

Report

# Shipa Cloud : Future with Kubernetes

---

**Made By:**

Atul Thakre 42

Nikhil Likhari 58

Yash Agrawal 85

**Subject : Cloud Computing [CST 452-4]**

**Teacher Assessment - 04**

**Semester: 7 Shift : I Computer Science & Engineering**



## Table of Contents

<b>Introduction to Kubernetes</b>	<b>2</b>
What are Kubernetes clusters?	2
Why Kubernetes ?	3
When to use Kubernetes ?	3
<b>Future of Kubernetes with Shipa Cloud</b>	<b>4</b>
<b>Shipa Cloud</b>	<b>4</b>
What is Shipa Cloud?	4
Need for Shipa.io	5
Advantages of Using Shipa Cloud	6
Future Scope of Shipa	6
Shipa Cloud - Simple and Faster than Kubernetes	7
Walkthrough Shipa - Application Dashboard for Kubernetes	8
Use cases of Shipa Cloud	14
Comparison of Shipa with its Competitors	15
<b>Conclusion</b>	<b>16</b>
<b>References</b>	<b>17</b>



## Introduction to Kubernetes

**Kubernetes** is a portable, extensible, open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation, letting us run distributed systems resiliently, with scaling and failover for our application.

In two words, it's a container orchestrator.

Kubernetes makes sure services are running smoothly, the way a software developer wants.

### What are Kubernetes clusters?


These are a set of nodes that run containerized applications.

Kubernetes clusters allow containers to run on multiple environments eg. physical, virtual, cloud-based, and on-premises. Kubernetes containers have an added benefit as they are not just restricted to a specific operating system, unlike virtual machines. Instead, they are able to share operating systems and run anywhere.

Kubernetes clusters mainly consist of two things:  
one master node and a number of worker nodes.

Nodes can be physical or Virtual machines. The master node is like an orchestrator, Its main function is to control the state of the cluster.

eg, which applications are running and their corresponding container images.



Master coordinates processes such as Scheduling, Scaling, Maintaining state, Implementing Updates.

### Why Kubernetes ?

- Helps develop, deploy and update software at scale
- Helps better manage your applications

### When to use Kubernetes ?

- When you are looking to transition to microservices architecture then kubernetes will be beneficial to use.
- When the development and deployment process is slow and you are not able to meet customer demands . You don't need to focus on manual capacity provisioning as kubernetes will do it for you.
- When you want to reduce your Infrastructure cost, as kubernetes will help you efficiently manage resources at container, pod and cluster level



## Future of Kubernetes with Shipa Cloud

### Shipa Cloud

**Developed By:** Shipa

**CEO of SHIPA:** Bruno Andrade

**Objective:** To enable organizations to devote more time and resources for application development and deployment

#### What is Shipa Cloud?

Shipa is an Application Management Framework which is designed to help developers and community to ease the deployment of Application. There are a lot of tools for deployment of applications. Now the question comes, Why is there a need for deployment on Third Party service if we can keep it in our local machine. Answer to this question is to remove the limitation of access to huge amounts of storage, processing power and bring cost efficiency for developers.

Many times managing the application with taking the underlying architecture and its management into consideration becomes hectic and time consuming for developers. Thus Shipa acts as an abstraction layer for the underlying infrastructure of Kubernetes and Cloud. Also, Kubernetes is hard to learn and apply for developers for a simple task of deployment.

Hence, Shipa's main objective is to ease out the task for developers and make them focus on coding the application and its deployment rather than wasting time and energy and worrying about understanding and managing Kubernetes architecture.



Focus is on application rather than containers, clusters, ships, etc.

### Need for Shipa.io

Kubernetes being an open source platform for container orchestration invites a large community of developers to deploy their application on it. With growth in this platform a pertinent problem arose. Once the application is deployed, post application management services are not provided by Kubernetes .

To one it might seem that the problem can be easily solved by developing analytics and management interfaces for each Kubernetes application. But this leads to the expense of precious development time in creating post deployment management tools rather than the main purpose of the application.

These development tasks can sometimes hamper mainstream application development. Also, it requires significant time to build this management system and requires robust testing for good output. Also with deployment of each new application , these analytic tools require technical support services which consume a lot of resources in terms of time, hardware and developers.

This problem leads to the requirement of having a structured solution to help the applications to scale out along with required technical support. Shipa Cloud offers a solution to this problem. Creating a Shipa account gives one access to a post deployment application management dashboard in just a few minutes.



## Advantages of Using Shipa Cloud

- Shipa is one platform that helps developers go no-think about infrastructure and concentrate only on the mainstream services of the application during development as well as technical support stage.
- Shipa is a stable cloud native space with support from experienced investors in the domain.
- Shipa allows developers to attach Shipa pools directly into the existing CI/CD pipeline to deploy a single object, as the pool is responsible for creating all Kubernetes objects that are required by the application.
- Efficient and user friendly Ship CLI tool comes with a lot of functionalities that allow developers to manage applications.

## Future Scope of Shipa

Shipa Cloud has and will continue to make application deployment and management easier for its clients. Shipa will make it possible to have almost no effort for underlying architecture and make application development as the only responsibility of the developers. Shipa will be able to achieve almost entire infrastructure layer abstraction and application placement processes. This will lead to a scenario where developers need not even think about servers. Shipa has made the fine line of distinction between different programming languages blur further.

Shipa provides support for both relational and non relational databases and is expected to expand its support to trending scalable databases. Shipa Cloud is continuously providing tighter integration between different collaboration tools like git, CI/CD systems and flexible features to create customizable workflows. Shipa aims to decentralize the process of

managing the clusters(used for running applications) from the developer to an automated efficient system.

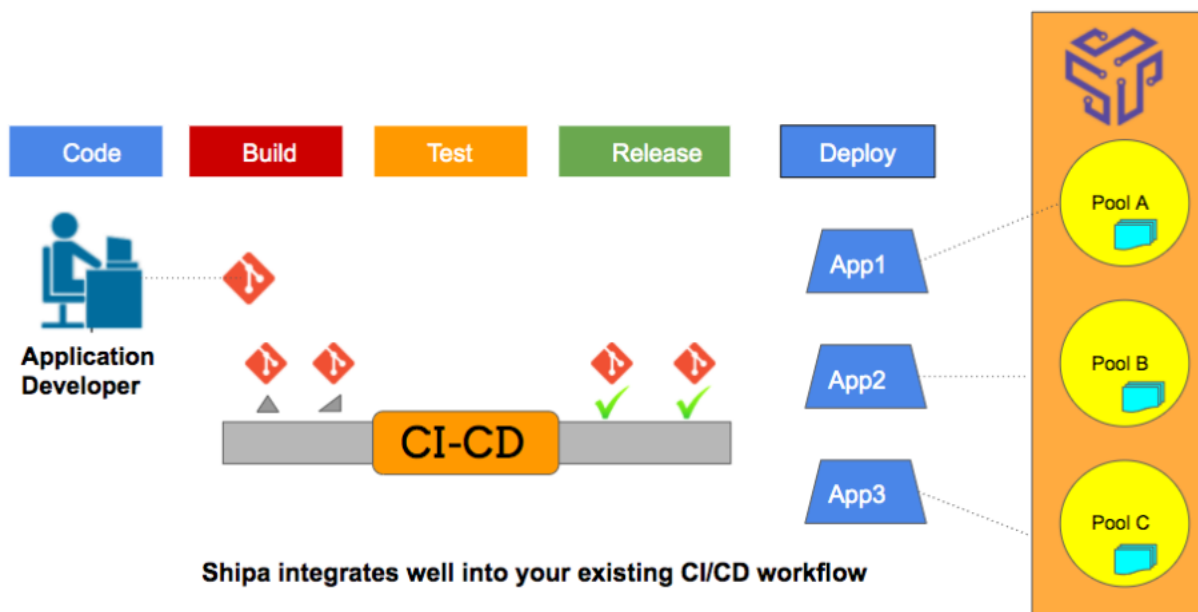
## Shipa Cloud - Simple and Faster than Kubernetes

The below information shows the difficulty faced by developers while using Kubernetes and how Shipa Cloud is a solution to it.


- **Difficulty in managing and visualizing application environment**

It takes a lot of time for developers to manage and scale up clusters using Kubernetes. Also, they have to look at cloud infrastructure for managing their applications which is a hectic task. Shipa eases out developer's life by providing visualization of these tasks, and leaving the task for developers to only concentrate on application building

- **Shipa Integrates well with the existing CI/CD workflow**







Shipa uses a concept of Pools where our CI/CD pipeline can deliver application code and then handed over Shipa for creating application objects, policies,etc.This eases the work for developers as it takes time for building such pipelines and managing in Kubernetes.

### Walkthrough Shipa - Application Dashboard for Kubernetes

It is a common practice when developers managing Kubernetes Clusters or if they are scaling Kubernetes application then some questions arise:

1. Is there a dashboard for Kubernetes?
2. How to see the status of the applications running on Kubernetes?
3. Are there any dependencies or objects dependent on my application?

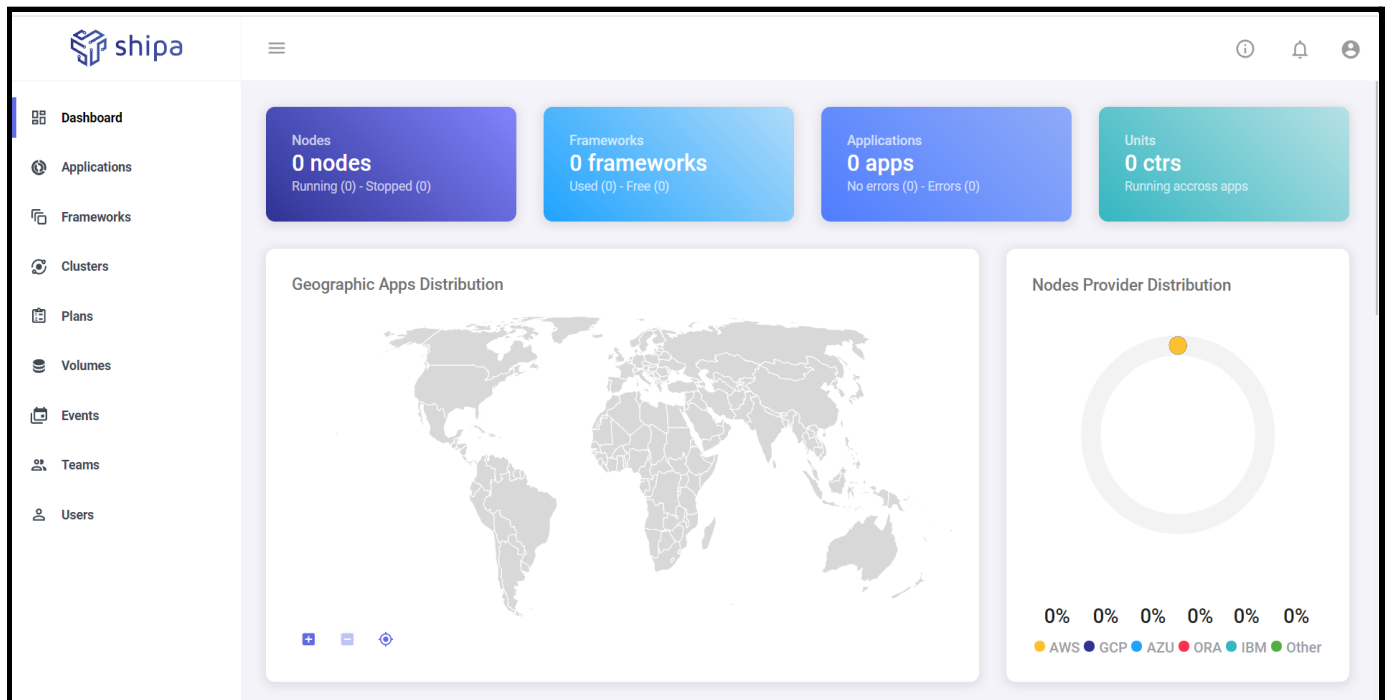
So there comes Shipa Application Dashboard which will answer all these questions. This is how one can start with using Shipa Application Dashboard for Kubernetes:

Things Needed:

- 1) A kubernetes cluster (Application to visualize)
- 2) Shipa Cloud Account

### Steps:

**Step 1:** Creating account on Shipa Cloud



## Step 2: Creating a Policy Framework

This is required for laying out ground rules for deployment of application building such as mentioning resource control, access and registry of applications,etc

## Create Framework



- ✓ Type selection
- General
- Resource Consumption
- Access Control
- Application Quota
- Registry Control
- Network Policy Setup
- Security Scans
- App Discovery

### Step 2 of 9 General Configuration

Please fill out all mandatory fields in this multi-step process and click on "Next" to continue to the following area of configuration.

* Name	* Provisioner
<input type="text" value="Starter"/>	<input type="text" value="kubernetes"/>
* Ingress	Kubernetes Namespace
<input type="text" value="traefik"/>	<input type="text" value="BasicApp"/>
<input type="checkbox"/> Make this my default framework <a href="#">i</a>	
CLUSTER CONNECTION	
Connect to cluster (Optional)	
<input type="text" value="Not connected"/>	

Cancel

Next

## Create Framework



- Type selection
- General
- Resource Consumption
- Access Control
- Application Quota
- Registry Control
- Network Policy Setup
- Security Scans
- App Discovery

### Step 1 of 9 Framework type selection

Select the level of configuration that you want to have when creating the framework

Configuration view [i](#)

☐ Basic

☒ Advanced

Cancel

Next

Create Framework

✓

Type selection

✓

General

Resource Consumption

Access Control

Application Quota

Registry Control

Network Policy Setup

Security Scans

App Discovery

Step 3 of 9

Resource Consumption

Control the resources that the applications within this frameworks will be able to use by selecting a plan with the desired limits

\* Plan

shipa-plan

PLAN DETAILS

This plan has been set as the "Default" plan

Memory  
Unlimited

Swap  
Disabled

CPU  
100

Public  
Yes

Cancel

Next

Create Framework

✓

Type selection

✓

General

✓

Resource Consumption

Access Control

Application Quota

Registry Control

Network Policy Setup

Security Scans

App Discovery

Step 4 of 9

Access Control

Mark the following checkbox if you want to enable the framework for all available teams within your organization (the option will be enable only if the selected plan is set as public). If the option is not selected, specifying the teams that can access the framework will be mandatory. Non-selected teams will be blacklisted.

☒ Make the framework public

Teams

Select team(s)

Cancel

Next

Create Framework

✓

Type selection

✓

General

✓

Resource Consumption

✓

Access Control

●

Application Quota

●

Registry Control

●

Network Policy Setup

●

Security Scans

●

App Discovery

Step 5 of 9

← Application Quota

Limit the amount of units that each application created within this framework can have, or leave it fully flexible by setting it as "Unlimited". This quota will be verified and enforced on every deployment.

App quota

Limited

\* App limit

6

Cancel

Next

Create Framework

✓

Type selection

✓

General

✓

Resource Consumption

✓

Access Control

✓

Application Quota

●

Registry Control

●

Network Policy Setup

●

Security Scans

●

App Discovery

Step 6 of 9

← Registry Control

Enter a list of image registries from which the applications on this framework will be able to deploy applications from. If no image is provided, apps will be able to deploy from any valid registry

Enable registries for app deployments

Enter one registry per line

Cancel

Next

Create Framework ✕

✓

Type selection

✓

General

✓

Resource Consumption

✓

Access Control

✓

Application Quota

✓

Registry Control

●

Network Policy Setup

●

Security Scans

●

App Discovery

Step 7 of 9

← Network Policy Setup

To enforce network policies in all the applications contained within this framework, check the box "Enforce Network Policy" presented below, and set up your own rules according to your preferences.

If allowing applications to overwrite the following network policy is required, please check the box "Allow app-level policies", so users with the right permissions can create their own rules accordingly.

☐ Enforce Network Policy

☐ Allow app-level policies

Create Framework ✕

✓

Type selection

✓

General

✓

Resource Consumption

✓

Access Control

✓

Application Quota

✓

Registry Control

✓

Network Policy Setup

●

Security Scans

●

App Discovery

Step 8 of 9

← Security Scans

☒ Disable app scans

Components to ignore on scans

Enter one component per line

CVEs to ignore on scans

Enter one CVE per line

Cancel

Next



## Use cases of Shipa Cloud

- **Application Operations (AppOps)**

On adoption of new technologies such as Kubernetes, the way to deploy and manage applications changes. This can slow down the application release .

This is where shipa comes to save the day, it provides a standard model to deploy and manage applications with the new as well as existing stack. This will help you adapt to new technologies faster while also syncing application consistency.

- **Policy As Code (Platform as a Code)**

If and when you wish to scale your applications, clusters, enforcing security policies of application, it becomes difficult to protect application and its environment from risks.

Here Shipa by its Policy as code enables platform operators to enforce security rules at application level by easily defining new policy Frameworks using standard tools such as Terraform, AgroCd etc.



## Comparison of Shipa with its Competitors

Helm (a competitor of Shipa Cloud) is a package manager for Kubernetes. It does not provide higher-level features such as shell interface & application dashboard, policy management and application logs. In comparison to this, Shipa provides an all round experience for infrastructure management, application deployment and policy enforcement.

PaaS services like Heroku and Openshift are platforms that help in deploying applications faster. However developers face many restrictions while working on these platforms. They do not provide flexibility to create custom load balancers or enforce certain network policies. Often the developers get restricted to use certain databases and services which might not be the best fit for their needs. Also, there are certain restrictions on available runtimes for languages. These platforms definitely provide consistent platforms for application building. Shipa has an edge over these services because it can provide features like a PaaS without constraints imposed by PaaS.

Building custom scripts and use case specific platforms for deployment and management of applications require a lot of resources and time. However these platforms are made for short lived purposes with a minimum set of features. This makes extension of these platforms for future use difficult. This limits the heights to which the application can be scaled. In contrast to this Shipa provides a structured service platform for policy enforcement and management tasks. Shipa saves valuable resources of an organization by reusing already available scalable microservices.





## Conclusion

Kubernetes is a open-source platform for managing containerized workloads and services. It is quite useful when developers require transition to microservice architectures. However, with increasing scale of development deployment and post deployment application management using kubernetes becomes a cumbersome process. It requires lot of manual development efforts to get some monitoring of the application. Managing the application with taking the underlying architecture becomes hectic and time consuming for developers. This is when Shipa comes to the rescue.

Shipa is an Application Management Framework which is designed to help developers to ease the Application deployment and management. It is one of the few platforms that helps developers concentrate only on the mainstream services of the application during development as well as technical support stage. Shipa has an edge over other contemporary PaaS as it can provide features like them along with application deployment and policy enforcement. Shipa will be able to achieve almost entire infrastructure layer abstraction and application placement processes in the years to come. This will make shipa an essential part of development on kubernetes.

## References

1. <https://learn.shipa.io/docs> (Official documentation of Shipa Cloud)
2. <https://dojowithrenan.medium.com/using-shipa-io-to-simplify-kubernetes-adoption-fcd2f49da5b1>
3. <https://collabnix.com/the-rise-of-shipa-a-continuous-operation-platform/> (The rise of Shipa)
4. <https://learn.shipa.io/docs/shipa-vs-other-software> (Ship and Its Competitors)
5. <https://www.capitalone.com/tech/software-engineering/create-and-deploy-kubernetes-clusters/>
6. <https://www.predicagroup.com/blog/why-kubernetes-2021/#:~:text=Kubernetes%20provides%20an%20easy%20way,resource%20management%20and%20provisioning%20tasks.>
7. <https://thenewstack.io/do-i-really-need-kubernetes/> (need of kubernetes)