SERVERLESS COMPUTING AND CONTAINERIZATION IN CHALLENGER BANKS SOFTWARE

Kolomytsev Artem, student of 657P1 group
National Aerospace University named after N. Ye. Zhukovsky "KHAI"

Challenger banks are small, recently-created retail banks that compete with the longer-established banks in the country. These banks distinguish themselves from the historic banks by modern information technology practices, such as online-only operations, that avoid the costs and complexities of traditional banking [1].

Challenger banks can be established firms – most likely midsize or specialist – that seek to compete with larger institutions. These banks don’t carry the weight of legacy technology, so they can leapfrog over traditional infrastructure and disrupt the status quo. They are more flexible, quicker to adapt to user needs, more user friendly and more personal than traditional banks. Their biggest advantage is that they have started fresh with a digital offering and the use of the latest technology available [2].

Because challenger banks’ backends often is created from scratch – they can use lot’s of modern cloud technologies for faster time-to-market in new features development, increases scalability and reliability. Two most important and growing trends in cloud application’s development in 2018 / 2019 are serverless and containerization [3, 4].

"Serverless" is a type of applications where server-side logic is still written by the application developer, but, unlike traditional architectures, it’s run in stateless compute containers that are event-triggered, ephemeral (may only last for one invocation), and fully managed by a third party. One way to think of this is "Functions as a Service" or "FaaS". AWS Lambda is one of the most popular implementations of a Functions-as-a-Service platform at present, but there are many others, too [5].

Deployment is very different from traditional systems since we have no server applications to run ourselves. In a FaaS environment we upload the code for our function to the FaaS provider, and the provider does everything else necessary for provisioning resources, instantiating VMs, managing processes, etc.

Horizontal scaling is completely automatic, elastic, and managed by the provider. If your system needs to be processing 100 requests in parallel the provider will handle that without any extra configuration on your part. The "compute containers" executing your functions are ephemeral, with the FaaS provider creating and destroying them purely driven by runtime need. Most importantly, with FaaS the vendor handles all underlying resource provisioning and allocation – no cluster or VM management is required by the user at all.

Functions in FaaS are typically triggered by event types defined by the provider. With Amazon AWS such stimuli include S3 (file / object) updates, time (scheduled tasks), and messages added to a message bus (e.g., Kinesis).

Second important trend in cloud software development is containerization. Monzo, on of the biggest challenger banks use them a lot [6].

Containers are a form of operating system virtualization. A single container might be used to run anything from a small microservice or software process to a larger application. Inside a container are all the necessary executables, binary code, libraries, and configuration files.

Compared to server or machine virtualization approaches, however, containers do not contain operating system images. This makes them more lightweight and portable, with significantly less overhead. In larger application deployments, multiple containers may be deployed as one or more container clusters. Such clusters might be managed by a container orchestrator such as Kubernetes.

In contrast with VM, each container shares the same host OS or system kernel and is much lighter in size, often only megabytes (figure 1). This often means a container might take just seconds to start (versus the gigabytes and minutes required for a typical VM).

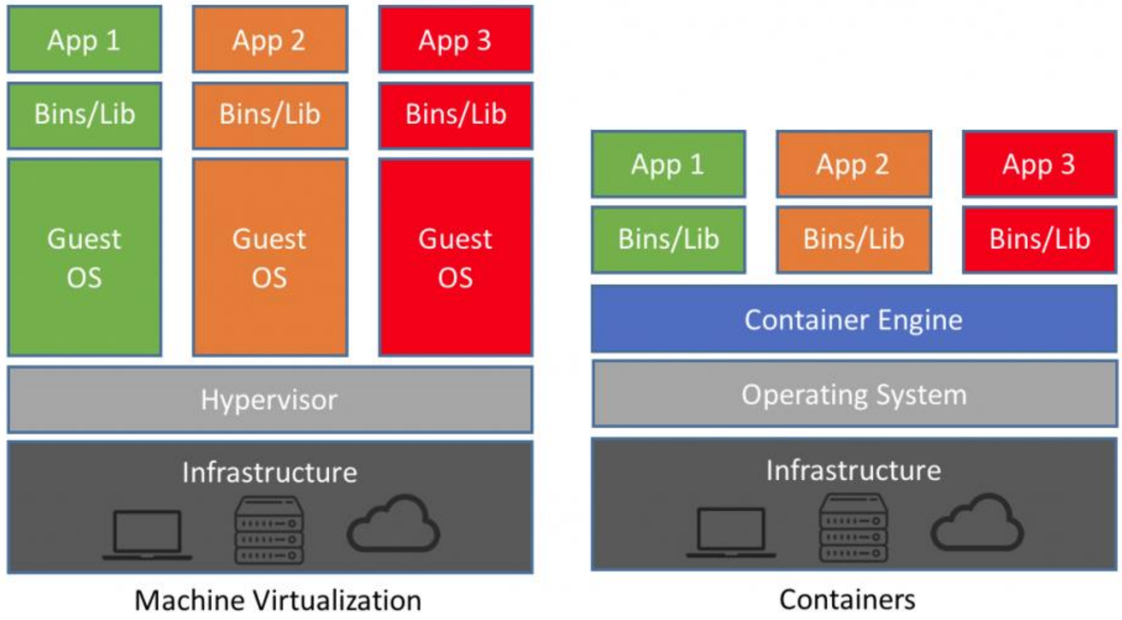

Figure 1. Container’s vs VM’s layers [7]

The fintech revolution has been reshaping the landscape of the traditional financial industry by cutting costs and improving the quality of financial services, using big data to assess credit risks in a more intelligent way, and creating a more diverse and stable credit sector.

Two most important and growing trends in cloud application’s development in 2018 / 2019 are serverless and containerization.

Using cutting edge technologies challenger banks can work more effective and deliver features faster.

References

1. The rise of challenger banks [Electronic resource] – Available at https://home.kpmg.com/xx/en/home/insights/2018/02/rise-of-challenger-banks-fs.html – 2 Mar 2019.

1. Monzo, Revolut and other challenger banks are shaking up the industry [Electronic resource] – Available at https://theconversation.com/monzo-revolut-and-other-challenger-banks-are-shaking-up-the-industry-99564 – 2 Mar 2019.

2. Serverless computing [Electronic resource] – Available at https://blog.g2crowd.com/blog/trends/digital-platforms/2018-dp/serverless-computing/ – 2 Mar 2019.

3. Cloud trends for 2018 [Electronic resource] – Available at https://www.doherty.co.uk/blog/cloud-trends-2018 – 2 Mar 2019.

4. AWS Lambda features [Electronic resource] – Available at https://aws.amazon.com/lambda/features/?nc1=h_ls – 2 Mar 2019.

5. Building a Modern Bank Backend [Electronic resource] – Available at https://monzo.com/blog/2016/09/19/building-a-modern-bank-backend/ – 2 Mar 2019.

6. Containers vs. Virtual Machines (VMs): What’s the Difference? [Electronic resource] – Available at https://blog.netapp.com/blogs/containers-vs-vms/ – 2 Mar 2019.

**Scientific Advisor –*
Kuznetsova Yu. A., Ph.D., Associate Professor of 603 Department.