

MEET

UP



AUTOMATIC DEPLOYMENT WITH KUBERNETES

FIXING PROBLEMS WITH CONTAINERS

Although one would think that automatic deployment of applications is a standard procedure in software companies, this actually still happens too often manually. Not only is this time-consuming, but it is also error-sensitive. In our recent meetup we discussed automatic deployment using Kubernetes.

Bi4 Group uses Kubernetes for automatic deployment of applications. One of the requirements for working with this tool is using Docker, a containerized infrastructure. App containerization is a popular virtualization method used to deploy and run distributed applications without launching an entire virtual machine (VM) for each app. A container is the lowest level of a micro-service which holds a running application, the libraries and its dependencies.

Kubernetes makes it possible to orchestrate and manage the deployment of containers, enabling more flexibility than virtualization technology. This results in a stronger focus on the service itself than the machine, as the containers become portable across different clouds and operating systems. Not only is there less room for errors when deploying containers through this platform, it also enables automatic scaling of containers and managing their health over time.

The platform uses its own terminology: containers grouped into units called "pods", along with shared storage/network, and a specification for how to run the containers. The machines that perform assigned tasks are called "nodes", that originate from a machine called a "master". The idea behind pods is that it's easier to schedule workloads and provide networking and storage services when containers are grouped together.



MEET UP

APPLYING CONTINUOUS INTEGRATION (CI)

Kubernetes can be used inside Gitlab, which gives access to advanced features. An example is Auto DevOps, which automatically detects, builds, tests, deploys, and monitors applications. During the meetup, several deployment activities were demonstrated: a simple application written in Python was built and tested. In real-time, it was possible to see how Gitlab handles the build and testing of code (or when someone commits changes to version control).

The first stage of this workflow

produces a build and push of a Docker image, the so-called 'recipe' of the Docker container. The test stage consists of a pull of this image and performance of a series of tests. If the tests go well, the code can be moved to the production environment. The last part of the workflow consists of placing the code package in the form of a Docker container on a server for development, so that the client can see its evolution on a day-to-day basis. The machine in this last step is created automatically upon deployment.

CONCLUSION

In this meetup we learned that Kubernetes is a very powerful tool for automatic deployment of applications. To reap its full benefits as an organization, you need to have a solid infrastructure and working methodology in place, using both Docker and a powerful Git repository manager.

