

Case Studies: Aneka, Coment Cloud, T-systems, AWS.

### \* Coment Cloud Architecture:

=> The ComentCloud Architecture integrates various layers to provides a robust platform for developing and managing autonomic applications in Cloud environments.

This are the main layer of the ComentCloud.

#### 1 Programming Layer:

This Layer serves as the Framework for application development.

#### - Master / BOT Paradigm:

A task distribution model where master generates task and worker execute them.

#### - Task scheduling and Monitoring:

Support scheduling tasks and monitores.

- Task Consistency Management: Implements strategies to handle lost task.
- Support For Workflow - Based Applications,
- Integration with MapReduce and Hadoop.

|                            |                                | Application      |                     |
|----------------------------|--------------------------------|------------------|---------------------|
| Programming Layer          | Master/BoT                     | Task             | Workflow            |
|                            | Scheduling                     | Consistency      | MapReduce / Hadoop  |
| Service Layer              | Clustering / Anomaly Detection | Coordination     | Publish / Subscribe |
|                            | Discovery                      | Event            | Messaging           |
| Infrastructure Layer       | Replication                    | Load Balancing   |                     |
|                            | Content-Based Routing          | Content Security |                     |
|                            | Self-Organizing Layer          |                  |                     |
| Data Center / Grid / Cloud |                                |                  |                     |

2 Service Layer: This layer provides a rich set of services to support an autonomic computing.

- Provides Basic Coordination Primitives.
- Query Capabilities: Supports wildcard search to retrieve all tuples.
- Context-Transparent Applications: Application operate without concern for the data's physical location.
- Messaging: Supports publish/subscribe messaging models for communication between nodes.

3 Infrastructure Layer:

The Infrastructure Layers forms the foundation of the CometCloud architecture.

- Content-Based Routing: The routing engine can handle diverse query types.

- Replication and Load Balancing  
Each node keeps a replica of its successor's state to Facilitating Fault tolerance.

Load Balancing is essential for optimizing resource utilization.

- Failure Management: If a node fails, its predecessor merge the replica into its to maintain continuity.

### \* Aneka Cloud Platform:

- => Aneka Cloud Platform is a software platform designed for building and managing distributed applications on the cloud.

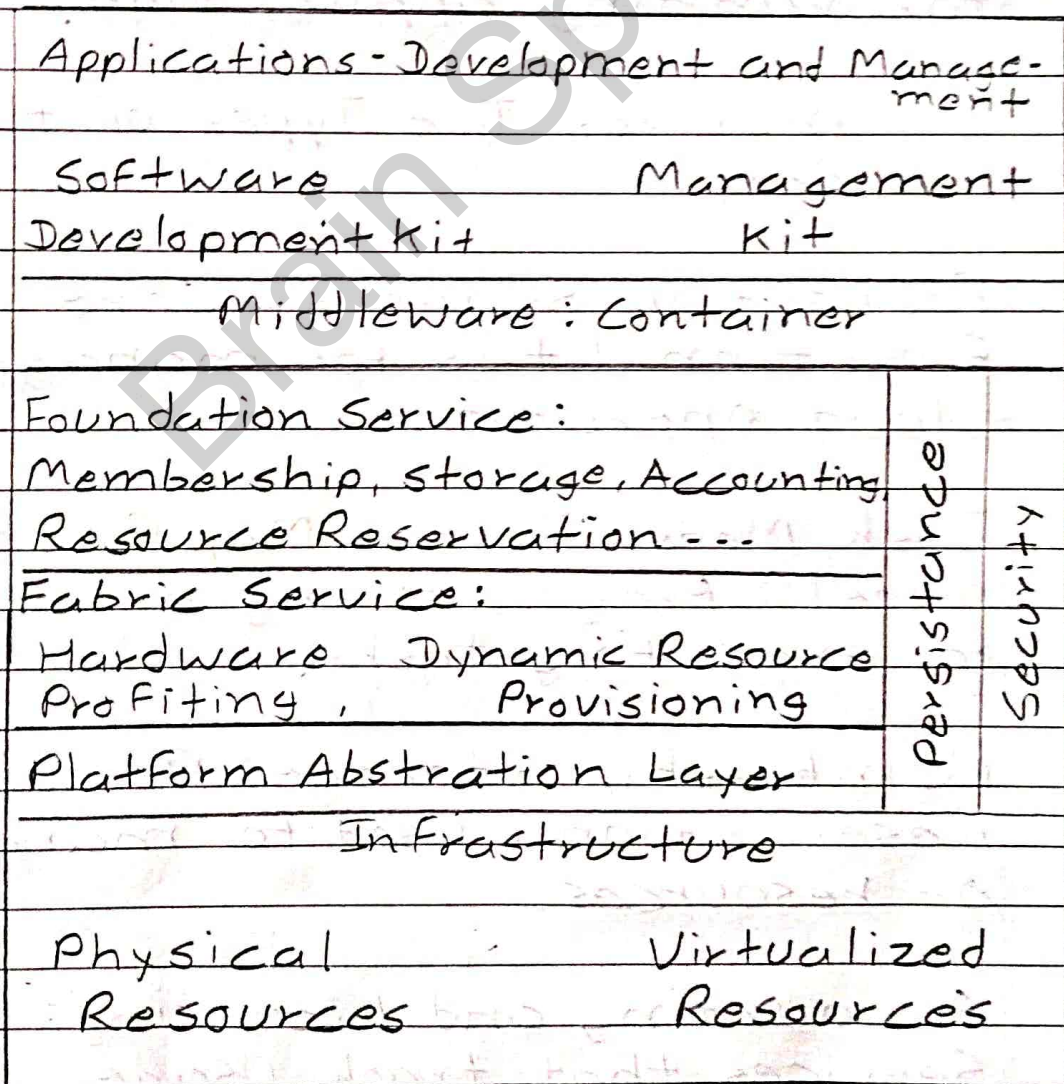
Aneka is versatile and supports public, private or hybrid cloud environments.

There are main Three layer in Aneka Architecture.

# 1 Applications - Development and Management:

This Layer includes tools and components for developing and managing cloud-based application

- Software Development Kit: Provides APIs, tutorials which use to create application for the Aneka Cloud.



- Management Kits: Contains tool like the Management Studio and Admin Portal for cloud administrators to manage applications.

## 2 Middleware - Container:

This Layer represents the Aneka container which is the core middleware environment, for running application.

It includes Two Types of the Services:

(i) Foundation Services: Core functionalities for managing cloud operations.

- Task Model, Thread, MapReduce model: Programming model for executing tasks.
- Membership, storage, Resource Reservation: Used for management of resources.
- Accounting and Licensing: Services that track usage.

### cii) Fabric Services: Low-level Service

- Hardware Profiting: Monitors the hardware available in the system.
- Dynamic Resource Provisioning: Allows for dynamic allocation of resource.

### 3 Infrastructure: This layer represent the physical and virtualized resource

- Private cloud: Internal resources available over a local area network.
- Public cloud: Publicly accessible resources.
- Platform Abstraction Layer: Make the Aneka container compatible with different runtime environment.