

Chapter 5 - Aneka

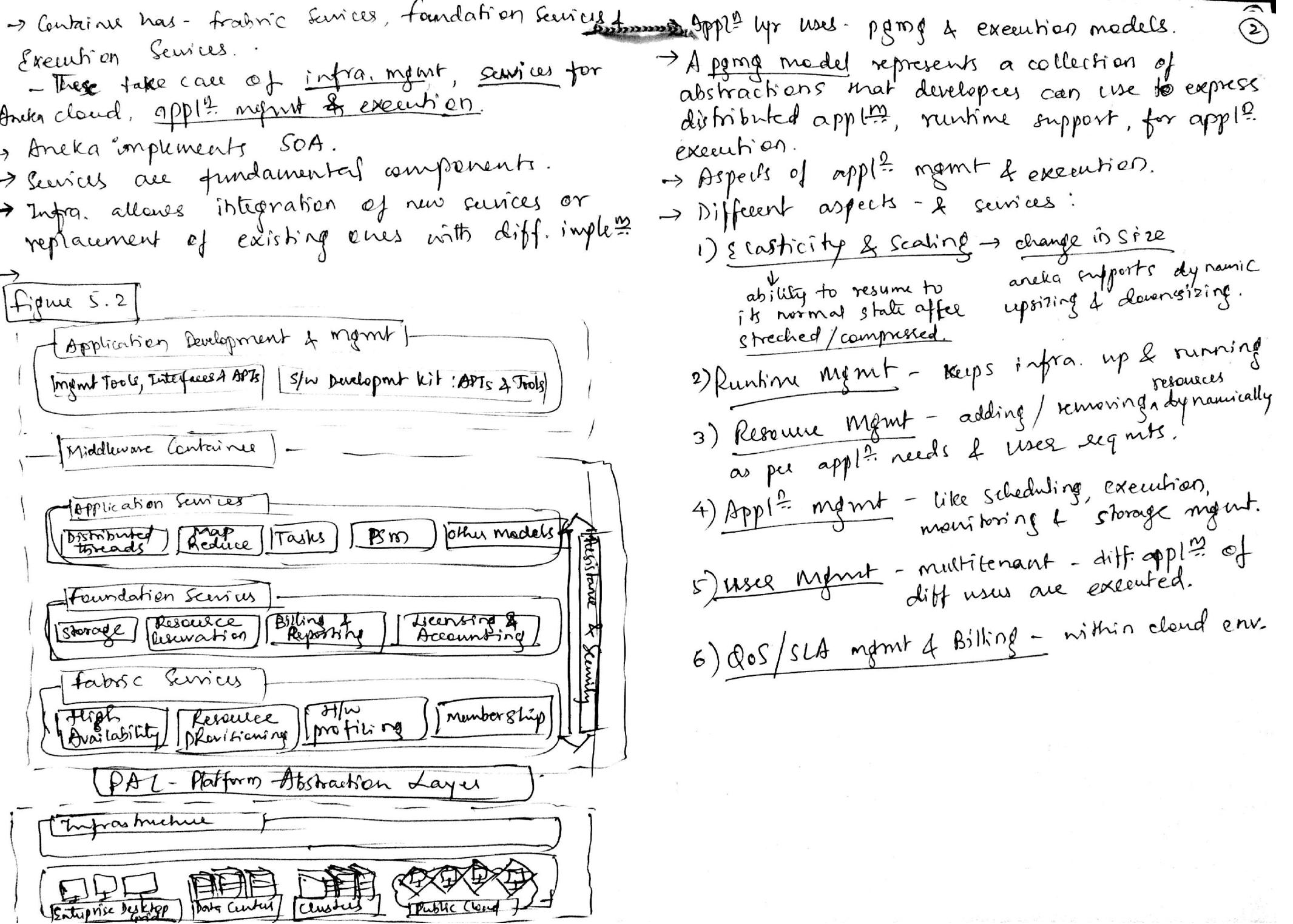
(1)

- 1) Aneka is Manjrasoft Pvt. Ltd's SaaS for developing, deploying & managing cloud applⁿ.
- 2) Consists scalable middleware that can be deployed on top of heterogeneous computing resources.
- 3) Set of APIs with different programming models - such as Task, thread, mapReduce - used for distributed applⁿ.

*★

5.1 framework Overview

- s/w platform for developing cloud computing applications.
- The Aneka Cloud - applⁿ. are executed.
- Aneka is Pure PaaS.
- Deployed on heterogeneous set of resources - s/w, multicore server, datacenters, virtual cloud infra. & mixture of these.
- fig 5.2 : Complete Aneka framework Overview.
 - ★ core infrastructure - frame to be deployed over different platforms & OSs.
 - Aneka container manages the bare metal resources of cloud. infra.
 - Container - is installed on each node basic building block of middleware.
 - Collection of interconnected containers constitute Aneka cloud - a single domain in which services are made available to users & devops.



- ②
- Applⁿ lyr uses - pgmg & execution models.
- A pgmg model represents a collection of abstractions that developers can use to express distributed applⁿ, runtime support, for applⁿ execution.
- Aspects of applⁿ mgmt & execution.
- Different aspects - & services:
- 1) Elasticity & Scaling → change in size
 ↓
 ability to resume to its normal state after stretched/compressed
 anka supports dynamic upsizing & downsizing.
 - 2) Runtime mgmt - keeps infra. up & running
 - 3) Resource mgmt - adding / removing dynamically as per applⁿ. needs & user reqmts.
 - 4) Applⁿ mgmt - like scheduling, execution, monitoring & storage mgmt.
 - 5) User mgmt - multitenant - diff. applⁿ of diff users are executed.
 - 6) QoS / SLA mgmt & Billing - within cloud env.

5.2 : Anatomy of Aneka Container

- Building Blocks of Aneka Clouds.
- Light weight s/w layer designed to host services & interact with underlying OS & H/W.
- 3 major categories
 - 1) Fabric Services → ① Profiling & Monitoring
② Resource mgmt.
 - 2) Foundation → ④ Storage mgmt.
② Accounting, Billing & Resourca Pricing
 - 3) Application → ③ Resource Reservation
① Scheduling
② Execution
- All reside on ^{top of} PAL - Platform Abstraction Layer interface to underlying OS & H/W.

5.2.1 from the ground up : the PAL

- Core infra - is based on .NET technology & allows Aneka container to be portable over different platforms & OSes.
- ECMA 334 & 335 standards for C# language.
- CLI - Common Lang. Interface, defines a common runtime environment & appl². model for executing programs.
- Each OS has different file sys - ; PAL addresses heterogeneity & provides uniform interface for accessing H/W & OS.
- Features
 - ① Platform & platform-independent implementation
 - Interface for accessing hosting platform.
 - ② Uniform access to extended & additional properties of hosting platform.
 - ③ Uniform & platform-independent access to remote-nodes.
 - ④ mgmt interfaces.

- PAL is small layer of software, - that has detection engine, which configures container at boot time.
- ④ based on types of OS - windows, Linux & macOS X,
 - the data collected by PAL :
 - ① no. of cores, freq. & CPU usage
 - ② Memory size & usage
 - ③ Aggregate available disk space
 - ④ N/W addresses & devices attached to node.

5.2.2 fabric Services

- lowest level of Container
- Resource provisioning - virtually allocate resource reqts - nodes on demand.
- Monitoring - allow H/W profiling & implement basic monitoring infra.

(1) Profiling & Monitoring

- exposed through - heartbeat, monitoring & reporting services
- heartbeat service periodically collects dynamic performance information about nodes & publishes to membership service in Aneka Cloud.
- These information is collected by index node, & optimizes the infrastructure.
- Node Resolver - collects data & provides to heartbeat service.

- * → Reporting Service - manages the store for monitored data makes them accessible to other services / external apps for analysis purposes.

- One each node, instance of monitoring service acts as gateway to reporting service & forwards all monitored data that has been collected on the node.

- monitoring provides several built-in services -
- ① Membership Catalogue tracks performance of nodes
 - ② Execution Service monitors several time intervals for execution of jobs.
 - ③ Scheduling Service tracks state transitions of jobs
 - ④ Storage Service monitors data transfer - up down times, file names & sizes
 - ⑤ Resource provisioning Service - tracks provisioning & lifetime info. of virtual nodes.

2 Resource Management

- tasks: Resource membership -
 - reservation
 - provisioning
- services: Index Services (membership catalogue) - Reservation Service Resource provisioning Service
- membership catalogue - tracks & updates nodes info. during container startup.
- Applⁿ. query to discover available services & interact with them.
- M.C. organized as distributed database. to address queries.
- ^{M.C.'s} Collectors of dynamic performance data of each node
- Dynamic Resource provisioning - allows integration & mgmt of virtual resources leased from IaaS - structure of Aneka will be modified. to meet different needs; handling node failures, DOS, constant performance & throughput. of cloud.

- Resource provisioning is based on resource pools ④ - which abstracts interaction with specific IaaS provider by exposing common interface so that all pools are managed uniformly.
- R.P. is designed to support QoS reqmts - driven execution of applⁿ.

5.2.3. Foundation Services

- Infrastructure mgmt features
- logical mgmt of distributed sys built on top of infrastructure & provide supporting services for execution of distributed applⁿ.

1 Storage Mgmt

- file/data transfer mgmt & persistence storage.
- 2 different facilities - centralized file storage - used for executⁿ of compute-intensive applⁿ.
- distributed file s/m storage - for executⁿ of ^{more} data-intensive applⁿ. - large data files - less processing.
- storage Service uses FTP to transfer large files between nodes. & end users.
- file channels - controller + where files are stored handlers - upload, download & browse files.
- for data-intensive applⁿ - GFS (Google file s/m) storing huge files - in terms of chunks of same size. Each chunk - unique ID - identifies files & stored in servers.

- few chunks are replicated to provide high availability & failure tolerance.
- Aneka provides simple (Distributed file sys) DFS - which relies of FS of windows OS.

2] Accounting, Billing & Resource pricing

- tracks status of applⁿ. in cloud.
- info. collected is related to infra usage & applⁿ. execⁿ.
- complete history. is captured.
- Billing - provides detailed info abt each user's usage of resources, ~~with~~ with associated costs.
- Integrated view of budget spent for each applⁿ.
- Accounting Service - keeps track of info. that is related to applⁿ. execⁿ. - jobs distribution, of resources, timing of each job & cost.
- Reporting Service - info. collected from monitoring service for accounting purposes.

3] Resource Reservation - QoS, SLA.

- Reserving resources for exclusive use by specific applⁿ.
- 2 kinds of Services - Resource Reservation & Allocation Service.
 - Resource Reservation & allocates time slots for execⁿ.
 - manages db of each job installed on nodes, & applⁿ. execⁿ.
- Protocols are used for resource reservation process.

- Aneka has APIs, for this service, framework supports 3 different implementations
- a) Basic Reservation - reserves execⁿ slots on nodes, implements alternate offers protocol.
 - b) Libra Reservation - variation of previous, price nodes differently acc to h/w capabilities
 - c) Relay Reservation - very thin implementation, allows resource broker to reserve nodes in clouds & central logic with which these nodes are reserved.
 - Helps in intercloud operations.

→

5.2.4. Application Services

- manages execution of applⁿ.
 - 2 services - scheduling & execution.
- Scheduling -
- planning execⁿ of distributed applⁿ on top of Aneka.
 - allocation of jobs wrt applⁿ.
 - Integrate with foundation & fabric services.
 - Common tasks -
 - 1) job to node mapping
 - 2) Rescheduling of failed jobs
 - 3) Job- status monitoring
 - 4) Applⁿ. status monitoring.
- No centralized services. - these belong to fabric & foundation layers.

- managing the following tasks
 - multiple jobs sent to same node at same time
 - jobs w/o reservations sent to reserved nodes.
 - jobs sent to nodes where reqd. services not installed.
- foundation services provide sufficient info. to avoid such situations.

[2] Execution

- Control execⁿ of single jobs for applⁿ.
- Set up runtime env. hosting execⁿ of jobs.
- Common operations by execution service:
 - unpacking jobs received from scheduler
 - retrieval of i/p files reqd. for job execⁿ.
 - sandboxed execⁿ of jobs
 - submission of o/p files at the end of execⁿ.
 - execⁿ failure mgmt.
 - performance monitoring
 - Packing jobs & sending them back to scheduler.
- Handle less info → integrate with storage, local allocation & monitoring services.
- Application services constitute runtime support of pgmg model: such as
 - Task model - supports independent "bags of tasks"
 - Thread Model - extends classical multithreaded pgmg to a distributed infra & executes remotely.
 - Map Reduce model - proposed by Google
 - Parameter Sweep model - specialized task model.

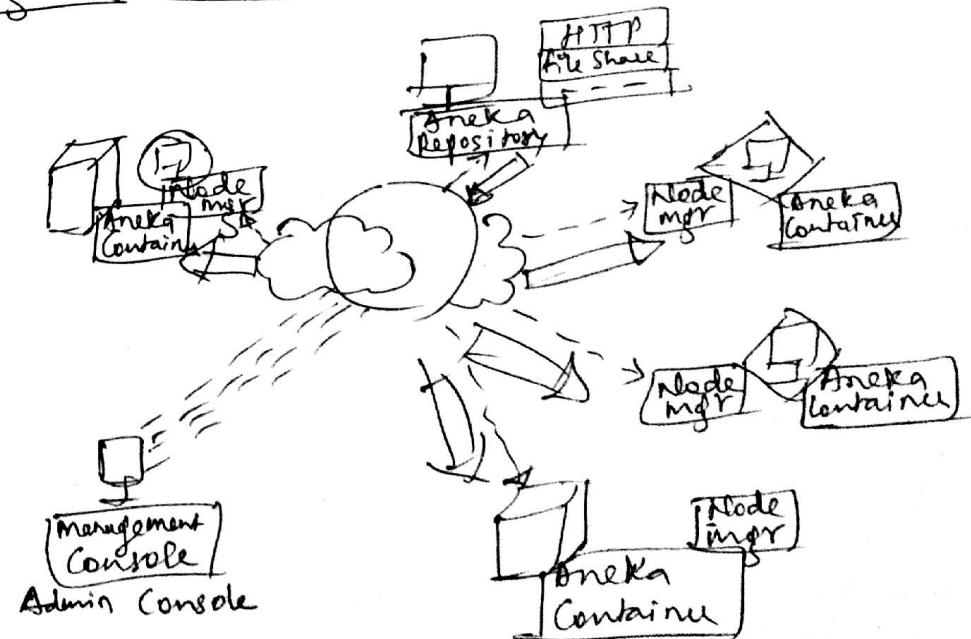
described as template task, whose instances are created by generating different combinations of parameters.

5.3 : Building Aneka Clouds

- platform for developing distributed appl^m for clouds.
- 5.3.1 - Infrastructure Organization
- 5.3.2 - Logical Organization
- 5.3.3 - Private Cloud Deployment mode
- 5.3.4 - Public Cloud Deployment mode
- 5.3.5 - Hybrid cloud "

5.3.1 Infrastructure Organization

fig 5.3 - Aneka Cloud Infrastructure Overview



- main role by Administrative ~~Management~~ console
- Repository - storage for all libraries reqd. to layout & install basic Aneka platform.
- These libraries constitute S/W image for node mgr & container pms.
- Libraries are made available through variety of communication channels - HTTP, FTP, common file sharing, & so on.
- Mgmt console - manages multiple repositories & select one best suits for specific reqmt.
- Aneka node manager installed on nodes - it has Aneka daemon - control & deploy container instances.
- Infra. point of view, mgmt of physical or virtual nodes is performed uniformly as long as it is possible to have Internet.

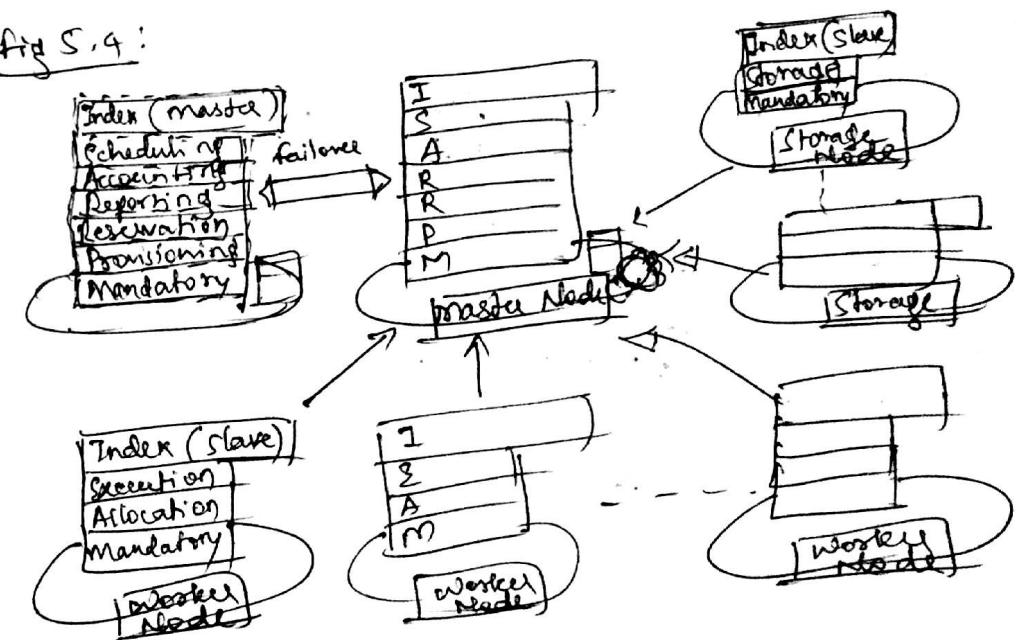
5.3.2 Logical Organization

fig 5.4: logical organization of Aneka cloud

- Most common - master-worker with separate storage configuration.
- Master node features all services to be present in one single copy that provide intelligence.
- presence of Index Service in master node. other services may present in other nodes.
- Master node services
 - Index Service (master copy)

- ⑦
- Logging Service
 - Reservation "
 - Resource provisioning service
 - Accounting Service
 - Reporting & Monitoring "
 - Scheduling Service for supported pfgm models.
 - RDBMS - Relational DBMS.
- worker nodes constitute workforce of Aneka Cloud & configured for exec. of applns.
- features
 - Index Service
 - Heart beat "
 - Logging "
 - Allocation "
 - Monitoring "
 - Execution " for supported pfgm models

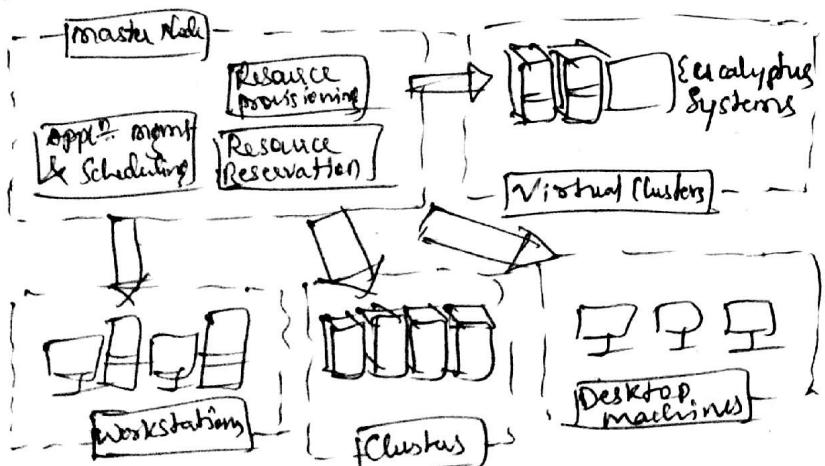
Fig 5.4:



- Storage Nodes are optimized to provide storage support to applⁿ.
- Common configurations
 - a) Index Service
 - b) heartbeat "
 - c) Logging "
 - d) Monitoring "
 - e) Storage "
- One storage node for less reqmt.
- Very small deployments - storage node is master node.

5.3.3. Private Cloud Deployment mode

fig 5.5 : Private Cloud Deployment

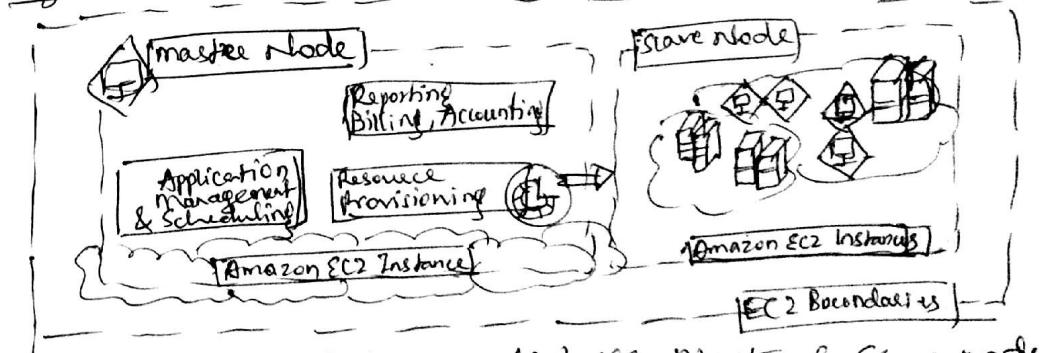


- Constitutes local physical resources & infrastructure mgmt s/w providing access to local pool of nodes
- Nodes are virtualized.
- Aneka cloud is created by heterogeneous pool of resources - desktop m/cs, clusters or workstations.

- Aneka cloud leverages these resources based on application needs.
- Resource provisioning Service is performed by s/w such as XenServer, Eucalyptus & OpenStack.
- In private clouds, workload is predictable & VM can address excess demand.
- Nodes have static configuration, need not reconfigured.
- Resource mgmt is accomplished by Reservation Service.
- Workstation clusters have specific legacy s/w req for execution of applⁿ.
- Desktop m/cs used during day for office automation.

5.3.4. Public Cloud Deployment mode

fig 5.6 : Public Aneka Cloud Deployment

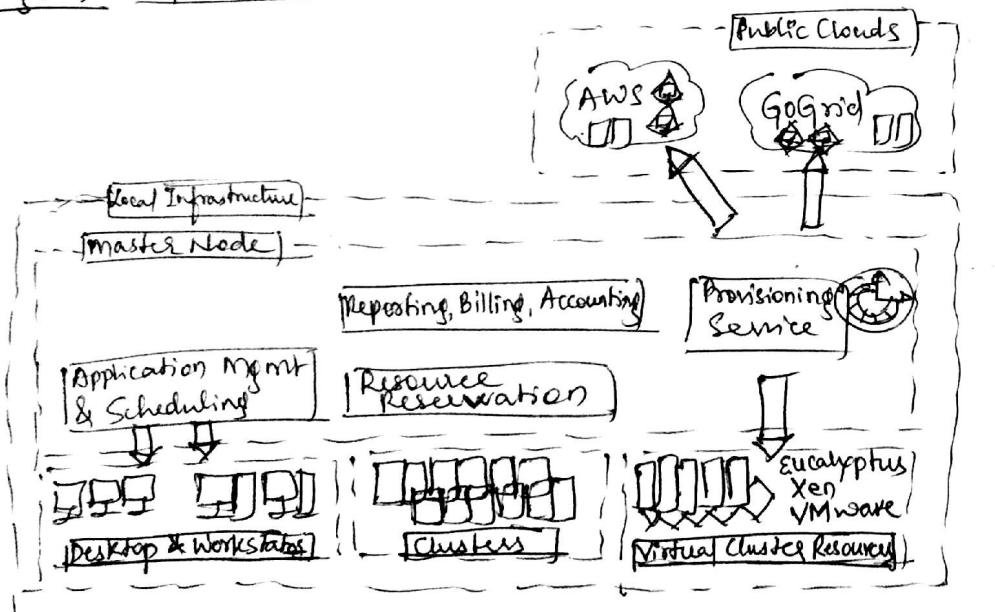


- Public cloud of Aneka deploys Master & Slave nodes of virtualized infrastructure of resource providers such as Amazon EC2 or Google.
- Static Deployment of nodes are provisioned & used as though they were real machines.
- Elastic IaaS Service is provided - Cloud is Completely Dynamic

- Deployment is within single IaaS provider.
- to minimize data transfer between different IaaS providers.
- fundamental role is played by Resource Provisioning Service - configured with different images & templates to start service.
- Master Node has another imp. service - Accounting & Reporting - details of resource utilization by users & applications.
- Multitenant Cloud. - users are billed according to consumption
- Dynamic instances are configured on worker nodes.
- Application execution, elastic scaling are performed by dynamic instances & provisioning.

5.3.5. Hybrid Cloud deployment mode

fig 5.7 : Hybrid Cloud Deployment



- (9)
- The most common deployment of Aneka.
 - Aneka hybrid cloud leverages following capabilities :
 - 1) Dynamic Resource Provisioning
 - 2) Resource Reservation
 - 3) Workload Partitioning
 - 4) Accounting, Monitoring & Reporting.
 - Minimize expenditure for applⁿ execution by obtaining local resources & virtually configured.
 - Majority of applⁿ are executed on workstations & clusters, - constantly connected to Aneka cloud.
 - Any additional computing capabilities can be addressed by local virtualization facilities.
 - If more computation is req'd., external IaaS providers leverage resources.

5.4. Cloud Programming & Management

- Aneka's primary purpose is to provide a scalable middleware product in which to execute distributed applications.
- Aneka provides developers with a comprehensive & extensible set of APIs & Admins with powerful & efficient mgmt. tools.

5.4.1. Aneka SDK — Application Model

- Service

5.4.2. Management Tools — Infrastructure mgmt

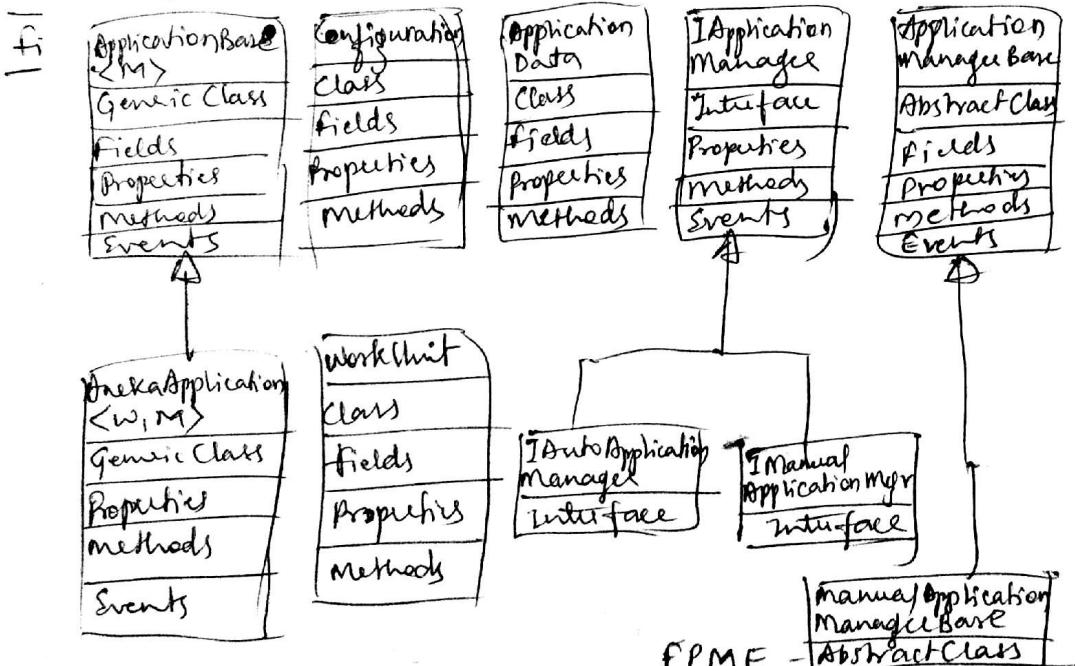
- = Platform
- = Application

5.4.1. Aneka SDK

- provide APIs for developing applications.
- SDK supports pgmg models & services by means of Application Model & Service Model.

① Application Model

- supports distributed execution in cloud using Pgmg models.
 - represents minimum set of APIs that is common to all Pgmg models,
 - model is further specialized according to needs & particular features of each of the pgmg models.
5. → Figure 5.8: Aneka Application Model



(10)

→ Each distributed Application running on top of Application Model represents instance of a ApplicationBase<M>, M = Specific type of Application model.

- Application class - constitute developer's view.
- Application managers - internal components that interact with Aneka clouds to monitor & control the execution of application.
- Aneka further specializes applications into 2 main categories:
 - applications whose tasks are generated by user.
 - " " " " runtime infrastructure.

- (1) is most common & used as reference.
- Task model, Thread Model & Parameter Sweep Model.
- Applications that fall into this category are composed of collection of units of work submitted by user - Workunit class.
- Inherit / Instances of AnekaApplication<W,M>
 - w = Workunit class
 - m = Application manager
- Used to implement IManualApplicationManager Interface.
- (2) covers MapReduce - work is generated by runtime infrastructure.
- No Workunit class used, specific classes used by application developer depend on pgmg model used.

~~Ex: mapreduce pgm model - developer express their appl. in a form - map & reduce.~~

- hence MapReduceApplication class provides interface for specifying Mapper<K,V> & Reducer<K,V>.

→ ∴ ApplicationBase<M> is used as class & M implements IAutoApplicationManager.

→ Other classes - Configuration Class - settings, required to initialize application & customize its behaviour.

→ Application Data Class - runtime information of application.

→ Table 5.1 : ~~Anka's~~ Anka's Application Model Features

Category	Description	Basic Application Type	Work Units?	Pgmg models
Manual	Units of work are generated by user & submitted through Application	ManualApplication<W,M> IManualApplicationManager<W> ManualApplicationManager<W>	Yes	Task Model Thread n Parameter Sweep model
Auto	Units of work are generated by run-time infrastructure & managed internally	ApplicationBase<M> IAutoApplicationManager	No	Map Reduce model

② Service Model

(1)

→ Anka Service model defines basic requirements to implement a service.

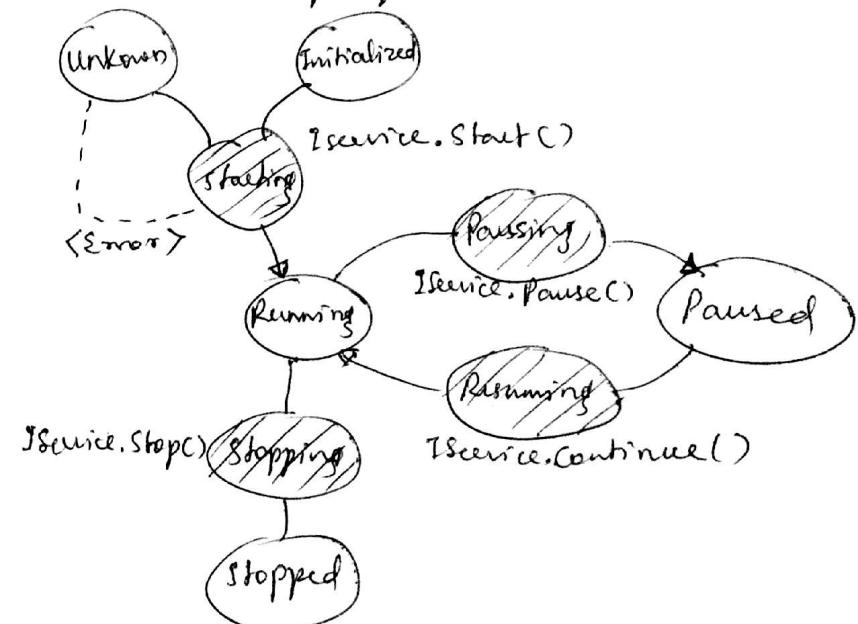
→ Container defines runtime environment in which services are hosted. - IService interface has following mtds & properties:

(a) Name & Status

(b) Control operations - Start, Stop, Pause & Continue methods

(c) Message Handling by means of HandleMessage mtd.

→ figure 5.9 : Service Life Cycle.



→ Shaded balloons - transient states
White - - - Steady states.

→ Service instance can initially be in Unknown/Initialized state, - creation of service by invoking constructor.

→ If we invoke start() mtd, service will be in starting state.

- later exhibit Running state - service will last as long as container is active & running.
- Service will process msg's.
- If Exception occurs, service falls back to Unknown state - signalling error.
- Pause / Resume Service while in Running state.
- Pause method & Continue method.
- When container shuts down, Stop method is called on each service. Services first move in stopping state & finally stopped state.
- All allocated resources will be released.
- Aneka uses strongly typed message-passing communication model.
 - each service defines its own messages.
 - new services - define type of messages used for communication.
- Each message type inherits from base class Message - defining common properties:
 - (a) Source node & Target node
 - (b) Source Service & " " Service
 - (c) Security Credentials.

3.4.2. Management Tools

- Aneka is pure Paas implementation & requires virtual or physical hardware to be deployed.
- Hence infrastructure mgmt + installing logical clouds on such infrastructure is fundamental feature of Aneka's mgmt layer.

(1) Infrastructure mgmt

- (1) Platform - " -
- (2) Application - " -

(1) Infrastructure mgmt -

- Virtual & physical h/w is deployed.
- Virtual h/w is managed by Resource Provisioning Service. - acquires resources on demand.
- Physical h/w is directly managed by Admin console - using Aneka mgmt API of PaaS.

(2) Platform Mgmt

- Collection of connected containers defined platform on top of which applications are executed.
- concerned with logical organization & structure of Aneka Clouds
- Partition available h/w into several clouds
- Variably configured for different purposes.
- Operations - cloud monitoring, resource provisioning & reservation, user mgmt & application profiling.

③ Application Management

- Identify user contribution to cloud.
- Mgmt APIs provide administrators with monitoring & profiling features & help them track usage of resources & relate them to users & applications.