment of all costs associated with an IT asset over its lifecycle

- Direct costs: hardware, software, operational expenses
- Indirect costs: training, downtime, security management
- Cloud calculators from providers help estimate TCO

Return on Investment (ROI)

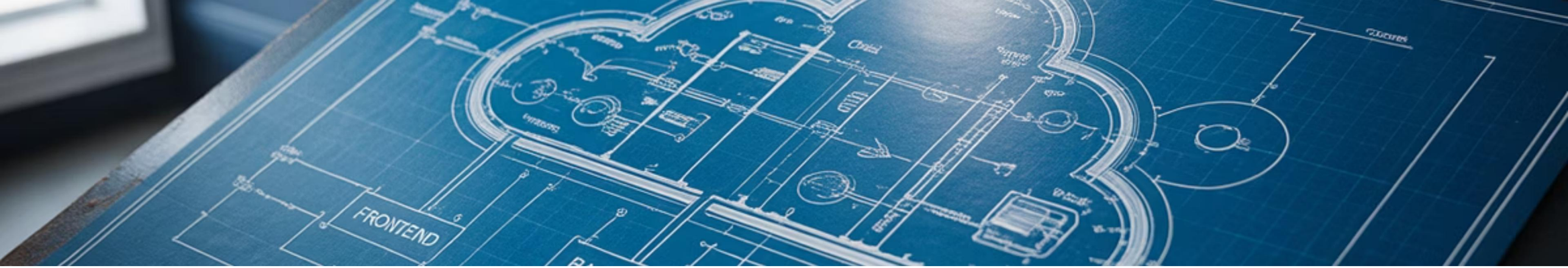
Measurement of the value gained relative to the investment made

- Time-to-market acceleration
- Innovation enablement
- Competitive advantage
- Business agility and scalability

TPM Decision Framework

Balanced approach to evaluating cloud investments

- Align with business objectives
- Consider both financial and non-financial factors
- Account for risk and compliance requirements
- Plan for evolution and future needs



Module 4: Hands-On Lab – Web App Architecture

Duration: 1.5 Hours

Apply your knowledge by designing a basic web application architecture using cloud-native services from either GCP or Azure.

Lab: Design a Basic Web App Architecture

Objective:

Design a scalable, cloud-native web application architecture that includes:

- Static frontend hosted on cloud storage/CDN
- Backend compute service for business logic
- Database for persistent storage
- Load balancing and monitoring

Team Exercise:

"Whiteboard above request”

Considerations:

- Cost optimization strategies
- Security best practices
- Scalability requirements
- Geographic distribution
- Monitoring and observability

Migration Pathways

Assessment & Discovery

Inventory applications, dependencies, and infrastructure. Establish migration goals and constraints.

Strategy Selection

Choose migration approach: Rehost (lift and shift), Replatform (lift and optimize), Refactor (rearchitect), Rebuild, or Replace.

Pilot Migration

Start with non-critical applications to validate approach and build team expertise.

Full Migration

Execute migration waves based on application groups. Implement testing and validation at each stage.

Optimization

Refine architecture, implement cloud-native features, optimize costs, and enhance security.

Cloud Cost Models

Pay-as-you-go

Most common pricing model with charges based on actual resource consumption

- No upfront costs
- Charges for compute by the second/minute
- Storage billed by GB-month
- Network transfer fees for outbound traffic

Reserved Instances

Commit to using resources for 1-3 years in exchange for discounts

- Savings of 40-60% compared to on-demand
- Different payment options (all upfront, partial, none)
- Best for predictable, stable workloads

Spot Instances

Use spare capacity at steep discounts (up to 90%)

- Can be terminated with minimal notice
- Best for fault-tolerant, flexible workloads
- Examples: batch processing, CI/CD pipelines

Activity: "Cloud Trade-Offs Roundtable"

Debate the best deployment model for these real-world scenarios:

1

Scenario 1: Healthcare Provider

Requirements:

- Patient data with strict regulatory compliance (HIPAA)
- Legacy system integration needs
- Budget constraints with preference for OpEx over CapEx

Discussion Points: Data sovereignty, security controls, integration challenges

2

Scenario 2: Global E-Commerce

Requirements:

- Customers across multiple continents
- Seasonal traffic spikes (10x during holidays)
- Real-time inventory management

Discussion Points: Latency, scalability, global data replication

3

Scenario 3: Financial Services

Requirements:

- High-frequency trading platform requiring ultra-low latency
- Strict data residency requirements by country
- Need for both security and innovation

Discussion Points: Performance, compliance, competitive advantage