**Create an Openshift Origin 3.11 portable private cloud**

**Why create a private cloud**

The first question we could ask ourselves is: «Why have a private cloud?».

Well the answer is quite simple: «I do not wish to entrust my data and my applications to third parties». Indeed, before enjoying the advantages of a Cloud, you must think about the value and the confidentiality of the data and computer processing that you will deploy on this Cloud. I personally remain convinced that what has the most value on the internet is and always will be my data. And it is true that more and more people are worried about the value of the data they share and also of the algorithms that process that data. It then becomes obvious that you cannot entrust just anything to just anyone and that reading the GCUs of different cloud providers is very important. Experience what a Cloud is, understand that it is not magic, certainly helps to use it better.

At the bottom of this article, you will find a link to my github where I share a simple recipe for building your own cloud, so you can experiment. You can choose to simply read the recipe and see the photos of the result or build your own.

**Let’s talk about data now**

To convince you, I have placed in the next section some links to very current topics about data and what can be done with it.

Some information about data

Many governments, companies and individuals are aware today and take action to know and regulate the use of the data they entrust. To convince you, I can quote:

* The Cloud Act which in 2018 raised awareness, because this law notably allows American courts to solicit from operators the personal communications of an individual without the latter being informed, nor his country of residence, nor that the country where those data are stored. [CLOUD Act](https://www.congress.gov/bill/115th-congress/senate-bill/2383/text)
* The GDPR (General Data Protection Regulation) is a regulation of the European Union which strengthens and unifies data protection for individuals within the European Union. [GDPR](https://afcdp.net/reglement-europeen-rgpd-indexe-commente/)
* An example of the Google Drive GCU, which explains very clearly that Google can use your data to improve its services. [Google Drive GCU](https://support.google.com/drive/answer/2450387?hl=en)

**OpenShift presentation**

[OpenShift](https://github.com/openshift/origin) is a PaaS (Platform-as-a-Service) solution. In general, companies use it to build, deploy and run applications in containers. It is available in open source under the Apache 2.0 license, it is available in two products: Origin (community) or Enterprise.

Openshift is clearly carried by [Red Hat](https://www.redhat.com/) and the main contributors to this project mostly work at Red Hat.

Since version 3 and the big redesign, Openshift relies completely on [Docker](https://www.docker.com/) and [Kubernetes](https://kubernetes.io/) from [Google](https://www.google.com/). The life of this product will therefore depend on these two backbones.

**Why use Openshift**

If I had to retain a main reason for using Openshift, well I would say that it’s much easier for a Cloud team to benefit from the Docker encapsulation work and especially Kubernetes which is carried out by the Red Hat teams. This makes it possible to concentrate on the services to be provided to end customers in companies such as trades and projects.

Creating and running a private cloud in production requires a lot of technical skills and time, using Openshift has a cost, of course, and this allows you to have support and a stabilized base by the Red Hat teams.

**What can be done with Openshift**

Typically, build and deploy applications automatically. There are several ways to do this, the main ones are :

* Docker File mode, which allows you to automatically build a Docker container by providing OpenShift the address of a source code manager pointing to a Docker File and its dependencies. We use it a lot to migrate projects from old virtual environments to our Cloud.
* Source To Image mode which allows you to automatically build an application by pushing the source code application in OpenShift. I use it to make starter-kits for trades, it indicates a Git repository where is the code that a startup has built with them, their credentials and presto the application starts.
* Custom Build mode which allows you to provide your own logic for building an application by providing OpenShift a Docker image designed for this purpose. It can be used for applications which have an original stack compared to most of the usual applications deployed on a Cloud. This allows for tailor-made.

One of the very interesting advantages of OpenShift is that it allows you to define an automated deployment strategy of an application when a new image version is published in the registry or when the configuration of the application is updated. Typically it is taking advantage of the best advantages of DevSecOps for projects or trades.

In addition to these build and deployment modes, OpenShift offers the possibility of defining its own application “blue prints” in the form of “template” files in Json or Yaml format. These «blue prints» describe both the topology of the application architecture and the container deployment policy. This makes it possible to industrialize application deployments for a company that has already worked on the standardization of its development stacks.

**A bit of openshift vocabulary**

Some concepts that are important to understand before starting:

* POD: it is a Docker container execution environment local to a server,
* Service: it is an entry point (VIP) making abstract «load-balanced» access to a group of identical containers. In principle, we deploy a Service by third of the architecture,
* Deployment Config: this is an object that describes a deployment policy for a container based on triggers (for example: redeploy when a new version of an image is available in the Docker registry),
* Replication Controller: it is a technical component in charge of POD resilience,
* Route: A route exposes an entry point (DNS hostname or VIP) outside of an application.

**How Openshift works**

Quickly, the node or nodes which control the others serve:

* Process requests to the administration API,
* Carry out image build operations and container deployment,
* Ensure the resilience (replication) of the PODs where your applications operate,
* It or they use a distributed etcd directory for configuration sharing and service discovery.

The managed nodes host the PODs and run containers (application and / or Registry). It is possible to interact with the platform through its REST API, CLI or via its Web portal.

**How I built my private cloud**

I started off with [Raspberry Pi](https://www.raspberrypi.org/) type boards, they are some of the most affordable boards and are very convenient to use. I was quickly blocked because I did not manage to build Openshift Origin for their ARM architecture. Some research later I discovered [LattePanda](https://www.lattepanda.com/) cards which have the advantage of not being much bulkier and being built on an Intel architecture.

The first thing to do to build a Cloud was then to collect the necessary elements to add to the LattePanda cards. These elements are very simple and fairly well known. You need :

* A network router so that all the devices in your cloud talk to each other,
* A network swith to link them together,
* Hard disks to store information,
* And electricity.

And yes, in its simplest form a cloud is that.

Une image contenant équipement électronique, table, ordinateur, assis

Description générée automatiquementA Cloud

You can see in this photo the final result of the assembly of these elements together. The advantage of these components is that they are really much smaller than the ones I use in my work, so you can set up a private experimental cloud in your home, without turning it into a server room.

Une image contenant équipement électronique, circuit

Description générée automatiquement

LattePanda cards are really powerful and easy to use. They are intended for use with [Windows](https://www.microsoft.com/) or [Ubuntu](https://ubuntu.com/). I use [Centos](https://www.centos.org/) a lot for my personal projects, because I work very often on [RHELs](https://access.redhat.com/products/red-hat-enterprise-linux) and I tried to install this OS on these cards. I only had a small problem with the SD card reading slot, I found the solution in the manufacturer’s forum and I thank this [community](https://www.lattepanda.com/forum/)which is very active and helps each other. I placed the solution to get around this little problem in my recipe on Github, don’t worry.

**Why use a cloud**

I’m not going to give you a long list of the benefits of a cloud, it’s not my job. Personally, what I appreciate the most is being able to share these resources between different projects and users. If you are really into a DevSecOps process you can use part of your cloud for a project and as soon as you no longer need the resources to free them up and therefore no longer pay for their use. This is for me the main advantage of this type of computer architecture. The second advantage is scalability, which allows you, if your project is successful, to immediately obtain the resources necessary to maintain the availability of your project for all your new customers. And that is really great. Some solutions put in place react immediately to the arrival of new customers and add resources without any of them realizing an impact on the service. Once activity decreases, resources are freed up and adapt to the new flow of users.

**What is this cloud for me?**

I want to regularly test tools and new ways of doing my DevSecOps activity. I have a fairly powerful laptop, on which I installed my software factory tools ([GitLab](https://gitlab.com/explore), [Jenkins](https://www.jenkins.io/), [Sonarqube](https://www.sonarqube.org/), …). I have project sources that allow me to test the new tools that I add to this software factory. Some sources have security vulnerabilities, others have quality flaws, and plenty of other markers that should be detected by the tools I’m testing. My laptop is limited in power like anything in life. I really wanted to build an experimentation cloud that would allow me to have more computing power and a target infrastructure to test some of my ideas. I develop some security tools with my colleagues, to improve our software production chain, but the majority of our work consists in using products and setting them up and associating them as best as possible to obtain the best results for us.

Finally, I also use this portable cloud to do demonstrations around me, it’s really nice to help people understand what this rather abstract notion of cloud is. This portable assembly and the Openshift Origin console help me a lot in showing others what a cloud is, how it is built, how and why it can be used.

Here is the link to the recipe that I want to share with you: <https://github.com/AntoineMeheut/Rokc-Mk3>

I can’t resist the urge to give you a last photo of the Openshift Origin console.

Une image contenant moniteur, intérieur, ordinateur, portable

Description générée automatiquementOpenshift Origin 3.11 console

I don’t use it often, because I prefer the CLI and it is also very useful for demonstrating and helping to understand this which takes place on a cloud, when you submit a project, it is also a fairly nice popularization tool.