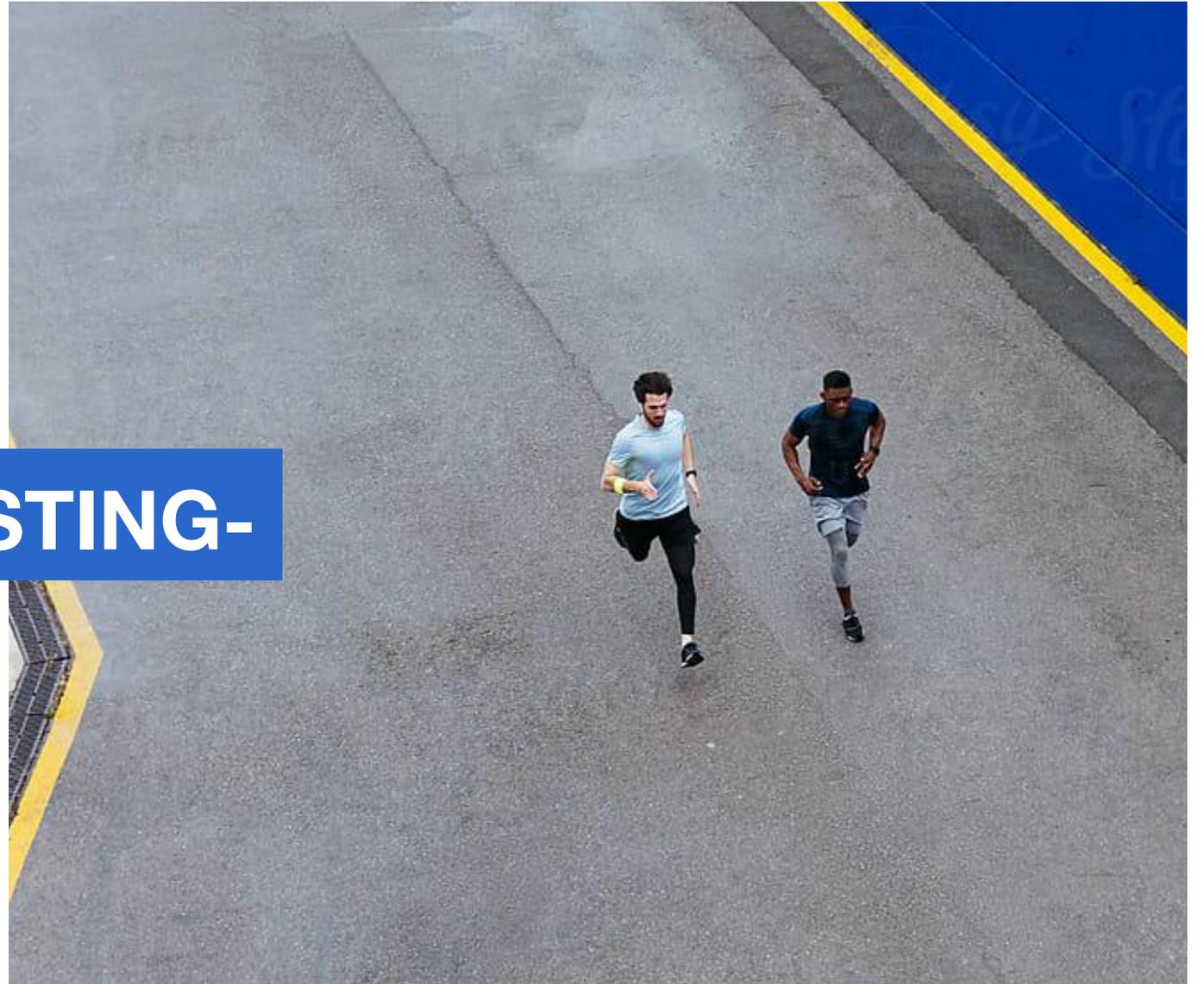




**THE FUTURE OF  
APPLICATION HOSTING-  
CONTAINERS  
AND FaaS**

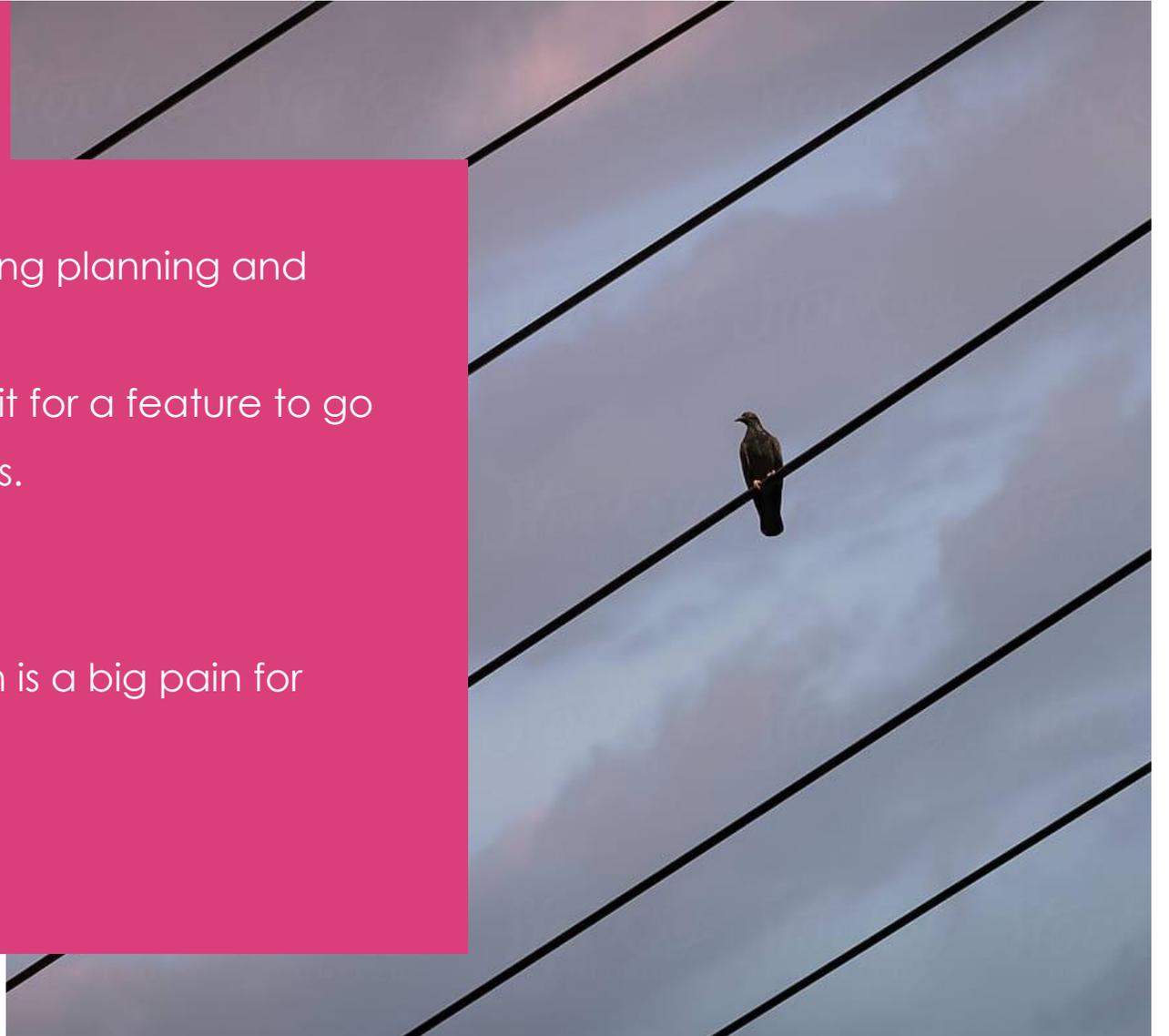


Organizations faces the biggest problems when developing an Application that needs its functions to be scaled and executed to serve millions of users at a time. Often they spent a fortune in setting up and managing the infrastructure it needs.



# CHALLENGES

- Application deployment takes long planning and take a lot of time to implement.
- Development teams need to wait for a feature to go live with limited testing possibilities.
- Downtimes
- High Availability
- Rolling back to a previous version is a big pain for Operations teams.



# THE SOLUTION

## An open source solution that supports:

- Configure a highly available, highly scalable and easy to manage cluster to run endless number of containers and functions.
- Easy to deploy and easy to manage is the way we should think in this era.
- Scale your application or a part of your application with just a click of a button.
- Use the rolling update functionality and forget downtime during updates and also this happens with just a click of a button.
- Want to revert 1000s of containers to the previous version ? just click Revert.

# TRADITIONAL ARCHITECTURE

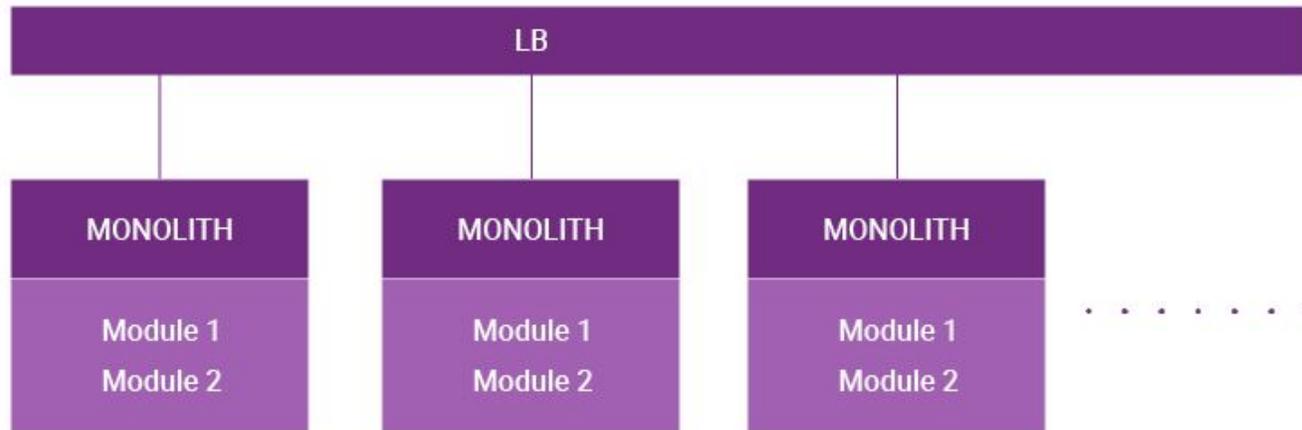


Fig.1

Traditional web development consists of a single application ( monolith ). In production, there would be multiple machines running this application and would have a load balancer distributing the load.

# CURRENT & FUTURE ARCHITECTURE

The modern web development is being more modular, that means, the application is further split into independent working functions. This method can leverage the latest technologies in the market like Docker and Docker swarm ( for high availability ).

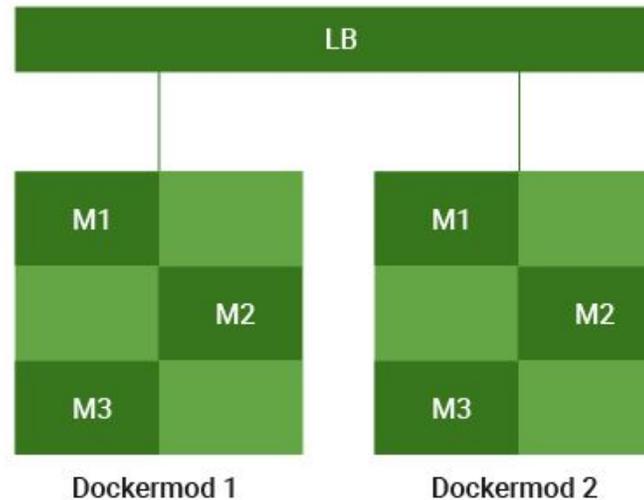


Fig.2

In fig 2, we see multiple docker instances hosting multiple containers. These containers are independent and each container can talk to each other to use its functionalities. But when the number of containers increases, it becomes extremely difficult to manage them.

# CURRENT & FUTURE ARCHITECTURE

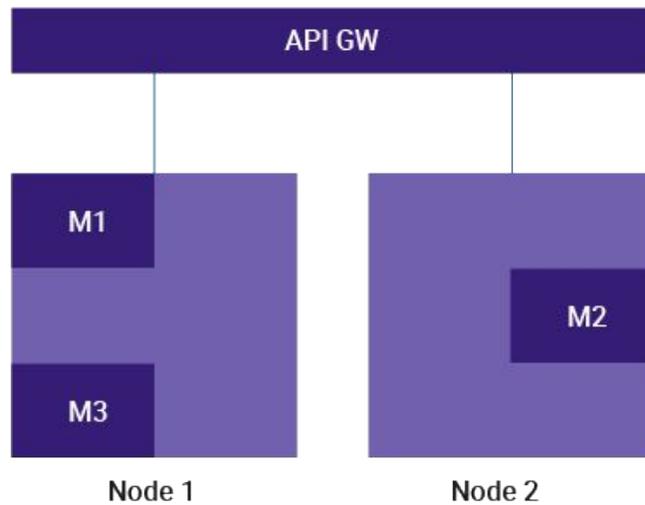


Fig.2

When container management becomes difficult , an orchestration layer comes in.

In Fig 3,, We see an API gateway and cluster of servers. Each micro service would have an API endpoint, and a developer does not have to bother about where in the cluster the service is running. Can scale the service with a click of a button, setup a rolling update scheme, have monitoring of the containers setup within the cluster so it spins up a new container if it detects an unhealthy one are some of the benefits thereby completely removing manual interventions

# ADVANTAGES

- You do not need to use public cloud when developing an Application that needs its functions to be scaled and executed in the 1000s. We can set up a serverless ( Functions as a Service ) offering in your VMware or any infrastructure.
- Move your application development into re-usable and highly scalable Microservices.
- As high availability is absolutely necessary, this is just the architecture you need.
- Create application groups to make managing them easy.

## WHO NEEDS IT?

Application development is moving to microservices. And microservice orchestration is very important. It can get very messy quickly if you don't organise your containers. This is where giants like Apache and Google dedicated their resources in building a orchestration layer on top of docker. Solutions like Kubernetes have been massively successful in managing container, and Mesos in the pioneer in controlling a big scale compute cluster. We can run containers to simple shell commands.

When Software Development has moved to Agile, organizations needs deployments to be quick and easy scalable

# ROLE OF OPENSOURCE

## Apache Mesos

- **Apache Mesos** is an open-source project to manage computer clusters.
- Can span across any number of server, creating a central compute cluster
- Easy Rest-full apis for integrating with different application
- Run containers freely and scale them, update them, destroy them with just a few clicks
- Use resources only when needed (faas)

# ROLE OF OPENSOURCE

## Docker

- Containers are capsules that contain your application and all its dependencies
- These containers can run on any environment with the help of docker
- With having containers, operations do not have the pain of setting up dependencies for the application to run ( as a container contains all the application dependencies)
- Docker has a major role in modernizing the application infrastructure ( with the birth micro services concept )
- The also have a role to play in the upcoming faas technology

# ROLE OF OPENSOURCE

## Kubernetes

- Best software to run sets of micro services that need to be highly available.
- Reduces operational efforts as we only need to maintain the physical/virtual node, all the applications running in it can easily be managed by kubernetes
- More apt for services that depend on each other ( with its onboard networking, it makes communication throughout the cluster easy)
- Monitor, replicate and scale based on various criteria.
- Easy to get going. Setting up a production grade cluster would hardly take an hour.

# ABOUT US

PiServe is a leader in Automation, Application Development/Support, Infrastructure Management Services and Cloud.

We leverage our experience, knowledge and services to help build trust and confidence in the capital markets and in economies all over the world.

Develop, deploy, and maintain quality IT systems become infeasible in terms of efficacy and cost-effectiveness, choosing the right IT partner can be a game-changer. PiServe has proven expertise and experience to provide you the right IT solution for the right purpose at the right time.

For more information, please visit [www.PiServe.com](http://www.PiServe.com).

