Embrace microservices and platform services

TO DEPLOY ITEXT SOFTWARE EFFICIENTLY

WHITEPAPER

Microservices Architecture, Cloud Infrastructure and Platform Services are popular approaches in IT. iText is embracing these trends and we're helping our customers to use them to deploy our software efficiently.

iText's PDF software has long been an essential part of many document management workflows. PDF is the dominant digital document format due to its flexibility, portability, and openness, and has seen widespread adoption across diverse domains. The iText PDF library enables the generation and manipulation of PDF documents on the back-end. So even if you've not personally used our software, you've almost certainly experienced iText-generated PDFs, such as bank statements or airline boarding passes.

From monolithic architecture to microservices

For years, developers have been using a monolithic architecture for their applications: they would deploy an application on a single server, on-premise or in the cloud. Of course, for scalability reasons, a single server could become multiple servers with a load balancer in front of it. However, fundamentally the idea is still the same: an application is deployed as a single, indivisible unit.

A monolithic architecture has its strengths (it's simple to develop, deploy and test), but you can't scale parts of the application independently, only the whole application. And if you have to change part of the application, you have to change the whole application. The tight coupling of its parts makes every change to the application a challenge.

Another approach that has become popular in recent years is a microservices architecture. This breaks an application down into a collection of smaller (micro) independent units, each of them executing a specific task. Each of these units runs as a separate service. There are various mechanisms to communicate between microservices: synchronous (REST, GraphQL, gRPC) or asynchronous (MQTT, STOMP, AMQP).

While a microservices architecture adds complexity, it has important strengths. Its smaller components are easier to understand and hence easier to test individually. Moreover, each of these components can be deployed and updated independently. If there's a bug in one component, it doesn't impact the entire application. And fixing bugs and adding features to one component is easier and has fewer risks than doing the same to a monolithic application. In addition, each component can be scaled independently.

In a Forrester study from 2019, 76% per cent of IT decision-makers claimed they wanted to prioritize rearchitecting applications for microservices over the next 12 months. And in O'Reilly's Microservices Adoption in 2020 report, 77% of the respondents stated they had adopted microservices. Furthermore, 29% were looking to migrate the majority of their systems to microservices. Of those adopters, 92% reported a high-level success in their migration to microservices. As you can see from these numbers, the microservices architecture is a popular choice.

Docker and Kubernetes

If we talk about microservices, this is about a logical separation between the different services. But when it comes to implementing this architecture, you'll have to make some choices. Containers are a logical choice to implement microservices. You isolate each microservice in its container, and all containers share the host operating system kernel. These containers are attached to the same virtual network and can communicate this way with each other. The industry's de facto container runtime is Docker Engine, so if you implement your microservices as Docker images, you can make use of a big and powerful ecosystem.

Containers are the workhorse for microservices, but for complex systems you also need automated deployment, scaling and other management tasks. That's where Kubernetes (K8s) comes in, a container orchestration system for automating deployment and managing your containerized applications across multiple hosts.

The basic unit in Kubernetes is a pod. It consists of one or more containers that share the same resources, that can see each other and that are located on the same host. Kubernetes is the perfect fit for a microservices architecture. You deploy a pod for each microservice, consisting of a container for the microservice itself and one or more containers for supporting services such as a database.

Both Docker and Kubernetes are especially close to our hearts at iText, since they too are part of the open-source community and have seen widespread industry adoption.

Platform services

Another shift that has been happening in the market is the migration to Cloud Infrastructure and Platform Services (CIPS). Gartner defines CIPS as "standardized, highly automated offerings, in which infrastructure resources (e.g., compute, networking and storage) are complemented by integrated platform services. These include managed application, database and functions as-a-service offerings."

A lot of companies are migrating to those platforms because they don't have to maintain the equipment and guarantee network availability and Quality of Service (QoS) themselves. You can use the applications and other services manually with a web-based user interface and programmatically with a REST API. The equipment and network management are no longer your concern. Another big advantage of Cloud Infrastructure and Platform Services is scalability: all resources of a CIPS offering are scalable and elastic in near-real-time and metered by use. This means that you only pay for the value you get out of it.

CIPS includes both IaaS (Infrastructure-as-a-Service) and PaaS (Platform-as-a-Service) offerings. In July 2021, the leaders in Gartner's Magic Quadrant for Cloud Infrastructure and Platform Services are Amazon Web Services (AWS), Microsoft Azure and Google Cloud. In a recent report on the future of the public cloud, Gartner predicts that CIPS providers will shift even more to automated programmable infrastructure. This will reduce the operational burden even more.

A popular form of Cloud Infrastructure and Platform Services are cloud-based offerings to run Docker or Kubernetes. For instance, Amazon Elastic Container Service (Amazon ECS) is a managed container orchestration service that helps you easily deploy, manage, and scale containers. In the same vein, Amazon Elastic Kubernetes Service (Amazon EKS) is a managed container service to run and scale Kubernetes applications. Both cloud services save you from the hassle of setting up and managing a container orchestration system.

iText products on Amazon Web Services

We've been talking to companies and many of them are migrating to Cloud Infrastructure and Platform Services. In these conversations, we've realized that they're using our products in AWS. Instead of having them wrap our products in containers and deploy them to AWS, we decided to make this easier for them.

That's why we've published iText DITO on the AWS Marketplace. You can run it on Amazon ECS or Amazon EKS.

iText DITO is our data-driven, template-based collaborative PDF generator that reduces the time and cost of document automation projects. It allows you to mass-generate PDFs by passing JSON data to a REST SDK/API to combine with your templates designed in the iText DITO Editor. The product also comes with a workspace control room experience, called iText DITO Manager. This enables you to manage your users, security roles, and SDK instances. It contains a central repository where users can keep their templates, data collections and other resources. Business users can manage the assets such as logos, fonts, visuals and any other asset for a PDF template easily themselves in the tool.

By running the container we published in Amazon ECS or Amazon EKS, you don't need to install and operate your own container orchestration software. This way you can efficiently deploy iText DITO as a microservice. The other components of your application running in AWS talk to iText DITO using its REST API.

Our customers are embracing the benefits of Cloud Infrastructure and Platform Services, and so are we.

We deliberately chose the approach of offering a container on a platform service, instead of offering iText DITO as Software-as-a-Service (SaaS) or as an AMI (Amazon Machine Image) to run on Amazon Elastic Compute Cloud (EC2). Let's have a look at the differences between these approaches and why containers are the best architectural fit for microservices.

Before we move on, it's important to note that the well-known iText PDF library is also available on the AWS Marketplace. In order to grant as much flexibility as possible to our users, iText 7 Suite functionality can now be deployed in a BYOL (Bring Your Own License) model on AWS, allowing you to utilize the iText 7 Core PDF library for manipulating and processing PDFs, plus additional functionality provided by iText 7 add-ons. All using a scalable and cost-effective RESTful API.

Containers vs SaaS

Running your applications in containers is fundamentally different than using them as part of a SaaS offering:

- **Data sharing**

  With SaaS, you use a service from a service provider. This means the service is deployed on the service provider's infrastructure, including your data. You can't choose where your data is residing. If you're not comfortable with sharing your data with that provider, SaaS isn't a good choice. With a container-based approach, you can run a container where you want. For instance, on Amazon ECS or Amazon EKS you can choose the region of the data center where the container is running. And you can also run the container on your own private cloud.

- **Multi-tenancy**

  A SaaS solution services multiple customers in an efficient way. All data access happens with the customer's ID so you don't see other customer data. However, there's no real isolation between customer data: all customer data are in the same database. So, when someone breaks into a SaaS solution, all data are compromised. With a container-based approach, this doesn't happen. A related aspect is performance isolation. With a SaaS solution, you risk that the workload of another customer impacts the performance of your workload. If you're running your application in a container, you have more control over its performance and especially its predictability.

- **Maintenance**

  An advantage of SaaS is that the service provider has the responsibility to maintain the application, so you can focus on using it. If the service goes down, you have someone to hold accountable for it. The service provider is also responsible for upgrades. With a container-based approach, you deploy the container and you are responsible for its maintenance. However, both Amazon ECS and Amazon EKS have tools to help you with this task, such as monitoring. Moreover, upgrading a container is as easy as pulling a new version and recreating the container.

  All things considered, for a microservices architecture containers are better than SaaS. They offer you more flexibility and control, with the only drawback that you are responsible for upgrades yourself.

Containers vs AMIs

Running your application in containers instead of virtual machines on Amazon EC2 has also some differences:

Scalability

A container in Amazon ECS or Amazon EKS is easier to scale than an Amazon Machine Image on EC2. Both ECS and EKS have native scaling mechanisms. So, it's easy to scale up and down your containers when you need more or less performance.

Image size

Because an AMI is an image of a full virtual machine, it's always bigger and includes a lot of stuff that you don't need. A container image has just what you need and nothing more, it's tuned for a specific application.

All in all, this makes containers a much better fit for a microservices architecture than AMIs.

Conclusion

There are various important market shifts at play for swift document generation and management, with the Microservices Architecture on the one hand and Cloud Infrastructure and Platform Services on the other hand. iText is embracing these trends and we're helping our customers to use them for their deployments.

Our experts are at hand in case you want to learn more about generating PDF documents on AWS. We can also help you if you want more control over the process. Contact us if you want more information.

ABOUT ITEXT SOFTWARE

iText Software is a leading technology company in the digital documents space. The company's flagship product is an open source software library to create and manipulate PDF documents in Java and .NET (C#).

There are currently millions of iText users, both open source and commercial. iText's customers (software developers, technology vendors, software integrators) span the fields of technology, financial, public, government and health care sectors including many of the Fortune 500 companies.

iText has dedicated international teams with offices in Belgium (Ghent), Singapore and in the USA (Boston).

https://www.itextpdf.com