DAKU cloud project

Functions :

- upload
- delete
- restore
- download
- run beast

Scalability

- 1 Master server + Multiple container servers architecture
- Experiments with horizontal and vertical scalability
- Load Balancing (random + priority queue (using Redis Sorted Set))

Fault Tolerance

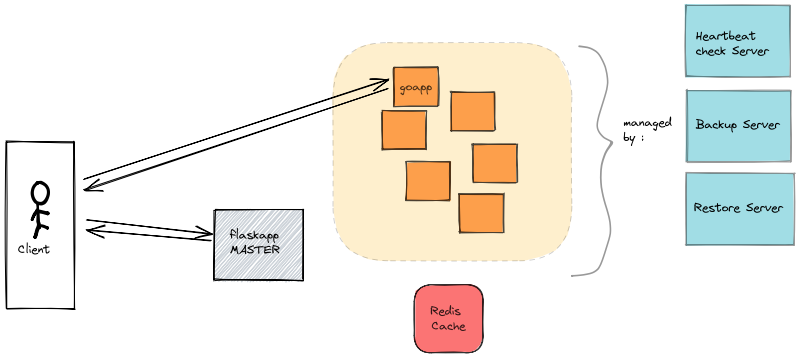
- Hourly backups
- Automatic transfer of user to new VM
- Heartbeat check

Technologies used :

- Docker containers
- pydocker sdk for RPCs
- Dockerfile
- Golang for writing REST endpoints for goapp that runs inside the container
- Python for writing heartbeat check, keep real time track of system stats (sort of like a realtime dashboard), container backup and restoration
- Redis as real time cache stored in cloud and also for experimenting priority queue based load balancing using SortedSets
- Linode for purchasing servers
- AWS for purchasing SQL database and hosting Redis
- Linux Shell scripting for monitoring real time stats (top, watch ...)
- NGINX for hosting the web app
- Zerotier One for creating VPN (useful when testing inside college server labs)

Architecture diagram of our app

A cluster of servers that run docker containers



Real Time System Performance
Monitoring
- CPU usage
- RAM usage
- Disk Usage