FUTURE VISION BIE

One Stop for All Study Materials

& Lab Programs

Future Vision

By K B Hemanth Raj

Scan the QR Code to Visit the Web Page

Or

Visit : https://hemanthrajhemu.github.io

Gain Access to All Study Materials according to VTU,
CSE – Computer Science Engineering,
ISE – Information Science Engineering,
ECE - Electronics and Communication Engineering
& MORE...

Join Telegram to get Instant Updates: https://bit.ly/VTU_TELEGRAM

Contact: MAIL: futurevisionbie@gmail.com

INSTAGRAM: www.instagram.com/hemanthraj_hemu/

INSTAGRAM: www.instagram.com/futurevisionbie/

WHATSAPP SHARE: https://bit.ly/FVBIESHARE

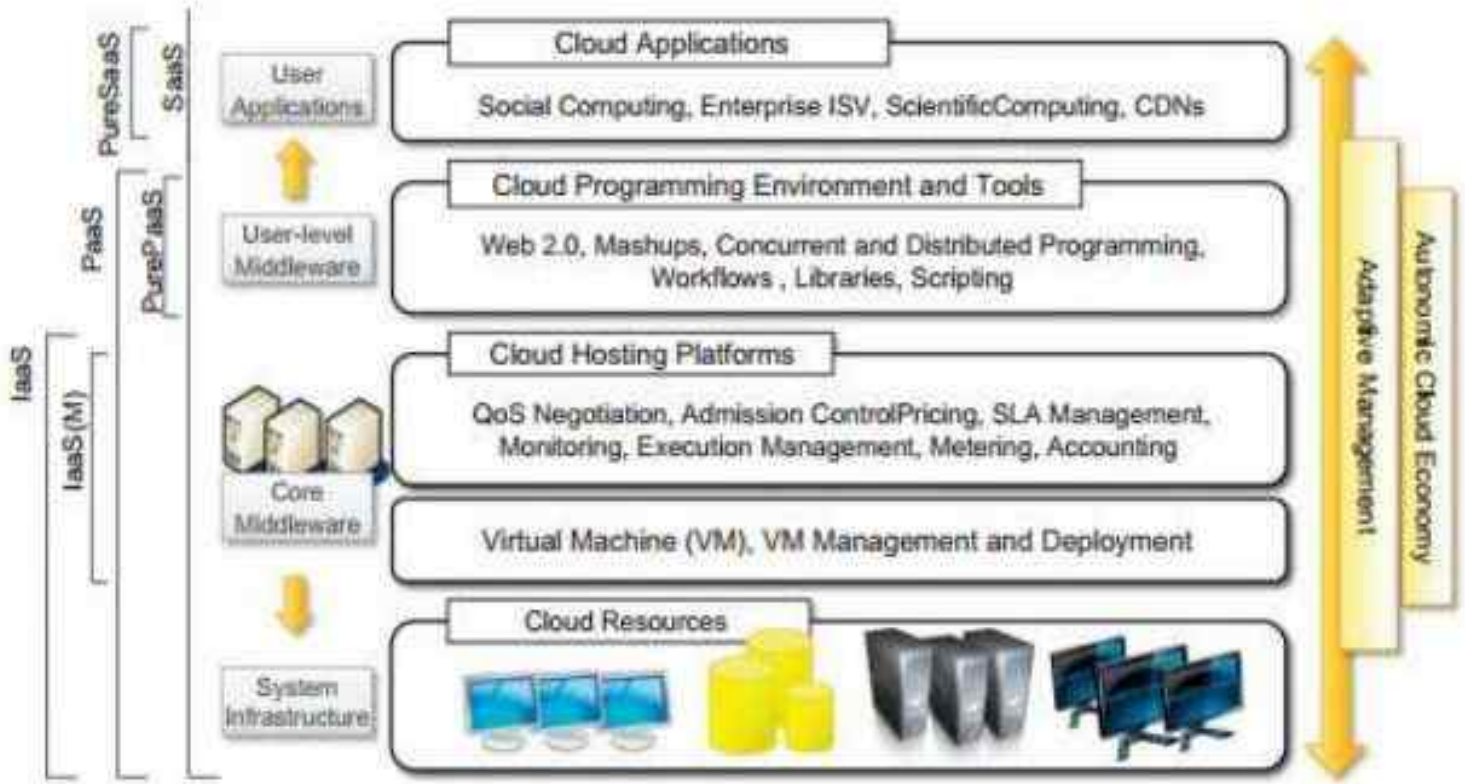
Module 2

Chapter 4: Cloud Computing Architecture
Chapter 5: Aneka Cloud Application Platform

By
Dr.Sunanda Dixit
Associate Professor
CSE Department, BMSIT&M
Bangalore

https://hemanthrajhemu.github.io

The Cloud Reference Model



https://hemanthrajhemu.github.io

Types of Services

- Infrastructure or hardware-as-a-service
- Platform as a service
- Software as a service

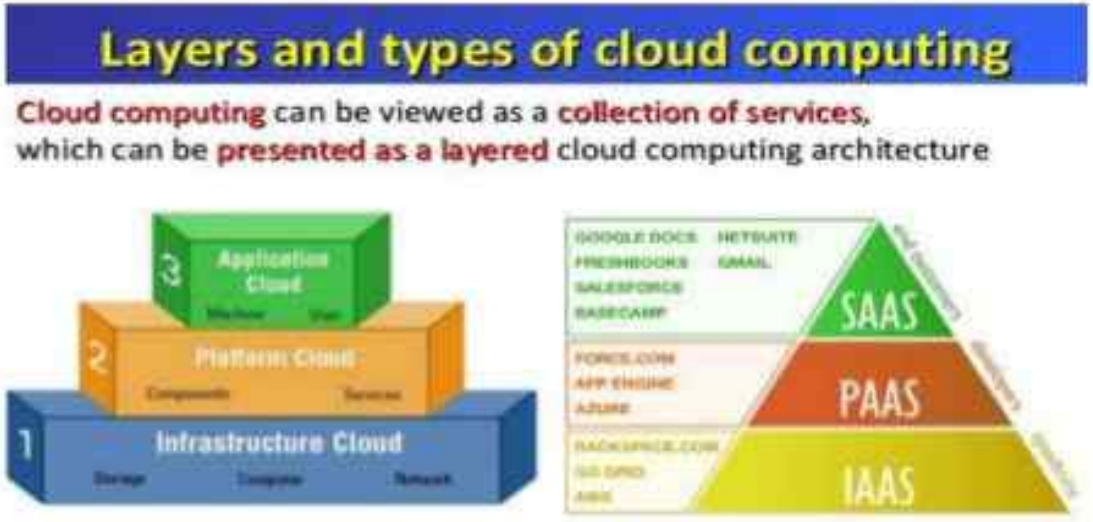


Table 4.1 Cloud Computing Services Classification

Category	Characteristics	Product Type	Vendors and Products
SaaS	Customers are provided with applications that are accessible anytime and from anywhere.	Web applications and services (Web 2.0)	SalesForce.com (CRM) Clarizen.com (project management) Google Apps
PaaS	Customers are provided with a platform for developing applications hosted in the cloud.	Programming APIs and frameworks Deployment systems	Google AppEngine Microsoft Azure Manjrasoft Aneka Data Synapse
IaaS/HaaS	Customers are provided with virtualized hardware and storage on top of which they can build their infrastructure.	Virtual machine management infrastructure Storage management Network management	Amazon EC2 and S3 GoGrid Nirvanix

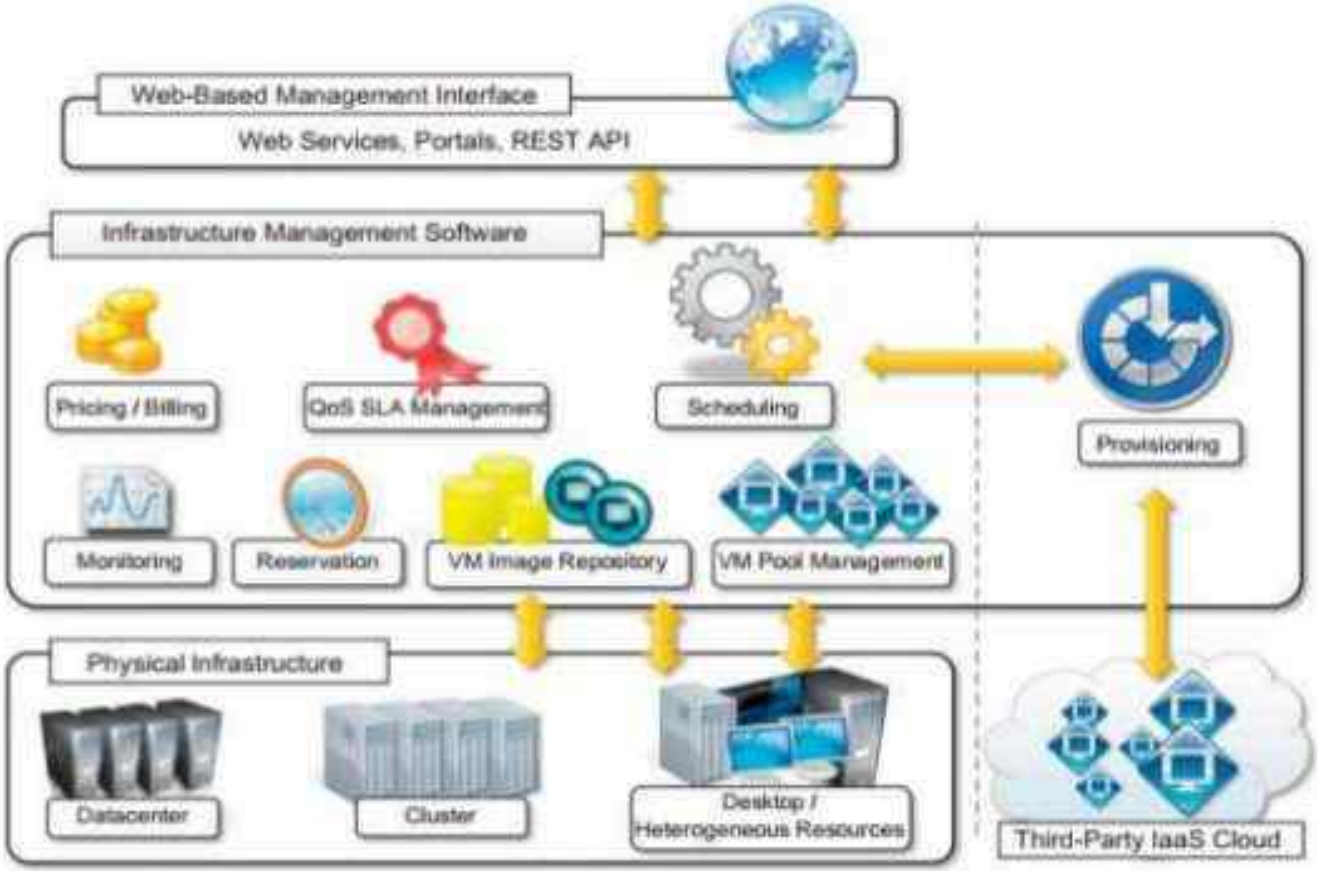
https://hemanthrajhemu.github.io

Infrastructure(IaaS) or hardware-as-a-service (HaaS)

- Most popular and developed market segment of cloud computing
- Customizable infrastructure on demand
- On demand computing resources
- No up-front commitment
  - Start small and grow as required
  - No contract, Only credit card!
- Pay for what you use
- No maintenance
- Measured service
- Scalability
- Reliability

https://hemanthrajhemu.github.io

Infrastructure-as-a-Service reference implementation

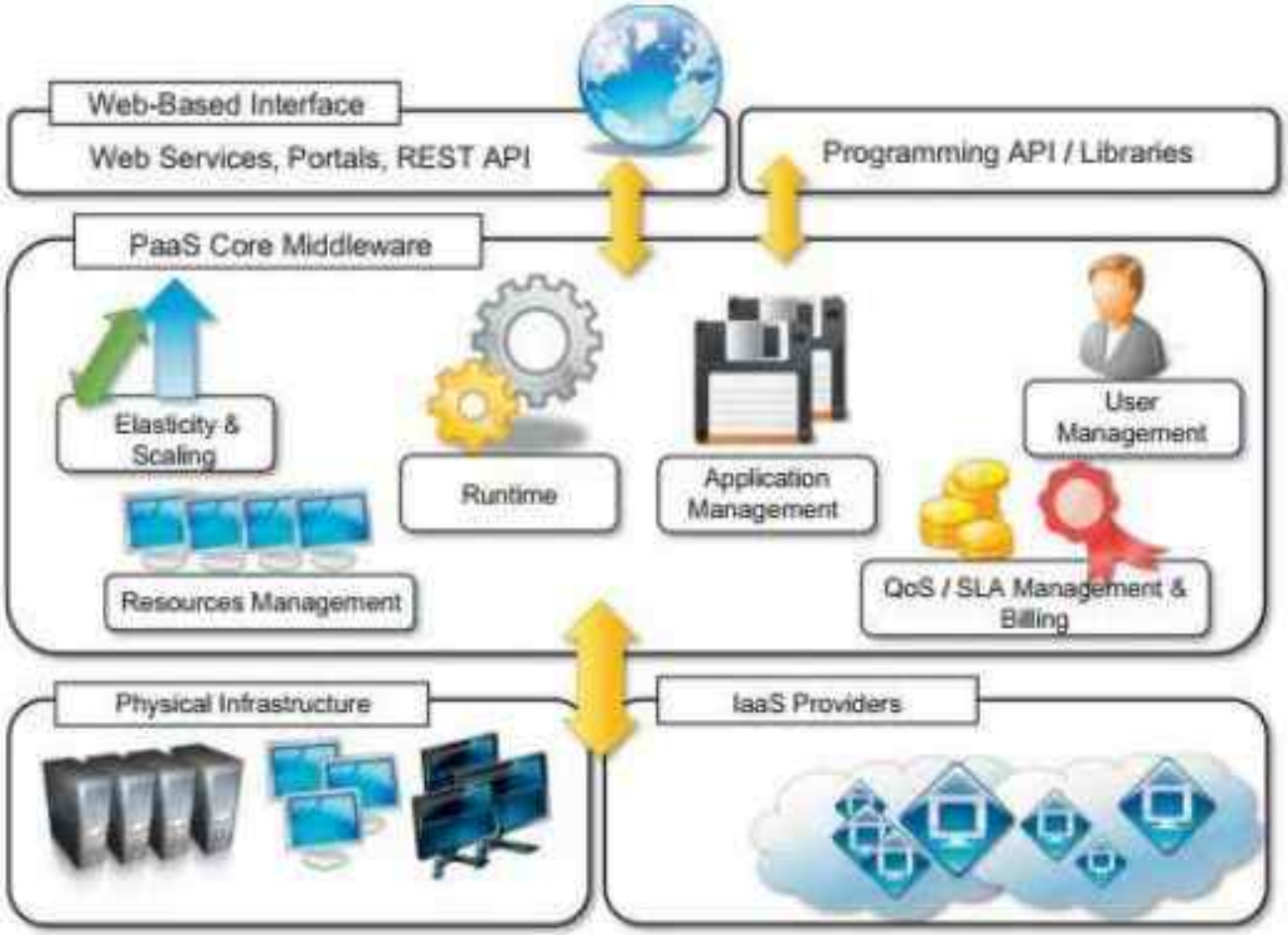


https://hemanthrajhemu.github.io

Example

- https://cloud.google.com/compute/
- https://www.google.co.in/about/datacenters/inside/streetview/

Platform as a service



https://hemanthrajhemu.github.io

Table 4.2 Platform-as-a-Service Offering Classification

Category	Description	Product Type	Vendors and Products
PaaS-I	Runtime environment with Web-hosted application development platform. Rapid application prototyping.	Middleware + Infrastructure Middleware + Infrastructure	Force.com Longjump
PaaS-II	Runtime environment for scaling Web applications. The runtime could be enhanced by additional components that provide scaling capabilities.	Middleware + Infrastructure Middleware Middleware + Infrastructure Middleware + Infrastructure Middleware + Infrastructure Middleware	Google AppEngine AppScale Heroku Engine Yard Joyent Smart Platform GigaSpaces XAP
PaaS-III	Middleware and programming model for developing distributed applications in the cloud.	Middleware + Infrastructure Middleware Middleware Middleware Middleware Middleware	Microsoft Azure DataSynapse Cloud IQ Manjrasof Aneka Apprenda SaaSGrid GigaSpaces DataGrid

https://hemanthrajhemu.github.io

PaaS Characteristics

- Runtime framework
- Abstraction
- Automation
- Cloud services

https://hemanthrajhemu.github.io

Example

- https://cloud.google.com/appengine/

Intro to SaaS - Definition

- **Software as a service** (**SaaS**) is a model of software delivery where the software company provides maintenance, daily technical operation, and support for the software provided to their client.
- It assumes the software is delivered over the internet.
- Software delivered to home consumers, small business, medium and large business

https://hemanthrajhemu.github.io

Software Trends

Business Trend

- Smaller package with flexible coupled component
- Cost effective
- Easily deployed, reconfigured and updated
- Standardized industrial application

Traditional enterprise Software

- Delivered as a large package
- Charge high license fee
- High deployment and maintenance fee

https://hemanthrajhemu.github.io

Characteristics of SaaS

- The product sold to customer is application access.
- The application is centrally managed.
- The service delivered is one-to-many.
- The service delivered is an integrated solution delivered on the contract, which means provided as promised.

https://hemanthrajhemu.github.io

Examples

https://hemanthrajhemu.github.io

Types of clouds

- Public clouds.
- Private clouds
- Hybrid or heterogeneous clouds
- Community clouds

https://hemanthrajhemu.github.io

Public Cloud

- The cloud is open to the wider public
- Anyone, from anywhere, and at any time
- one or more datacenters connected together
- Any customer can easily sign in with the cloud provider
- minimizing IT infrastructure costs
- handling peak loads on the local infrastructure
- Multitenancy
- QoS

https://hemanthrajhemu.github.io

Public Cloud

- A public cloud can offer any kind of service: infrastructure, platform, or applications.
- scale on demand and sustain peak loads.

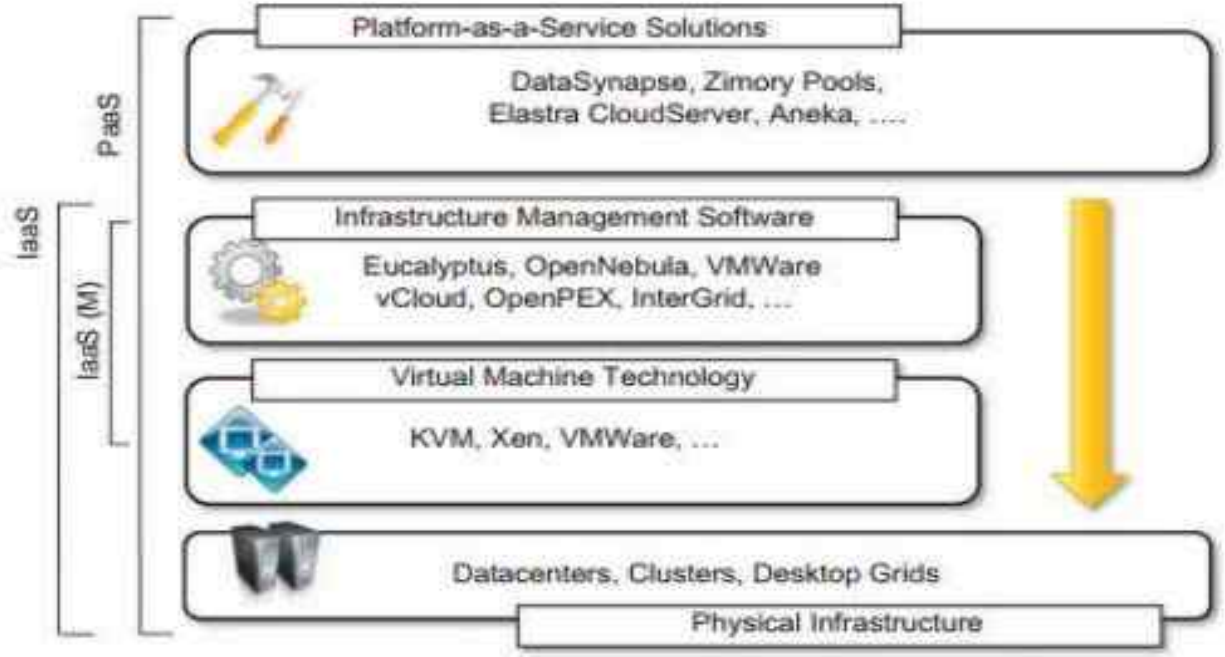
https://hemanthrajhemu.github.io

Private clouds

- The cloud is implemented within the private premises
- Public clouds are cut IT costs and reduce capital expenses, but they are not applicable in all scenarios
- loss of control
- control of the infrastructure
- core logic and sensitive data
- government and military agencies will not consider public clouds as an option for processing or storing their sensitive data

https://hemanthrajhemu.github.io

Private clouds hardware and software stack.



https://hemanthrajhemu.github.io

Advantage of Private Cloud

- Customer information protection
- protection Infrastructure ensuring SLAs
- Compliance with standard procedures and operations

https://hemanthrajhemu.github.io

Hybrid or heterogeneous clouds

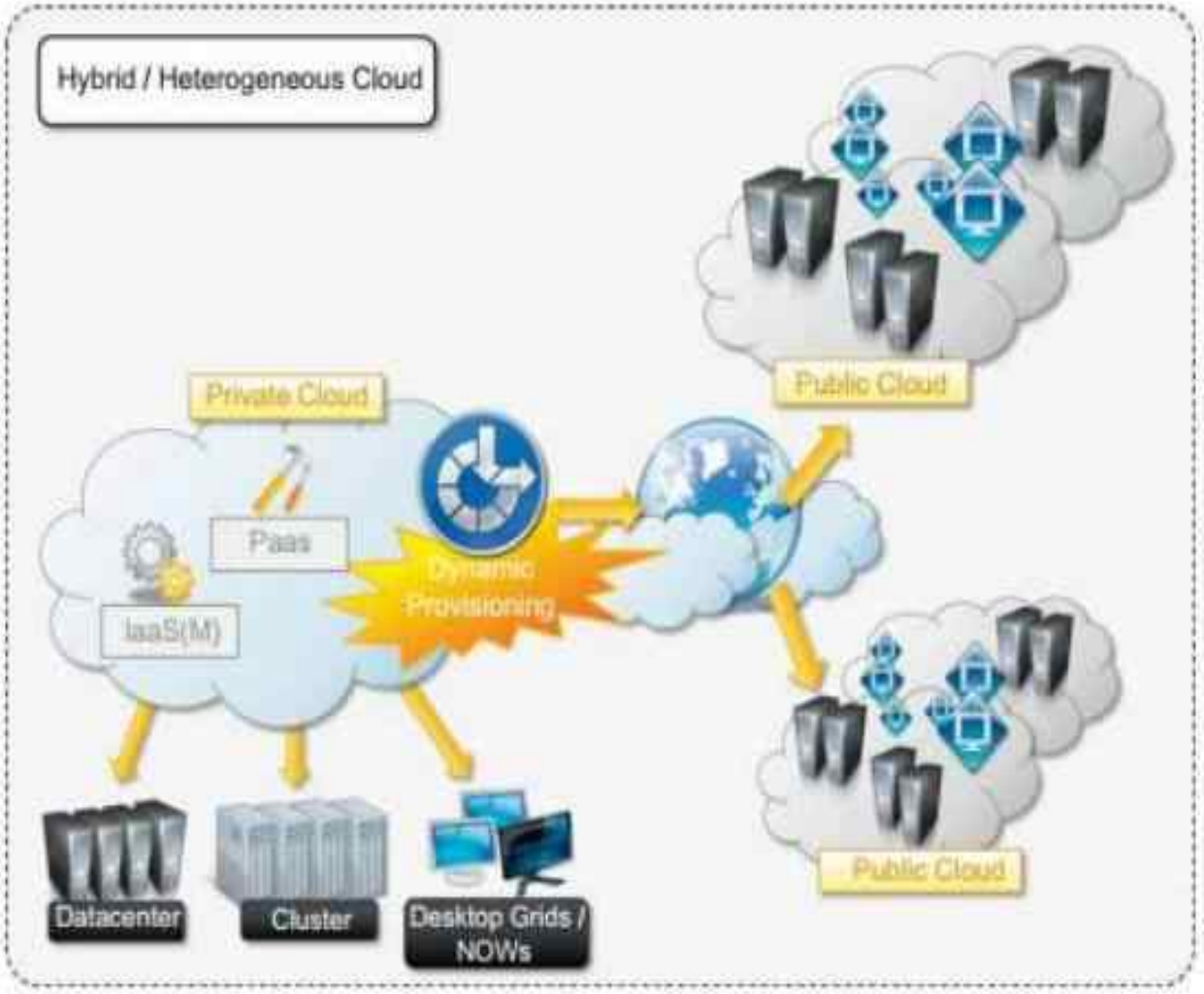
- The cloud is a combination of the two previous solutions
- IT infrastructure + Security
- Private clouds are the perfect solution when it is necessary to keep the processing of information within an enterprise's
- Drawback – Scalability
- Public Cloud Helps to Scaling Infrastructure

https://hemanthrajhemu.github.io

Hybrid or heterogeneous clouds

- maintain sensitive information within the premises,
- Service-oriented computing already introduces the concept of integration of paid software services with existing application deployed in the private premises.
- OpenNebula

https://hemanthrajhemu.github.io



https://hemanthrajhemu.github.io

A community cloud.

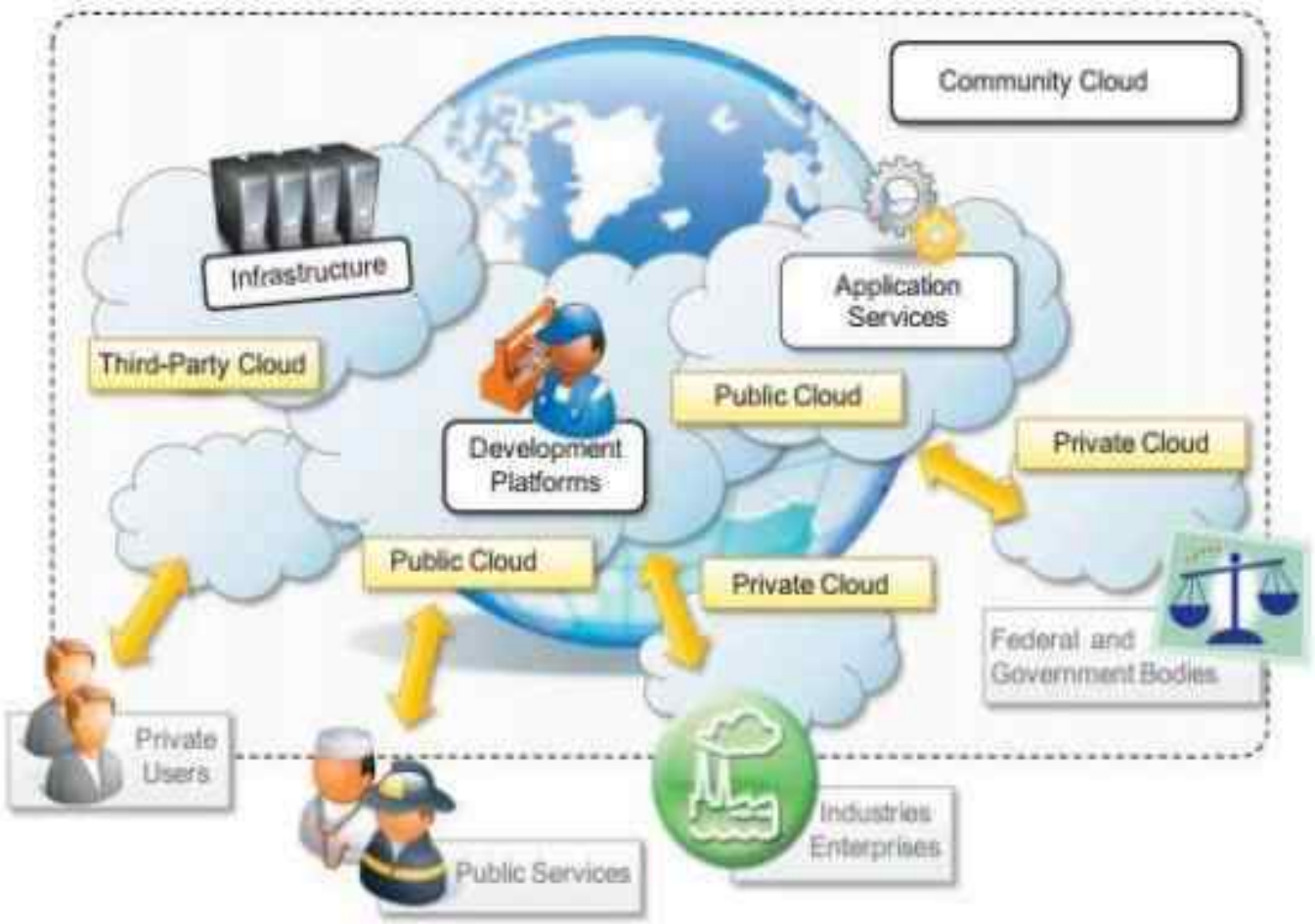
- Community clouds are distributed systems created by integrating the services of different clouds to address the specific needs of an industry

https://hemanthrajhemu.github.io

Candidate sectors for community clouds are

- Media industry
- Healthcare industry
- Energy and other core industries.
- Public sector
- Scientific research.

https://hemanthrajhemu.github.io



https://hemanthrajhemu.github.io

Benefits of Community Cloud

- Openness
- Community
- Graceful failures
- Convenience and control
- Environmental sustainability

https://hemanthrajhemu.github.io

Economics of the cloud

- Reducing the capital costs associated to the IT infrastructure
- Eliminating the depreciation or lifetime costs associated with IT capital assets
- Replacing software licensing with subscriptions
- Cutting the maintenance and administrative costs of IT resources

https://hemanthrajhemu.github.io

Open challenges

- Cloud definition

characterizes cloud computing : on-demand self-service, broad network access, resource-pooling, rapid elasticity, and measured service

- Cloud interoperability and standards
- Scalability and fault tolerance
- Security, trust, and privacy
- Organizational aspects

https://hemanthrajhemu.github.io

Chapter 5: Aneka
Cloud Application Platform

https://hemanthrajhemu.github.io

Chapter 5: Aneka
Cloud Application Platform

By
Dr.Sunanda Dixit
Associate Professor
CSE department, BMSIT&M
https://hemanthrajhemu.github.io

Platform as a Service (PaaS) Providers

http://cloudcomputingwire.com

https://hemanthrajhemu.github.io

Cloud middleware

Open Source Private Cloud Middleware

- Joyent /Reasonably Smart
- Eucalyptus
- Globus Nimbus
- Reservoir
- OpenNebula

Many of the grid & cloud products only

Support Open Source– what can we use on

Microsoft based systems ??

https://hemanthrajhemu.github.io

https://hemanthrajhemu.github.io

Aneka

– Aneka is a middleware for developing Cloud Computing flexible and scalable applications

https://hemanthrajhemu.github.io

What is Aneka?

- Platform for deploying Clouds developing applications.
- Service Oriented Architecture (SOA).
- Provides a runtime environment and set of APIs.
- Choice for flexible, extensible .NET enterprise Cloud application and deployment.

https://hemanthrajhemu.github.io

Who provides an Aneka?

- ***Manjrasoft***
  - *Named as manjrasoft due to the river called Manjira*
  - *CloudSim – Simulation Sofware*
  - *Aneka is the first Product.*
- ***Dr. Rajkumar Buyya***

  CEO – Manjrasoft Pty Ltd
  Director, GRIDS Laboratory, University of Melbourne, Australia

https://hemanthrajhemu.github.io

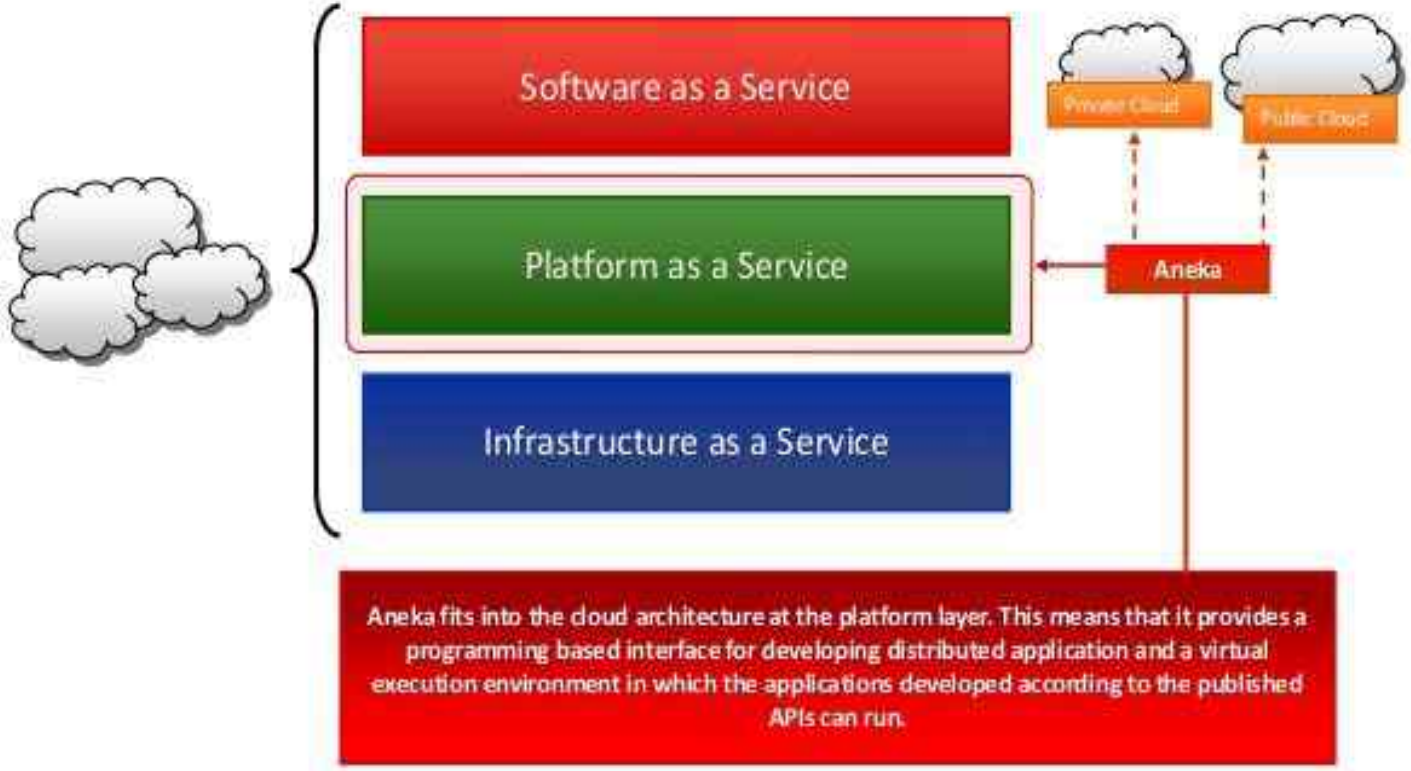
Why Aneka?

- Aneka (Sanskrit): many, in many ways, many in one…
- This means:
  - Multiple programming models
  - Multiple Application Scenario
  - Multiple deployment models
  - Multiple platform and OSs

https://hemanthrajhemu.github.io

Why Aneka?

- Aneka & Clouds



https://hemanthrajhemu.github.io

What Aneka does?

- *I need to harness a huge computational power for my application….*

Aneka provides a runtime environment that manages a collective of physical and virtual machines, which are coordinated transparently to applications for large computations

- *I want to develop distributed applications easily…*

Aneka provides an easy interface for developing distributed applications by using common and intuitive abstractions in distributed computing

- *I have a limited budget to execute my applications….*

Aneka provides algorithms and strategies allowing users to complete their application in time and within their possible expenditure

- *I don't care about operating systems, platforms, and hardware…*

Aneka provides a portable runtime environment that can be deployed on different operating systems and hardware (desktop PCs, clusters, datacenters, Clouds…)

https://hemanthrajhemu.github.io

Features

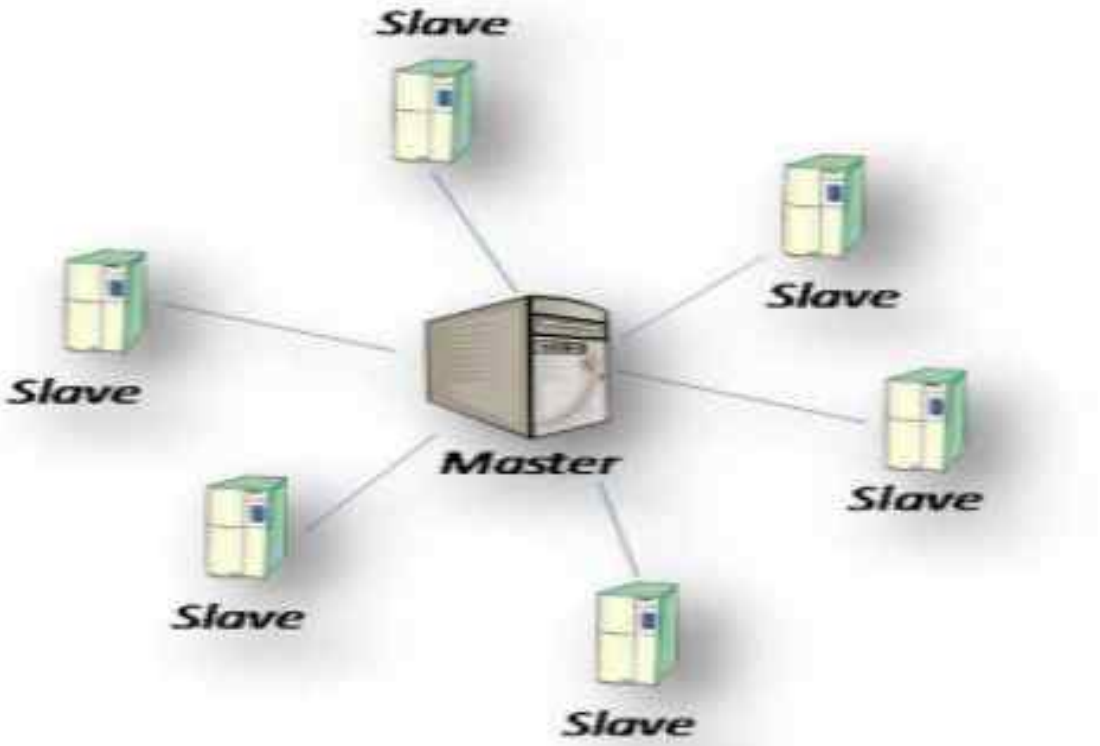
- Current Applications
  - Scientific
    - Distributed evolutionary computation
    - Proteine structure prediction
  - Commercial
    - Engineering: Go Front (China): Train models rendering
    - Media and games: platform for on-line gaming
    - Financial: risk analysis
    - Office automation: Excel integration
  - Educational
    - Image filtering
    - Image rendering
    - Distributed systems teaching

https://hemanthrajhemu.github.io

Private Cloud Setup Using Aneka

https://hemanthrajhemu.github.io

Aneka Network Configuration



https://hemanthrajhemu.github.io

System Requirements

- Hardware Requirements:
  - 1G RAM, 40 GB disk space
- Software Requirements:
  - Microsoft Windows XP Professional Edition SP 2
  - Microsoft .NET framework 2.0+
  - Microsoft SQL Server 9.0.x / SQL Server Express 9.0.x / MySQL Server 5.1.3.0 (Optional, if database support is required)
  - IIS 5.0+ (optional, if the role based security web service is required)

https://hemanthrajhemu.github.io

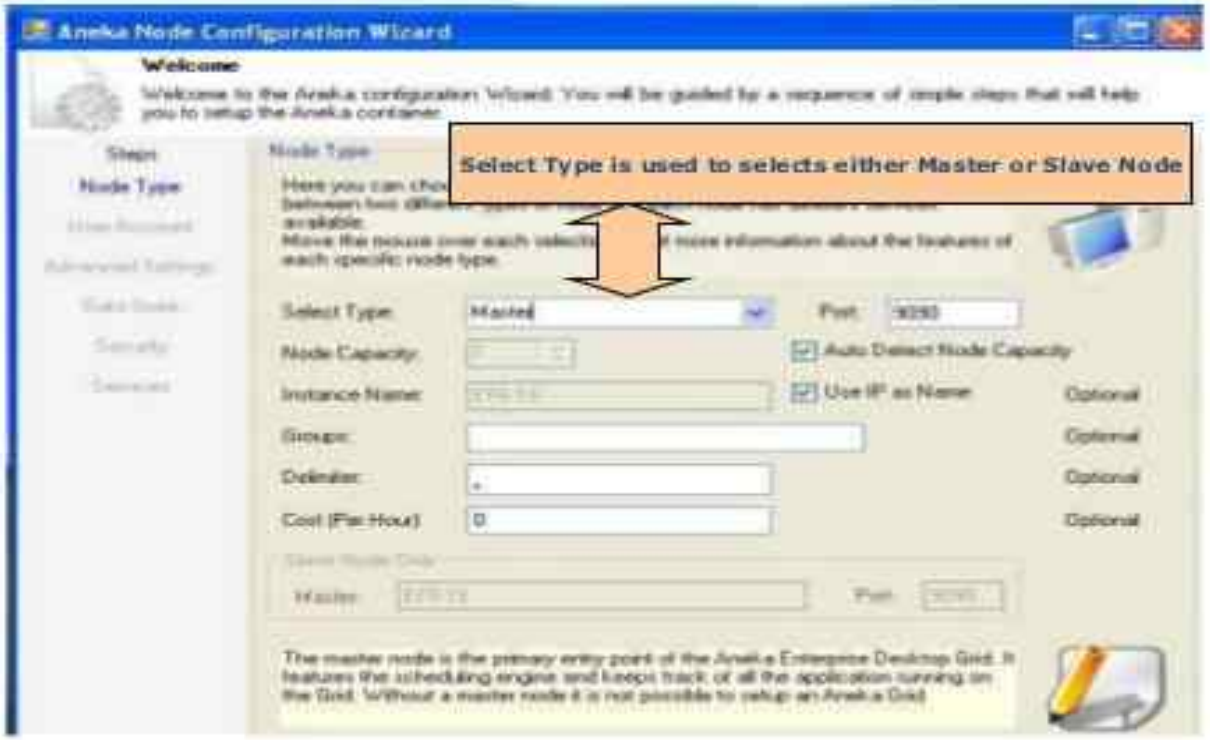
Installation of Master Node

Installation of Mater Node

- Specify the Name of the Cloud
- Used for future node references
-

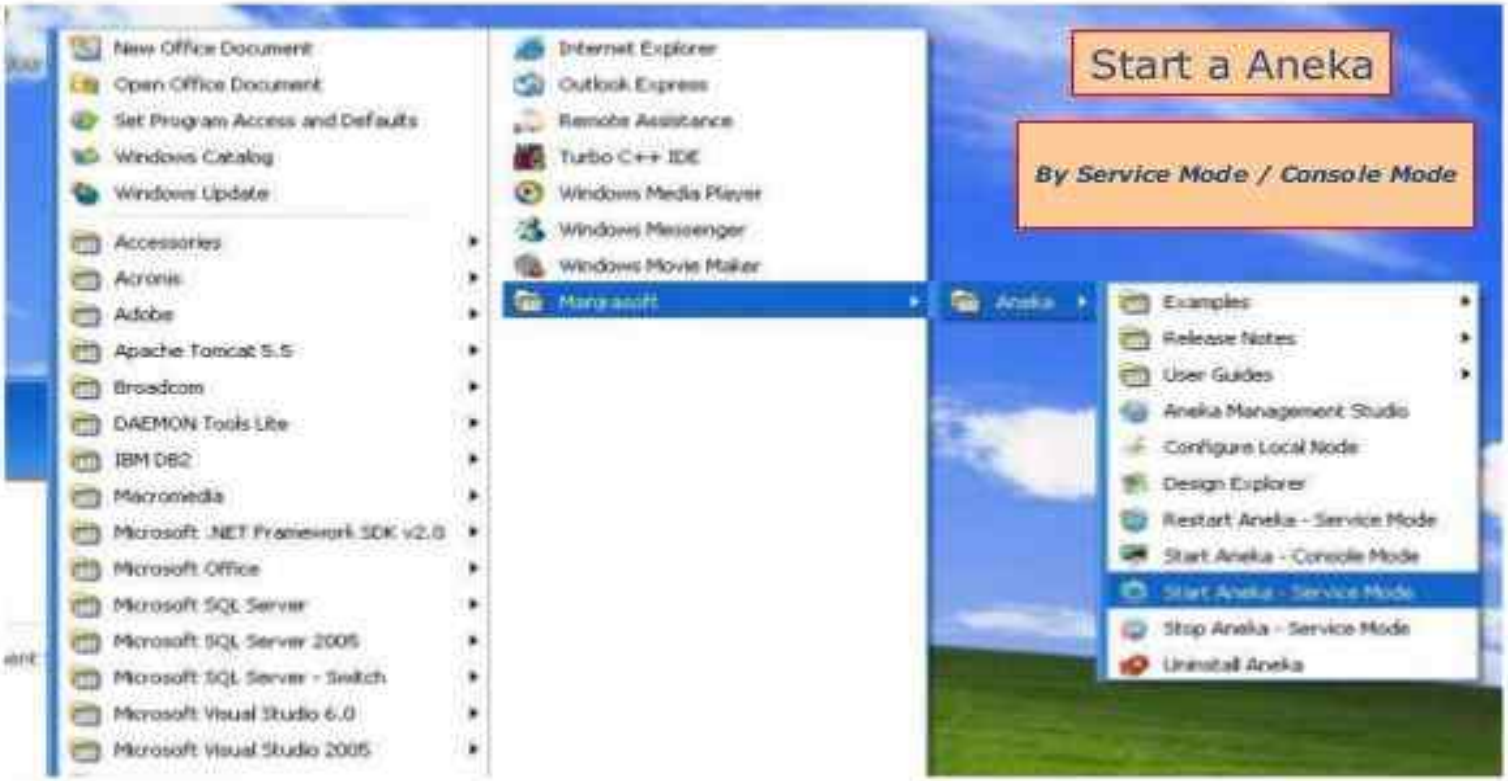
https://hemanthrajhemu.github.io

Node Selection



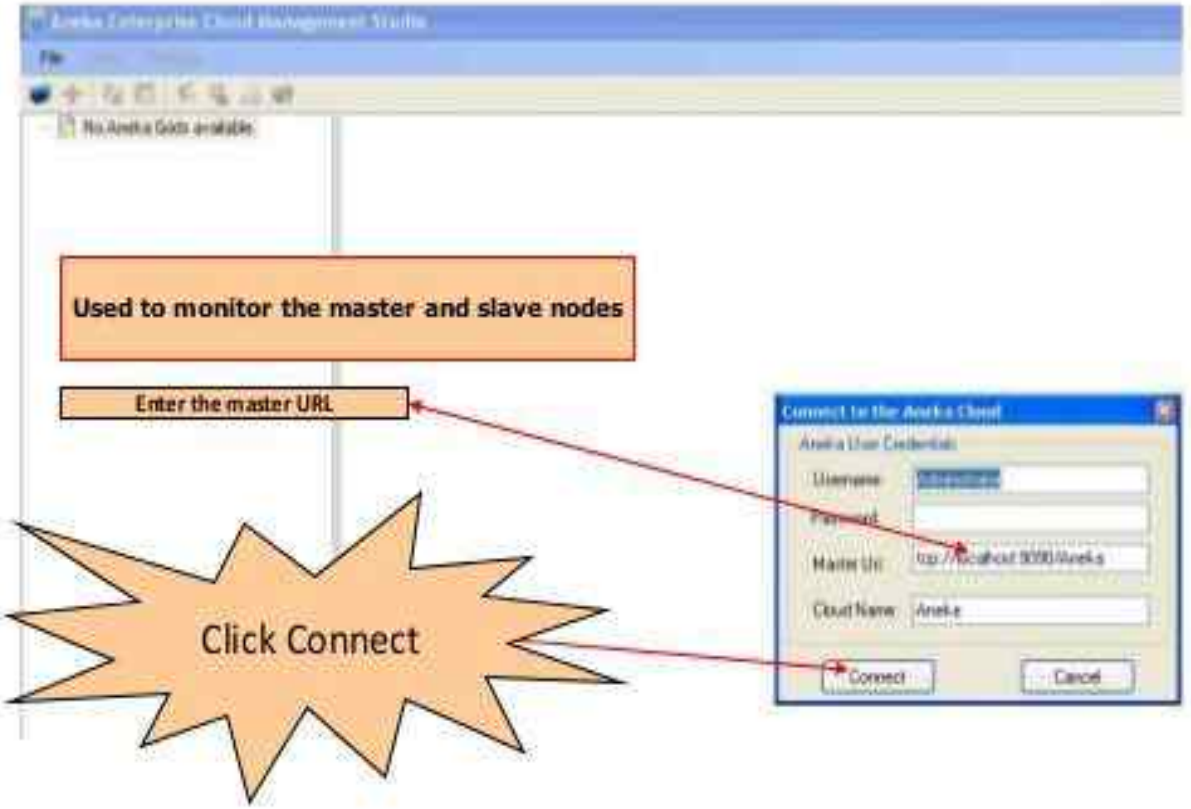
https://hemanthrajhemu.github.io

After the Installation,



https://hemanthrajhemu.github.io

Aneka Enterprise Cloud Management Studio



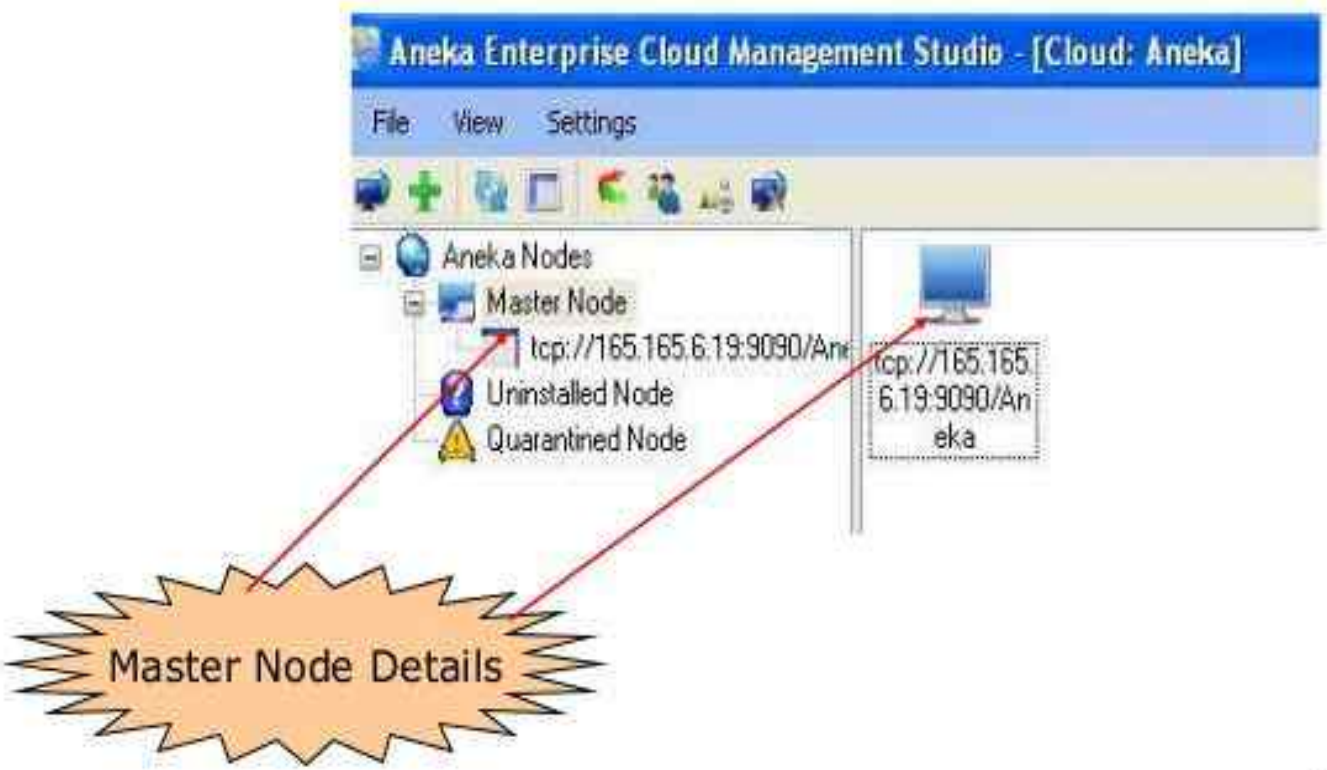
https://hemanthrajhemu.github.io

Windows Credentials

Windows Authentication Details

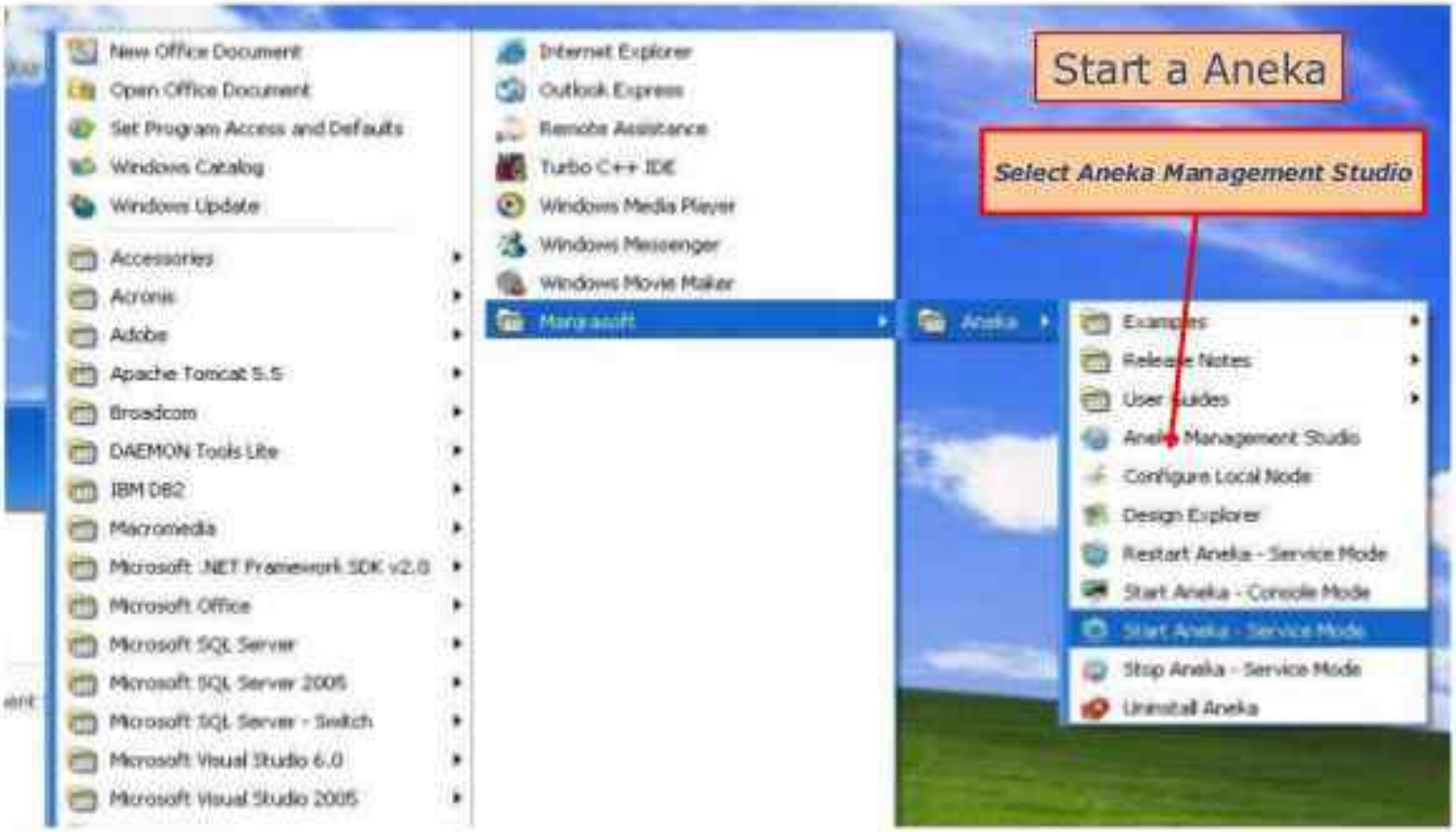
https://hemanthrajhemu.github.io

After Connecting…



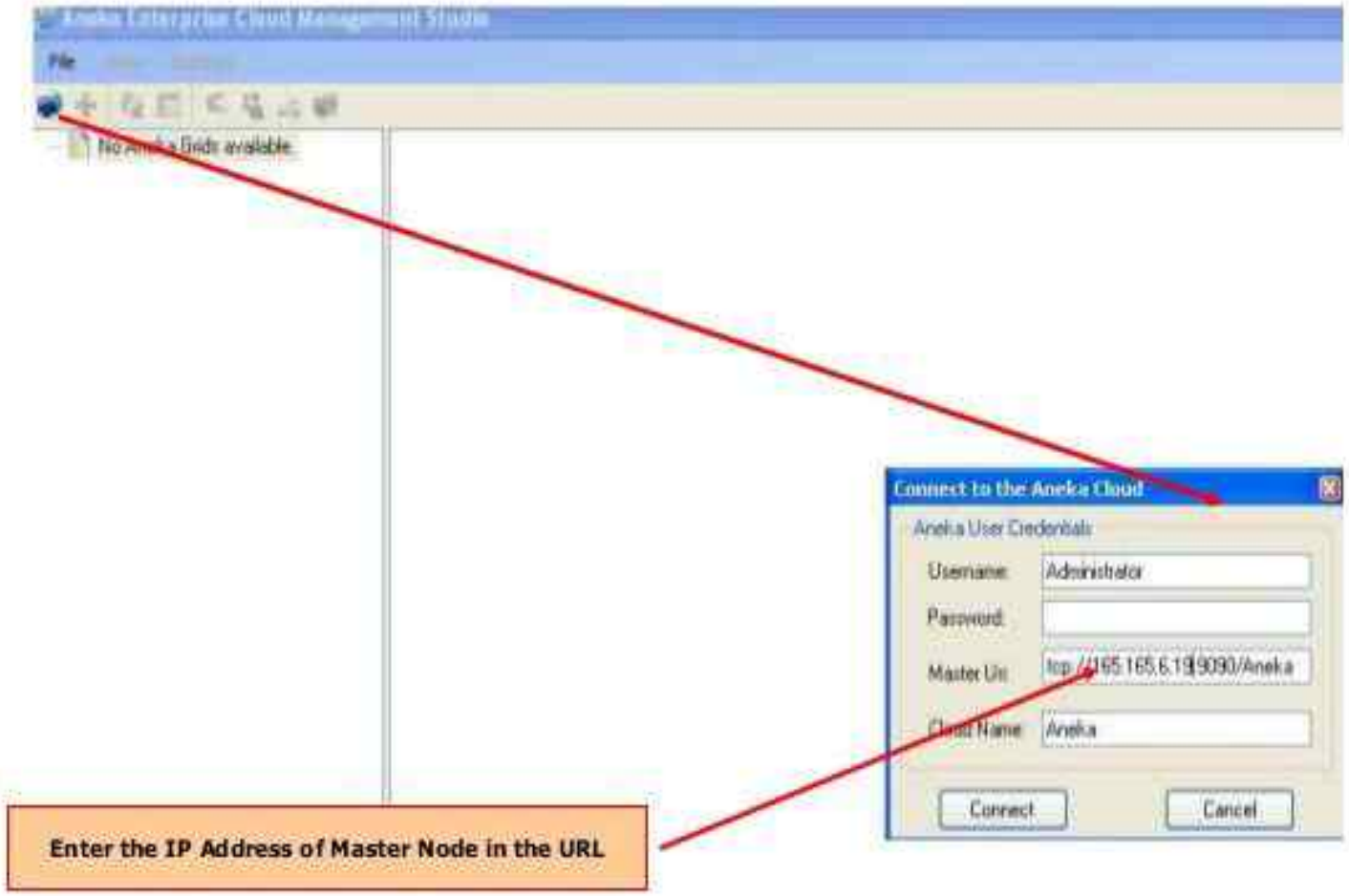
https://hemanthrajhemu.github.io

Aneka Node Management



https://hemanthrajhemu.github.io

Aneka Node Management

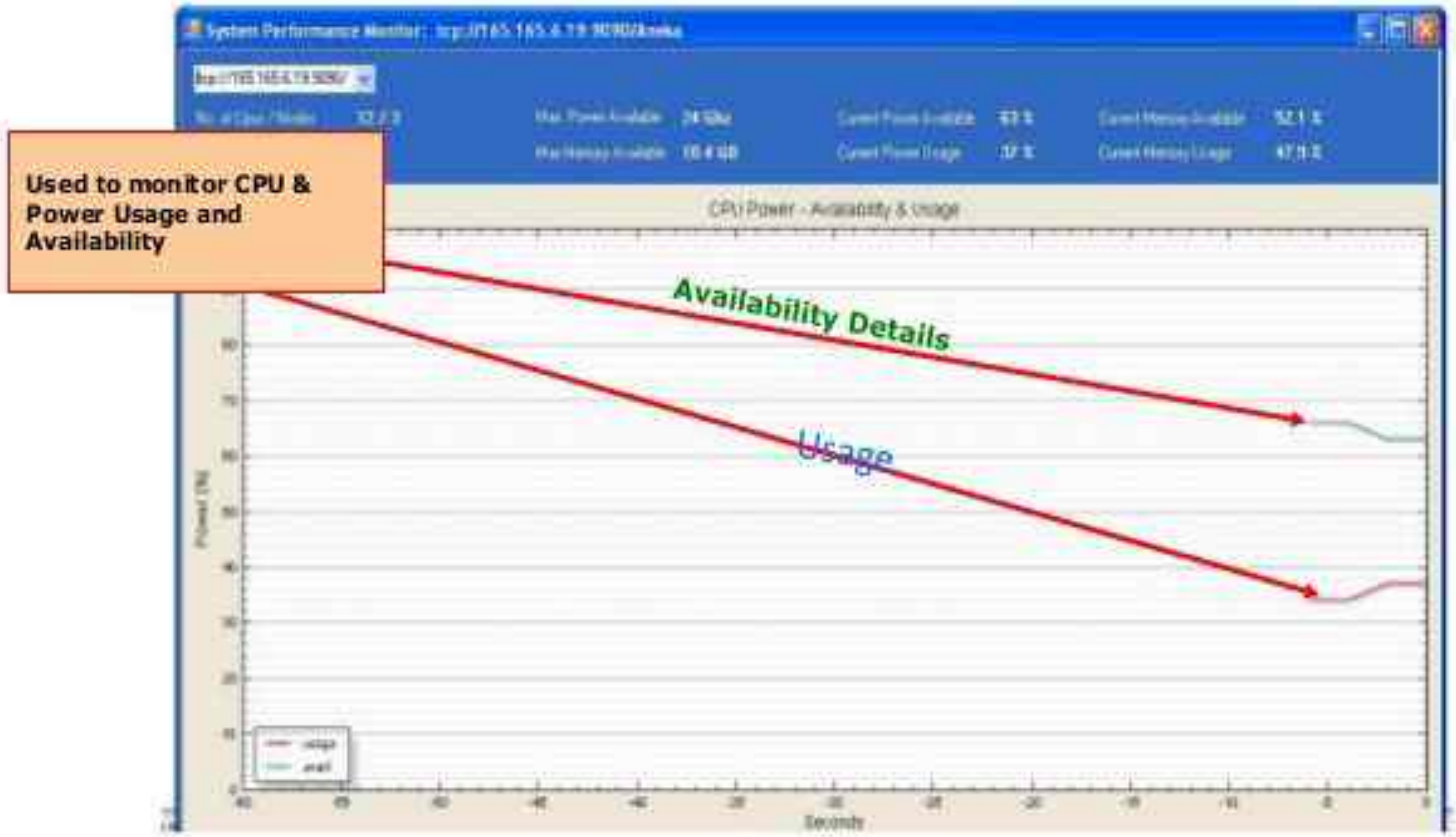


https://hemanthrajhemu.github.io

Aneka Network Overview

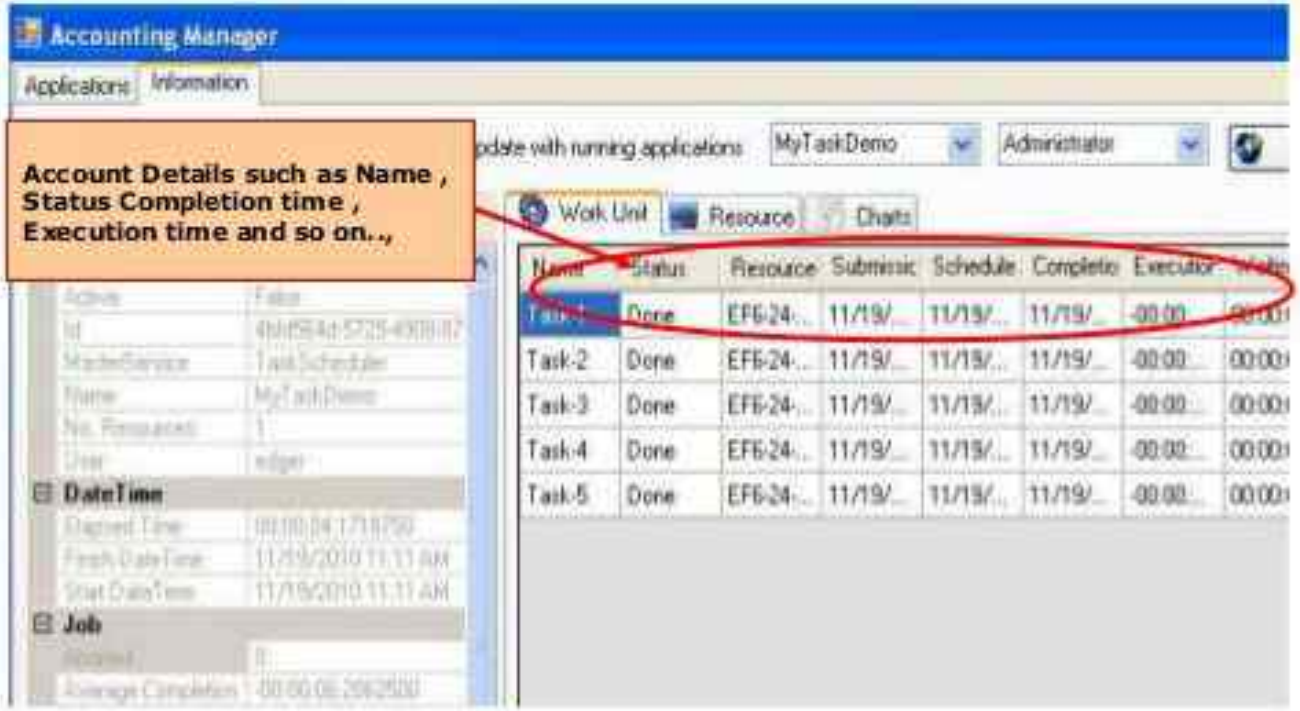
https://hemanthrajhemu.github.io

System Performance Monitor



https://hemanthrajhemu.github.io

Accounting Manager



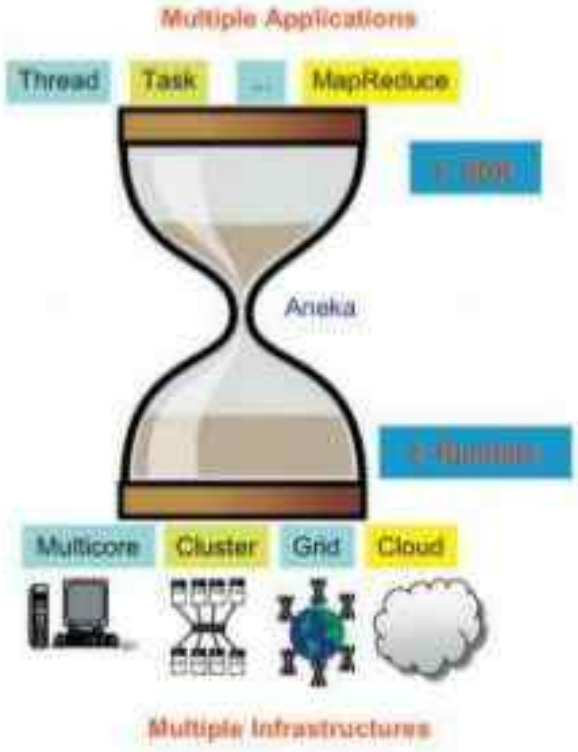
https://hemanthrajhemu.github.io

What Aneka Expect from User

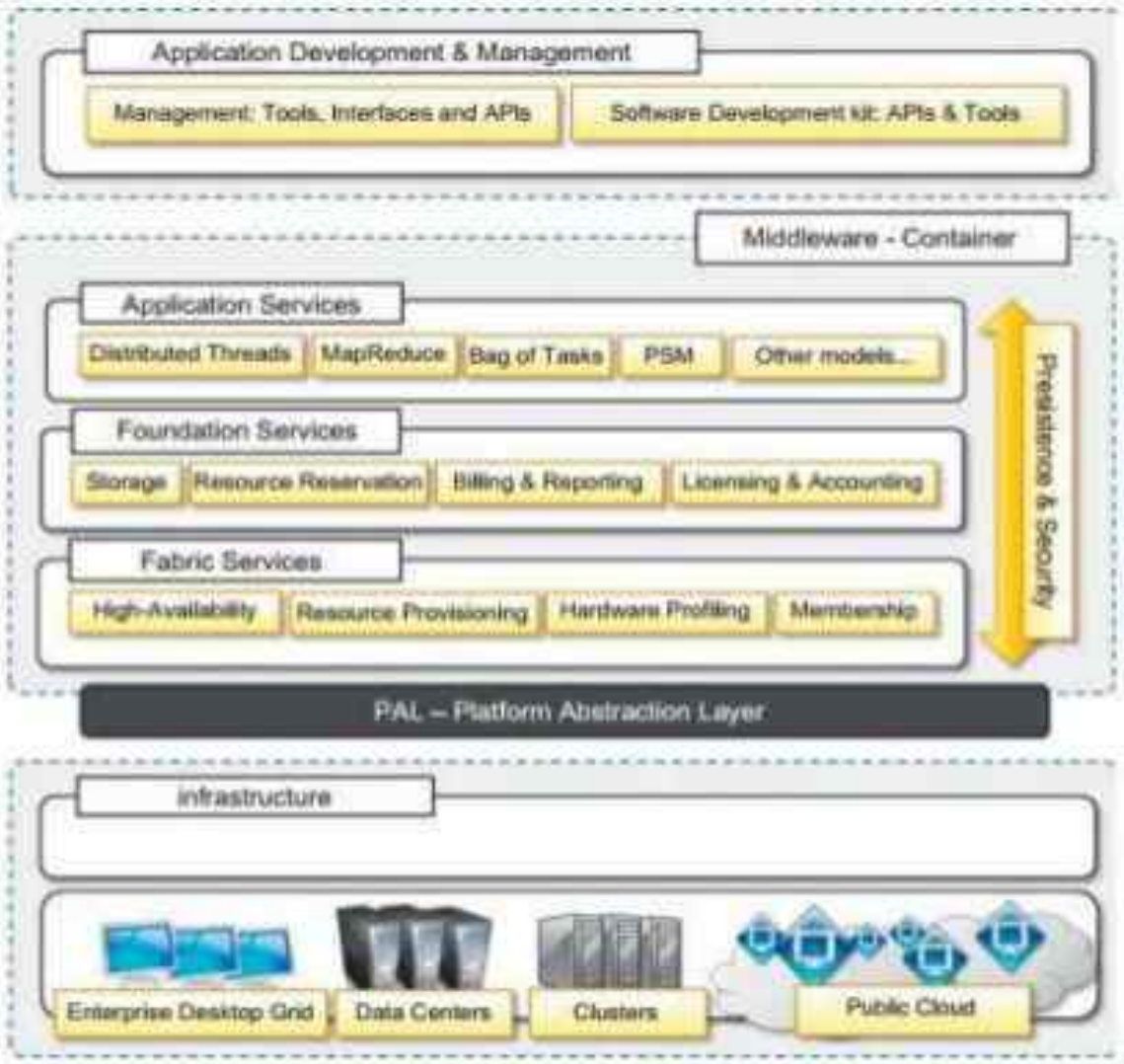
- Build new applications – Custom Model
- Make use of Aneka in existing applications to speed up the execution
- Build Commercial relationships
- Discuss investment and Business Opportunities

https://hemanthrajhemu.github.io

Aneka Overview



Aneka Frameworks



https://hemanthrajhemu.github.io

Platform Abstraction Layer(PAL)

- Core infrastructure of the system is based on .NET technology
- PAL provided features:
  - Uniform and platform-independent
    - implementation interface for accessing the hosting platform
    - access to remote nodes
    - management interfaces
  - Uniform access to extended and additional properties of the hosting platform

https://hemanthrajhemu.github.io

Anatomy of the Aneka container

- Fabric Services
  - A) Profiling and Monitoring Services
  - B) Resource Management Services
- Foundation Services
  - A) Storage management for applications
  - B) Accounting, billing, and resource pricing
  - C) Resource reservation
- Application Services
  - A) Scheduling
  - B) Execution

https://hemanthrajhemu.github.io

Fabric Services

- Lowest level of the software stack representing Aneka container
- Consists of
  - A) Profiling and Monitoring Services
  - B) Resource Management Services

https://hemanthrajhemu.github.io

Fabric Services contd..

- A) Profiling and Monitoring Services
  - Heartbeat, Monitoring and Reporting services
  - Heartbeat service periodically collects the dynamic information about the node
  - The basic information about memory space, disk space, CPU and operating system are collected.
  - All these information can be stored on RDBMS or a flat file.

https://hemanthrajhemu.github.io

Fabric Services contd..

- B) Resource Management Services
  - Comprises tasks: resource membership, resource reservation and resource provisioning service
  - Equivalent services: Index Service(Membership catalogue), Reservation Service, Resource Provisioning Service
  - The Membership catalogue tracks the performance information of nodes
  - The Resource Provisioning Service tracks the provisioning and lifetime information of virtual nodes.

https://hemanthrajhemu.github.io

Foundation Services

- Logical management of the distributed system built on top of the infrastructure
- A) Storage management for applications
- B) Accounting, billing, and resource pricing
- C) Resource reservation

https://hemanthrajhemu.github.io

Foundation Services contd..

- A) Storage management
  - Centralized file storage
    - More suitable for compute-intensive applications
  - Distributed file storage
    - More suitable for data intensive applications
  - FTP is default option installed in Aneka
  - To support different protocols, the concept of *file channel*, is introduced.
  - *File Channel* identifies a pair of components:
    - file channel controller : server part
    - file channel handler : client part
  - Storage service supports the execution of Different programming Model

https://hemanthrajhemu.github.io

Foundation Services contd..

- B) Accounting, Billing, and Resource Pricing
  - Accounting keeps track of the status of applications in the Aneka cloud
  - Shows the usage of infrastructure and the execution of applications
  - Billing service provides detailed information about the resource usage of each user with the associated costs.
  - Each resource can be priced differently according to the different set of services that are available on the corresponding Aneka container or the installed software in the node.

https://hemanthrajhemu.github.io

Foundation Services contd..

- B) Resource Reservation
  - Supports the execution of distributed applications
  - Allows for reserving resources for exclusive use by specific applications

https://hemanthrajhemu.github.io

Application Services

- Manage the execution of applications
- Constitute a layer that differentiates according to the specific programming model
- *Scheduling Service* and *Execution Service*
- *Scheduling Service* tasks:
  - Job-to-node mapping
  - Rescheduling of failed jobs
  - Job status monitoring
  - Application status monitoring

https://hemanthrajhemu.github.io

Application Services contd..

- Execution Service tasks:
  - Controls the execution of single jobs that compose applications
  - Unpacking the jobs received from the scheduler
  - Retrieval of input files required for the job execution
  - Sandboxed execution of jobs
  - Submission of output files at the end of execution
  - Execution failure management
  - Performance monitoring
  - Packing jobs and sending them back to the scheduler

https://hemanthrajhemu.github.io

Application Services contd..

- Currently supported programming model in the Aneka Cloud:
  - Task Model
  - Thread Model
  - MapReduce Model
  - Parameter Sweep Model

https://hemanthrajhemu.github.io

Building Aneka clouds

- Aneka is primarily a platform for developing distributed applications for clouds
- Infrastructure management tools
- Administrators
- public, private, and hybrid clouds.

https://hemanthrajhemu.github.io

Infrastructure organization

Logical organization

- depends on the configuration
- The most common scenario is to use a master-worker configuration with separate nodes for storage

Master Node

- A common configuration of the master node is as follows:
- Index Service (master copy)
- Heartbeat Service
- Logging Service
- Reservation Service
- Resource Provisioning Service
- Accounting Service
- Reporting and Monitoring Service
- Scheduling Services for the supported programming models

https://hemanthrajhemu.github.io

Storage Node

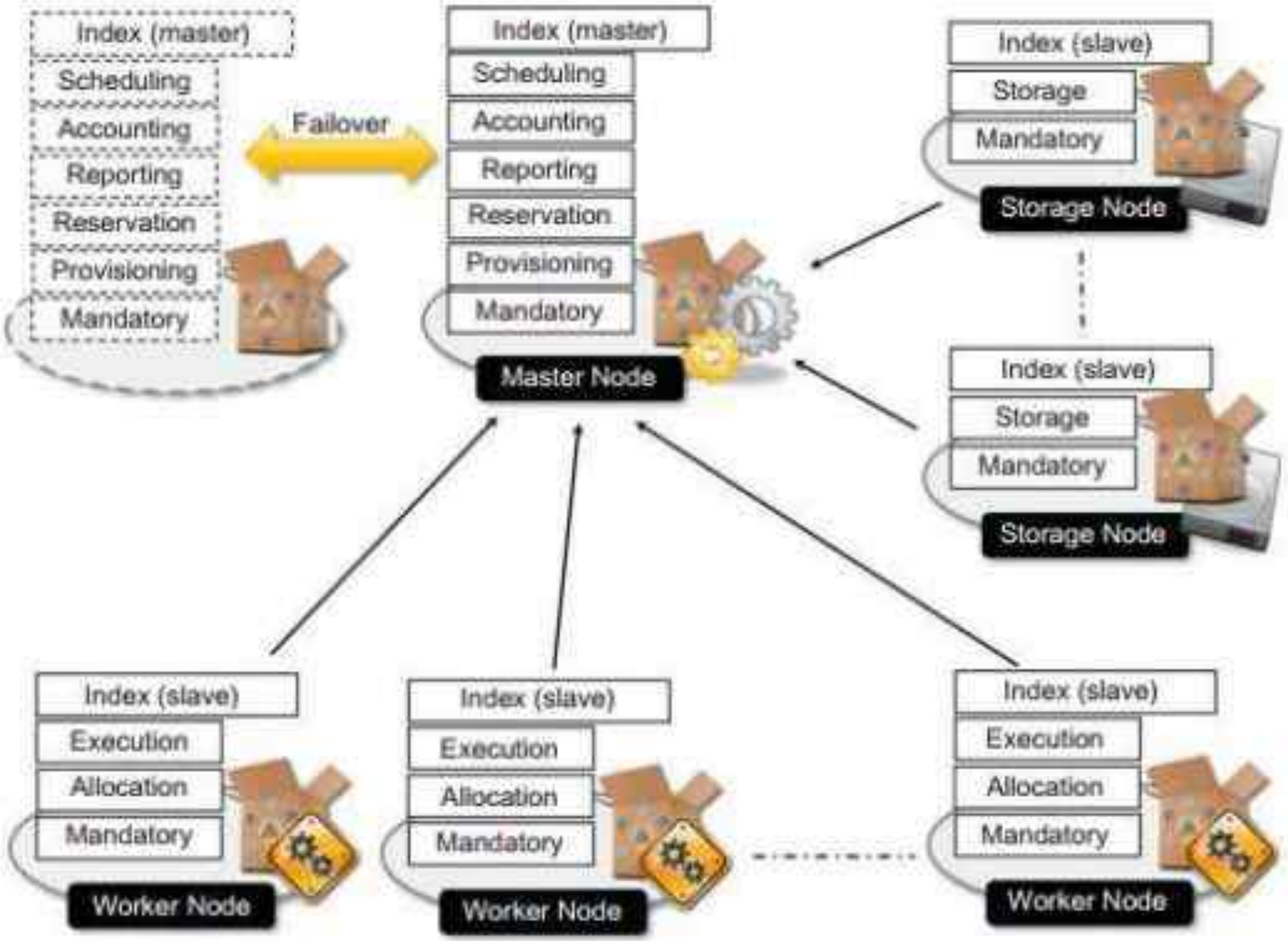
- The common configuration of a storage node is the following:
- Index Service
- Heartbeat Service
- Logging Service
- Monitoring Service • Storage Service

https://hemanthrajhemu.github.io

Worker Node

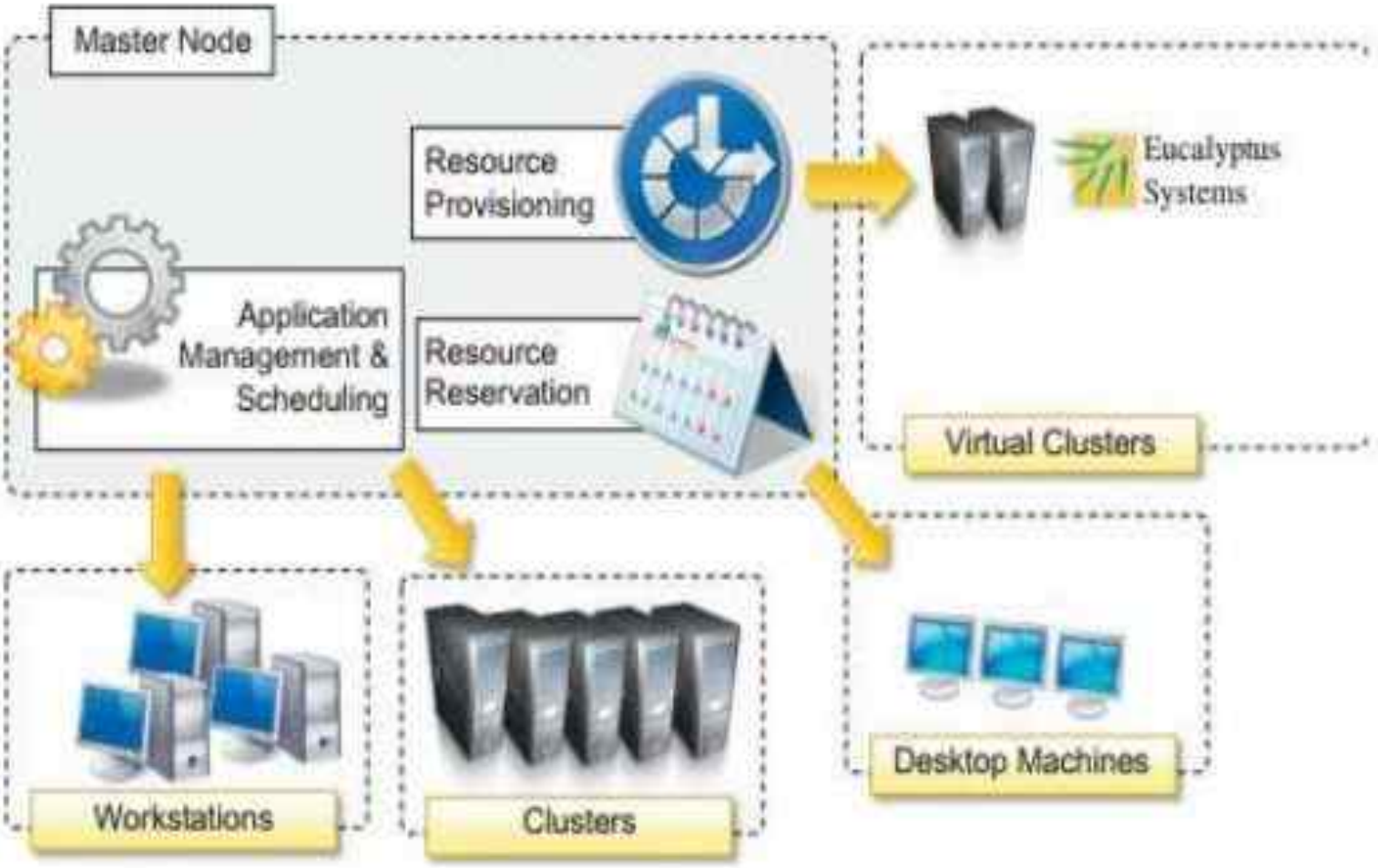
- Index Service
- Heartbeat Service
- Logging Service
- Allocation Service
- Monitoring Service
- Execution Services for the supported programming models

https://hemanthrajhemu.github.io



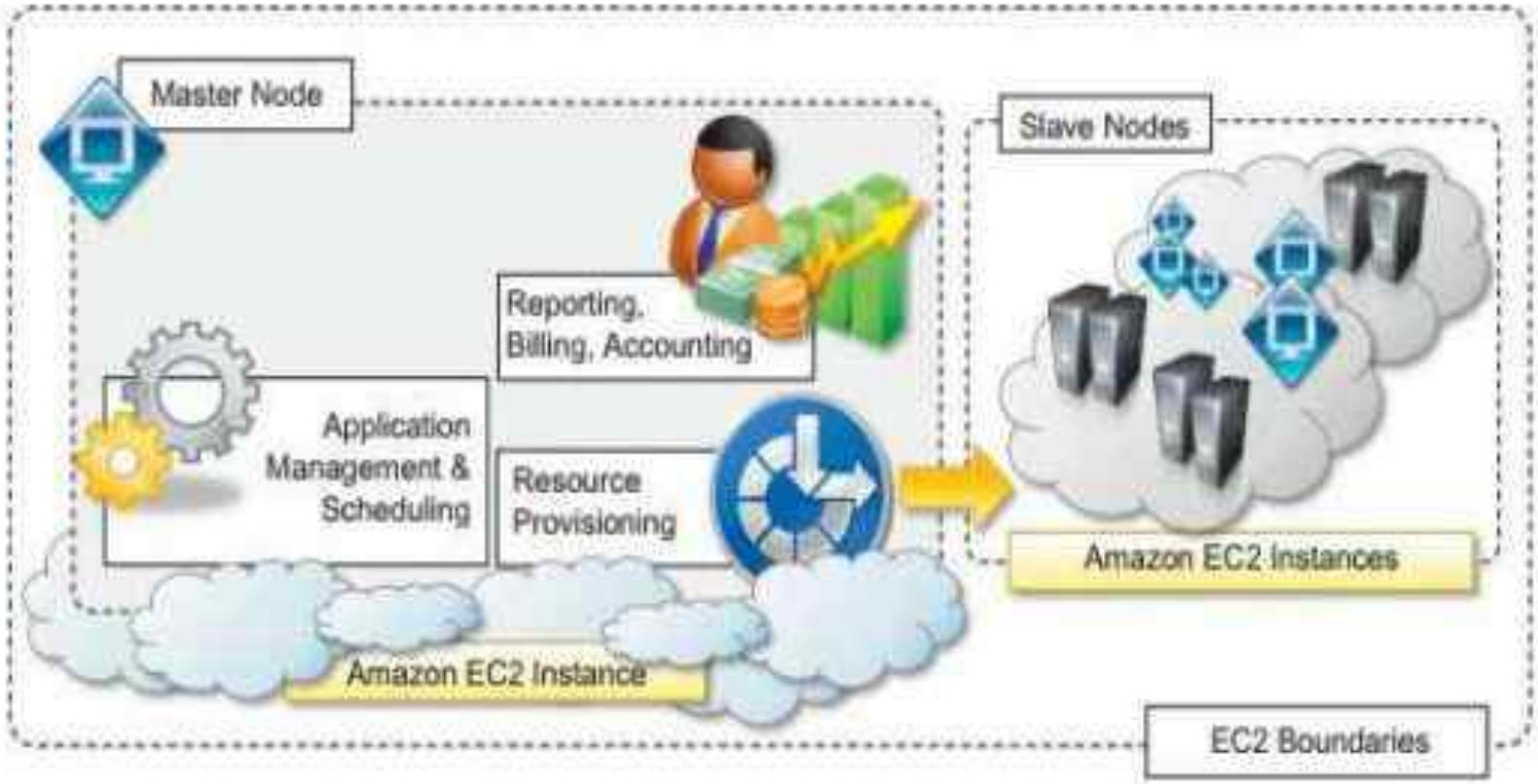
https://hemanthrajhemu.github.io

Private cloud deployment mode

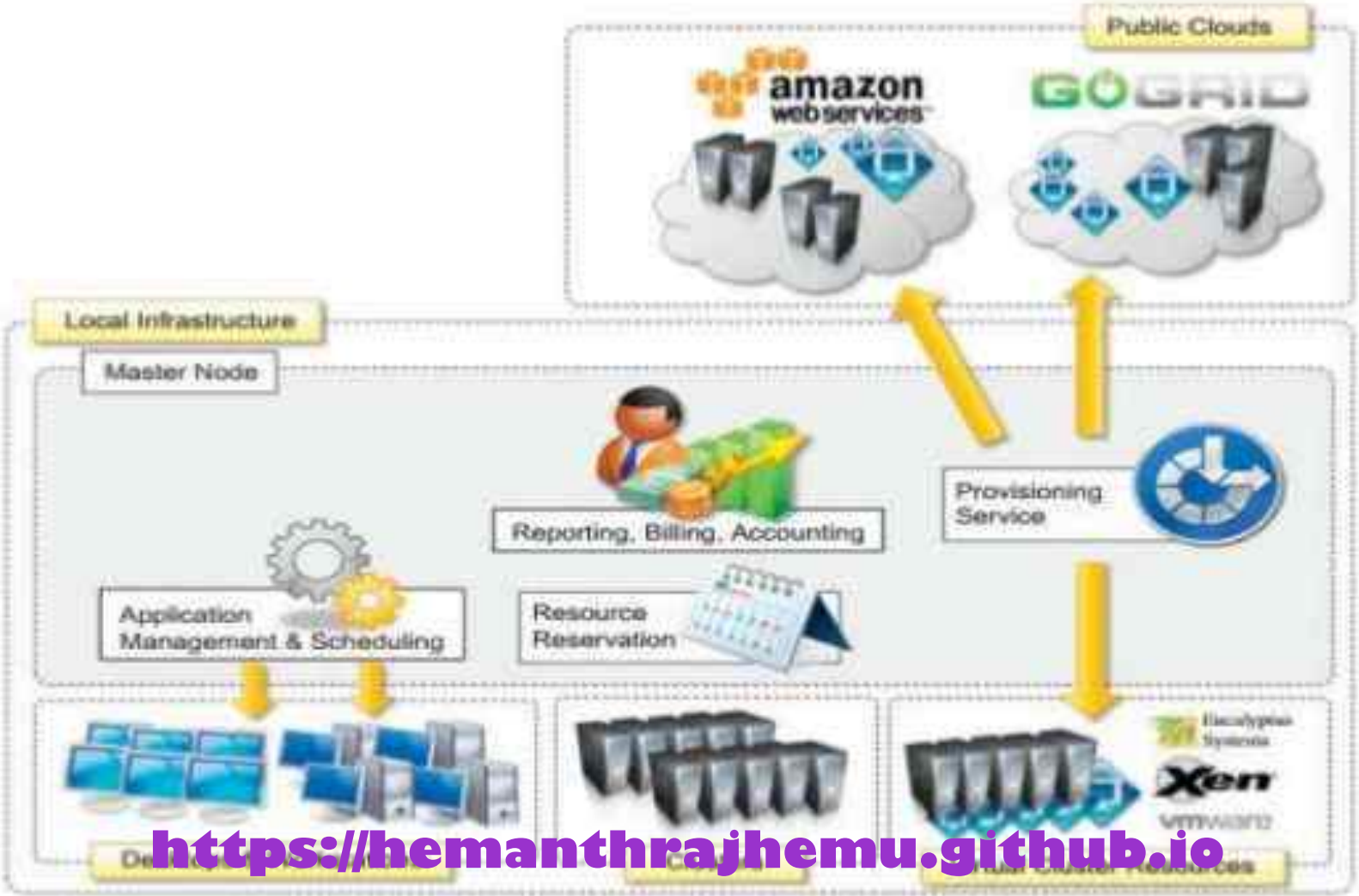


https://hemanthrajhemu.github.io

Public cloud deployment mode



Hybrid cloud deployment



https://hemanthrajhemu.github.io

Cloud programming and management

- Aneka SDK
  - -Application model
  - -Service model
- Management tools
  - -Infrastructure management
  - -Platform management
  - -Application management

https://hemanthrajhemu.github.io

Cloud programming and management

- Provide a scalable middleware product
- distributed applications
- Application development – Developer
- Management – System administrator
- Aneka SDK
- Management Tools

https://hemanthrajhemu.github.io

Aneka SDK

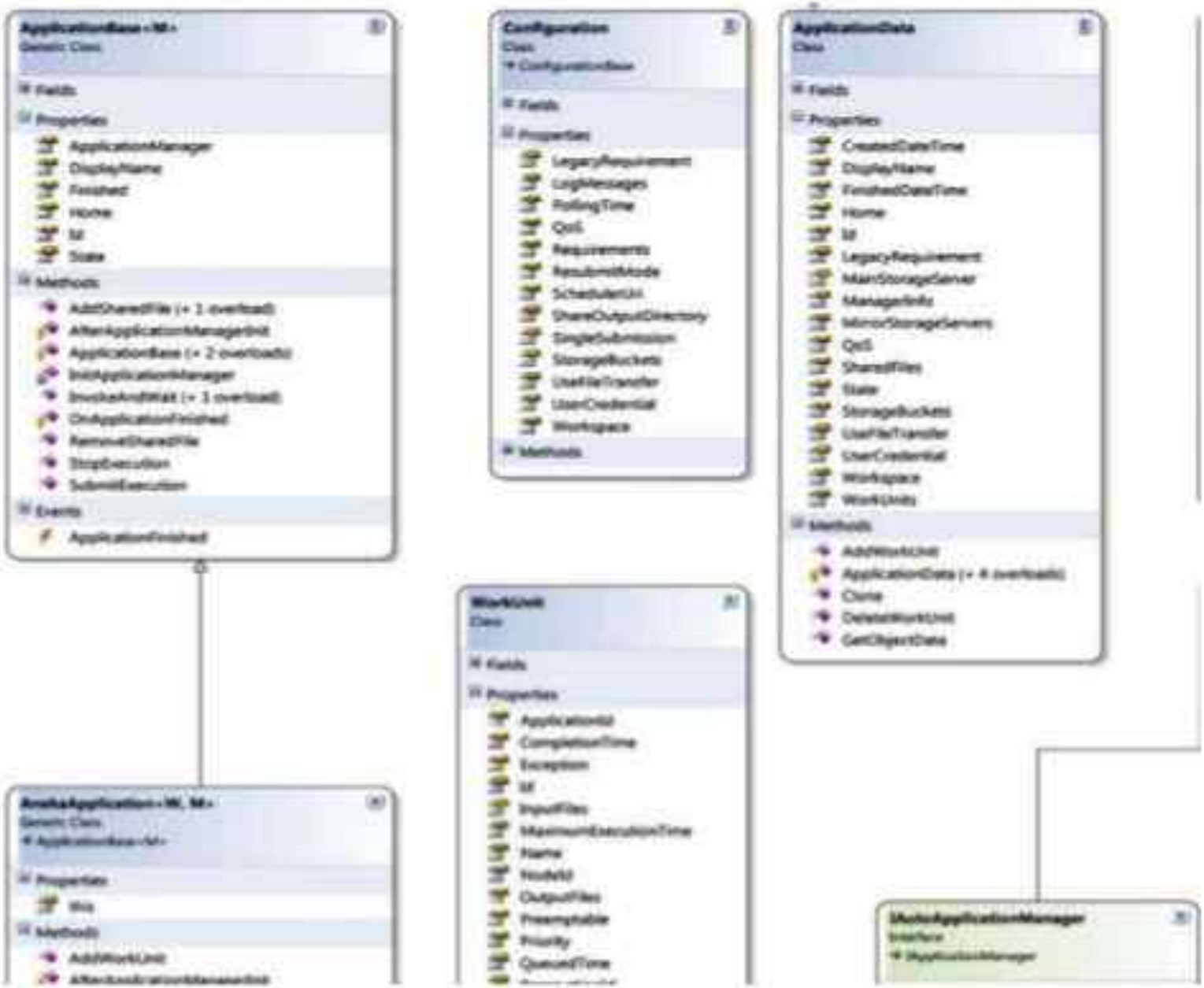
- Application Model
- Service Model
- Aneka provides APIs for developing applications
- use of existing features
- both programming models and services

https://hemanthrajhemu.github.io

Application Model

- Aneka provides support for distributed execution in the Cloud with the abstraction of programming models.
- used by the developers and the runtime support for the execution of programs on top of Aneka
- The Application Model represents the minimum set of APIs that is common to all the programming models

https://hemanthrajhemu.github.io



https://hemanthrajhemu.github.io

Table 5.1 Aneka's Application Model Features

Category	Description	Base Application Type	Work Units?	Programming Models
Manual	Units of work are generated by the user and submitted through the application.	*AnekaApplication<W,M>* *IManualApplicationManager<W>* *ManualApplicationManager<W>*	Yes	Task Model Thread Model Parameter Sweep Model
Auto	Units of work are generated by the runtime infrastructure and managed internally.	*ApplicationBase<M>* *IAutoApplicationManager*	No	*MapReduce* Model

https://hemanthrajhemu.github.io

Service model

- The Aneka Service Model defines the basic requirements to implement a service that can be hosted in an Aneka Cloud.
- IService interface
- Name and status
- Control operations such as Start, Stop, Pause, and Continue methods
- Message handling by means of the HandleMessage method

https://hemanthrajhemu.github.io

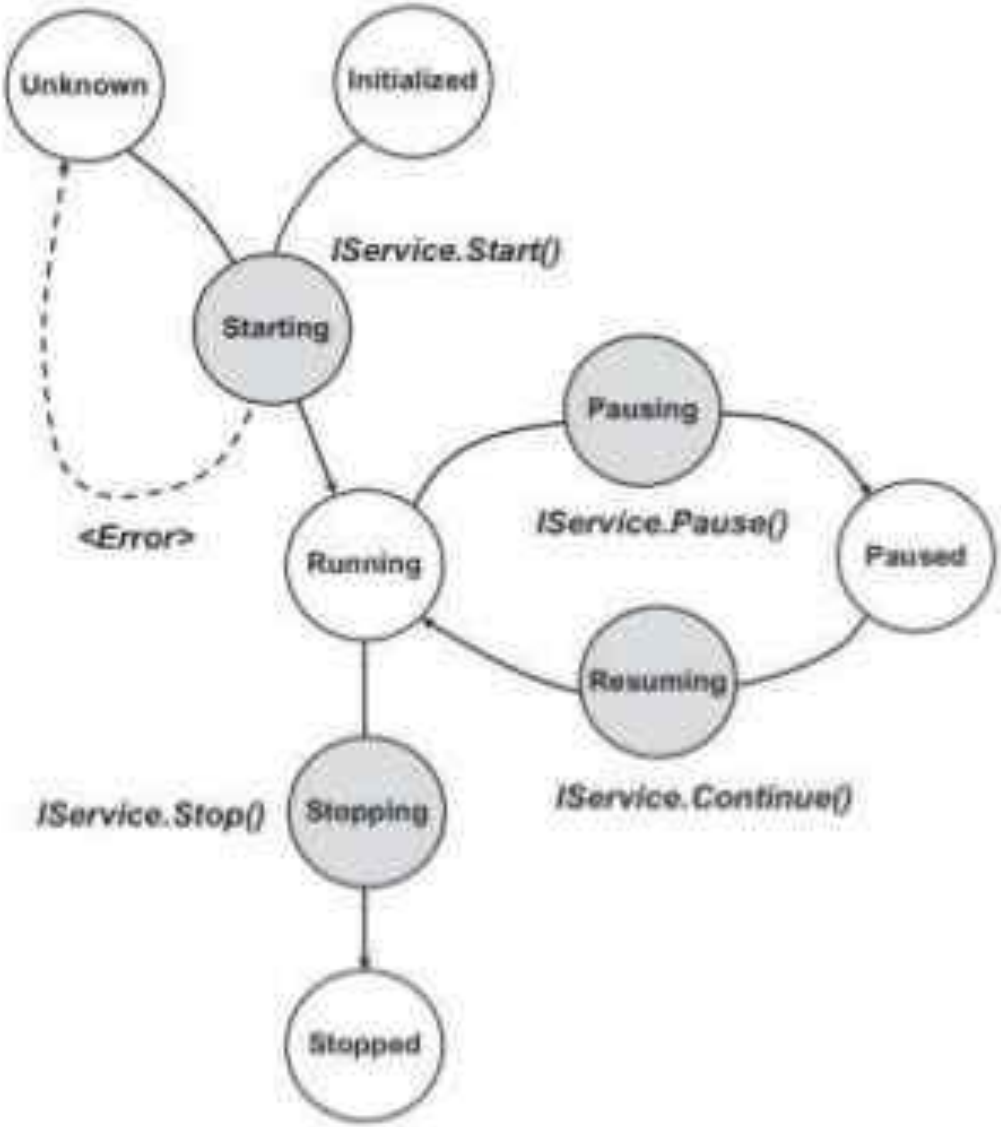


FIGURE 5.9

Service life cycle.

https://hemanthrajhemu.github.io

Management tools

- Infrastructure management
- Platform management
- Application management

https://hemanthrajhemu.github.io