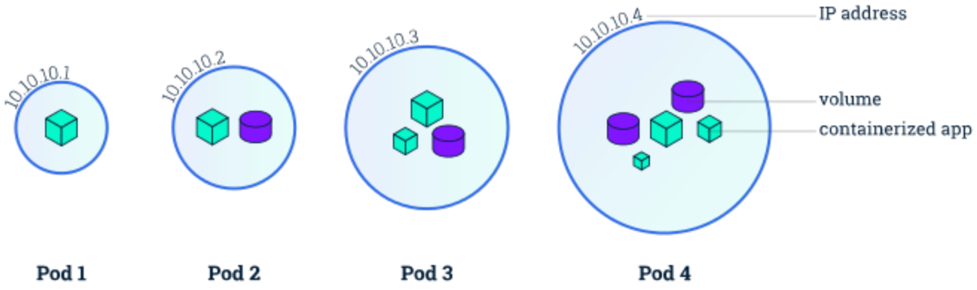
Working with Kubernetes

What is it?

Kubernetes is a portable, extensible, open-source platform for managing containerized workloads and services that operates at the container level developed by Google and later donated to the Cloud Native Computing Foundation.

"You can think of Kubernetes as you would a conductor for an orchestra. In the same way a conductor would say how many trumpets are needed, which ones play first trumpet, and how loud each should play, a container orchestrator would say how many web server front end containers are needed, what they serve, and how many resources are to be dedicated to each one."

- Dave Egts, **CTO** North America Public Sector, **Red Hat.**



At **Rootstack**, we work with the latest technologies and methodologies in the industry to deliver customized solutions that fit your business needs.

We can help you implement Kubernetes working with expert developers with +10 years of experience in their fields to ensure your technology infrastructure runs smoothly and with the best possible performance.

We've worked on +15 projects using Kubernetes. Hereunder read about our success story partnering with an important insurance company present all over Latin America.

Success Story: Improving user experience while optimizing costs

The Challenge

Our client had issues on their web portal and integration platforms regarding **load distribution.** Their server did not have the capacity to keep the project running. When the project faced traffic spikes, it became way too slow because their server was limited by hosting capacity.

For this reason, the client needed a **flexible and scalable platform**, based on containers, that would allow them to have available servers and hosting in the cloud and to adjust to the traffic demand of their company's web portal.

Our Solutions

Our senior software engineers at **Rootstack** recommended the client to implement a **Kubernetes** cluster as it was perfectly suited for the need they presented in their company: Is about a set of node machines for running containerized applications. The idea is that data response can be improved by separating the traffic load caused by data processing and database queries.

Thanks to this implementation, we were able to **optimize the workflow of the applications** associated with the web portal to accelerate development time, making it easier to deploy, scale and manage applications.

With this, then the company's web portal would not depend on a physical server but would have at its disposal several **virtual containers** that would be activated as the demand for resources grew, always showing optimal performance to users who entered the site. This technology allows for creating clones of the project that autoactivate depending on the software's demand. This means that the **hosting cost is dynamic.** The quantity will depend on the need, and hosting will be activated and deactivated automatically.

The advantage of Kubernetes is that it always privileges flexibility, being able to adapt perfectly to user requirements. For example, it has no limitations when it comes to working with different types of applications, since it supports a very diverse workload, be it of applications that process data or stateful/stateless applications.

In the deployment projects with Kubernetes, the software engineer can define that several scenarios are handled: if something fails in the deployment, the code is reverted and the original version that was before the issue is kept. Also, **containers have the ability to "self-heal"**: that is, the Kubernetes cluster can be configured in such a way that it can scale to itself and retrieve containers as needed according to its configuration.

This goes hand in hand with something called CI (Continuous Integration). The test environment and the production environment interact, if for some reason the deploy fails in the test environment, then the process is reverted and the system keeps the previous version.

Kubernetes has become one of the preferred platforms for companies due to its ease of use and the agility it brings to processes. And the key is that it is not just an "orchestration system" that organizes management. No, Kubernetes does not require centralized control as it is composed of a set of independent and combinable control processes, making the system easier to use, flexible and extensible.

Differences between traditional vs Kubernetes infrastructures

Traditional

- Running applications on physical servers
- There is no way to define limits for applications on a physical server, causes resource allocation problem
- Need to use applications on different physical servers, raises costs

Kubernetes

- Containers are lightweight, similar to virtual machines.
- They are portable, accessible through the cloud.
- They work the same way on the laptop as they do in the cloud.
- Containers are activated based on demand for resources.
- Continuous development, integration and deployment, with fast rollbacks in case of errors.

Why use Kubernetes?

These are some of the reasons every company should implement Kubernetes. Take advantage of these benefits:

- **Load balancing:** The work is divided between the different pods and the workload is balanced between them.

- **Extensibility:** Features and plug-ins can be added to Kubernetes, extending its functionality and scope, improving security and administration

- **Organize containers across multiple hosts:** Workflow can be carried out across multiple hosts, increasing agility, availability, and responsiveness.

- **Automated monitoring and rollbacks:** Automate and monitor application deployments and updates. If something goes wrong, Kubernetes reverses the action and stabilizes the platform.

- **Activate containers as needed:** Kubernetes automatically activates containers as there is increased demand for resources without sacrificing availability. Manage high workloads efficiently, controlling available resources.