CoreOS

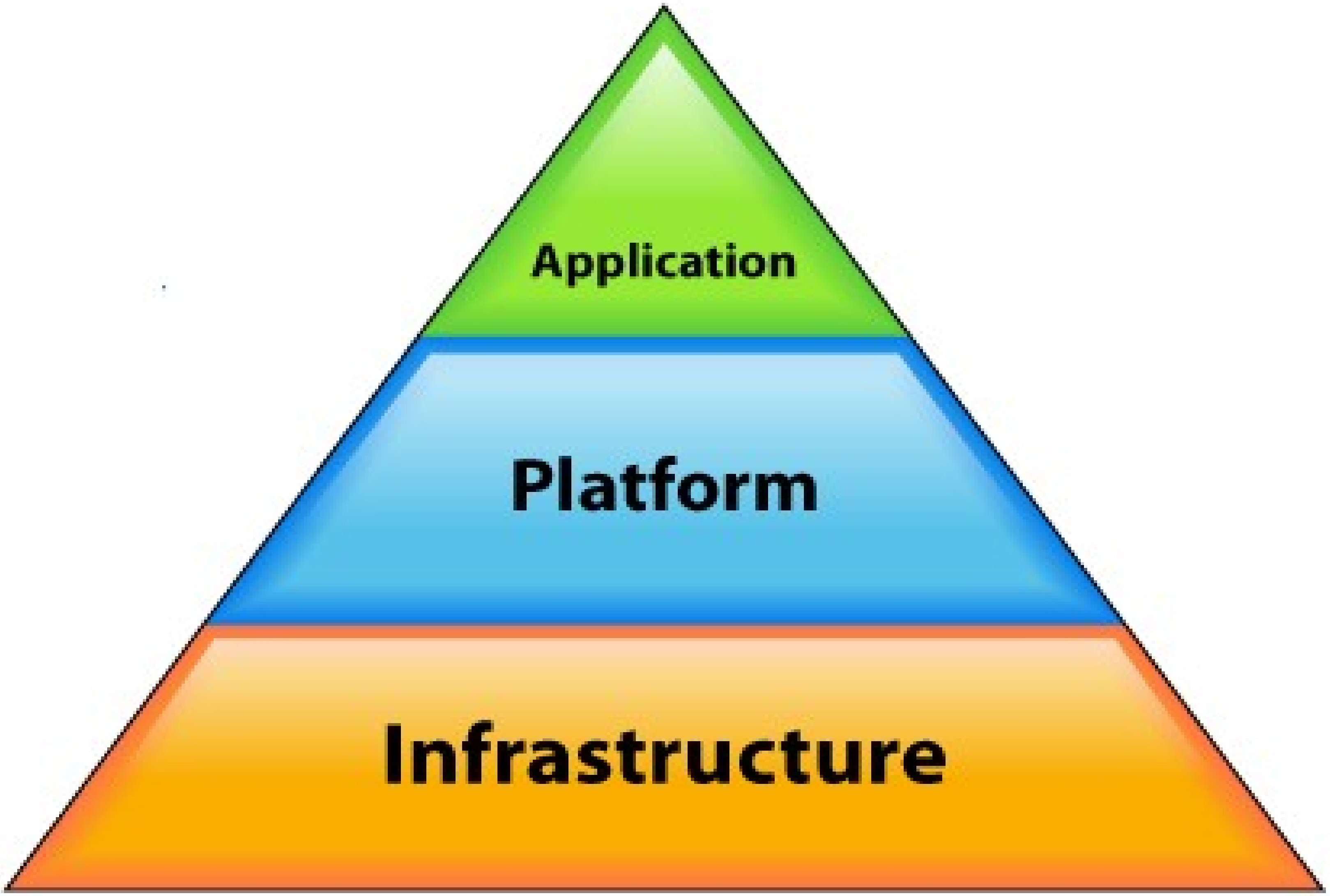
گروه کاربران لینوکس تهران

@majidazimi
majid.azimi@live.com

Problem

Design a PAAS Cloud

Cloud



Challenges

- Minimal OS maintenance
  - OS installation
  - Upgrade management
  - How many admin per n servers
- Agility
- Configuration management
  - Nginx
  - MySQL

Challenges

- Financial management
  - How much user should pay for a mysql instance?
- Launch n instance of same application
- Automation
  - Every possible configuration should be programmable
- HA

CoreOS

CoreOS is an open source lightweight operating system based on the Linux kernel, designed for providing infrastructure to clustered deployments

Wikipedia

CoreOS

- Read-only rootfs
- etcd
- Docker
- Systemd / Fleet
- Locksmith

Read-only rootfs

- Minimal OS (200MB)
- Fork of ChromeOS
- No package manager
- PXE Boot
- Automatic update
  - Automatic updates are under control

etcd

A highly-available Key/Value store for shared configuration and service discovery inspired by Apache Zookeeper.

etcd

- Data is stored as file system path
- Each node contain Key/Value data
- Atomic operation
- Notification when path changes
- Auto-Incrementing path generator
- Ephemeral nodes
- Rest interface
  - Apache zookeeper uses native protocol

etcd

Data is stored as file system path

/master

/worker/worker-1

etcd

Each node contain Key/Value data

/master contains:

- IP: 192.168.1.12
- Port: 5487

/worker/worker-1:

- IP: 192.168.1.154
- Port: 1547

etcd

Atomic operation

- When multiple clients want to create /master only one of them wins.
- When multiple clients want to add Key/Value to /workers/worker-1 only one of them wins.

etcd

Notification when path attributes changes

- Clients can set watches on a path
- When a path deleted, notification will be pushed to all clients watching that path.
- When a path data is changed, notification will be pushed to all clients watching that path.

etcd

Auto-Incrementing path generator

- client-1 requests to create a child in /workers/
- Server responds with /worker/worker-1

- client-2 requests to create a child in /workers/
- Server responds with /worker/worker-2

etcd

Ephemeral nodes

Client-1 registers /node

If client-1 is disconnected from etcd cluster then /node will be deleted.

Etcd: hot stand by master process

1. All clients try to create ephemeral /master with their IP/Port as data
2. Due to atomic operation only one of them wins
    1. The winner process becomes as master
3. All other process set a watch on /master
4. If master goes down /master will be deleted
5. All clients will be notified that /master is gone.
6. Goto 1

Apache Curator

- Leader election
- Shared Reentrant Lock
- Shared Reentrant Read Write Lock
- Shared Semaphore
- Distributed Atomic Counter
- Simple Distributed Queue
- Distributed Priority Queue
- Distributed Delay Queue

Systemd / Fleet

Distributed init System

- Deploy a single container anywhere on the cluster
- Deploy multiple copies of the same container
- Ensure that containers are deployed together on the same machine
- Maintain N containers of a service, re-deploying on failure
- Deploy containers on machines matching specific metadata

Locksmith

Reboot manager for cluster

- Control which mashine will be rebooted after update.
- All coordination is done via etcd
  - Shared semaphore

Docker

- Light-weight virtualization
- Application container using LXC
- Use cgroups for isolation of processes

?