SaaS, PaaS, IaaS? A taxonomy of cloud service models

Eleonora Borzi
borzi@kth.se
Andreas Kärrby
karrby@kth.se

April 2021

1 Introduction

One frequently used technology in DevOps is cloud services. These services can be used to facilitate testing and deploying software, as they offer ways of automating processes such as building and testing [1]. Given the predominant use of the cloud in DevOps, it's important to understand what the cloud is and its uses. The cloud we are all familiar with now began to take shape in 2002 when Amazon Web Services (AWS) released the first public cloud [2]. Since then, many other corporations have released their own cloud services, for example, Azure and DropBox. There are many different cloud services but all of them can be categorized into one of the following three categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The main difference between them lies in how much control the user has on the infrastructure [3]. This means that a client can choose the right service depending on his or her needs. However, it can be confusing to discern if a service is a PaaS, SaaS or IaaS. The goal of this essay is to introduce the cloud, to describe and to compare the three cloud service models, IaaS, PaaS and SaaS, and popular tools used in each of them. The aim is to clarify any misunderstanding on this topic and give a clear picture for beginners.

2 Background

In 2002, Amazon Web Services introduced the first public cloud [2]. Before then, only IT specialists had the expertise needed to use services that gave access to an organization's computational resources. This process was called grid computing [2]. The new cloud computing made it easier for small companies and individual entities to use resources that would fit their needs. This eliminated the risk of buying expensive servers that would only be used for some testing. As technology advanced, AWS was able to launch the Elastic Compute Cloud

(EC2) in 2006, which was open to the public [2]. A few years later, a database cloud service was released, Dropbox. Microsoft made its entrance in this field by launching Azure [2]. After 2012, cloud services became more advanced and could even process live streaming data. DevOps entered the game, microservices were used to develop applications on the cloud, and the first container services on the cloud were launched [2].

2.1 What is cloud?

According to the definition of cloud, as defined by the National Institute of Standards and Technology (NIST) [3], a cloud service should have 5 key-features.

- **On-demand Self-service:** *Self-service* means that a user should be able to access a service without any external help and the process should be automated. The *on-demand* part means that the service requested should always be available, and no external party should have to set up the service before it is available for use.
- **Broad Network Access:** Even with a basic network connection, a user should be able to access a service.
- **Resource Pooling:** When a resource is not used by a client, it should be available for use by another customer; this is usually solved by virtualizing user sessions.
- **Rapid Elasticity:** The cloud services should be able to expand as clients require it and also scale down when they don't. This should be automated so that when the clients use the max amount of resources given, the trigger should allocate more resources.
- **Measure Service:** The service should be quantified so the client knows the price.

The three cloud service models that will be described in this essay are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The main difference between them lies in how much control the user has on the interface. An intuitive way to show the difference is through a triangle shown in Figure 1. The control over the infrastructure becomes less at the top of the triangle and more at the bottom. SaaS is targeted at the end-user, PaaS for software developers, and IaaS for IT administration [3].

3 IaaS

In the IaaS service model, the provider is responsible for handling the physical hardware, i.e. infrastructure, needed for software development work, such as servers, storage, and network [5]. Meanwhile, it is up to the customers to select

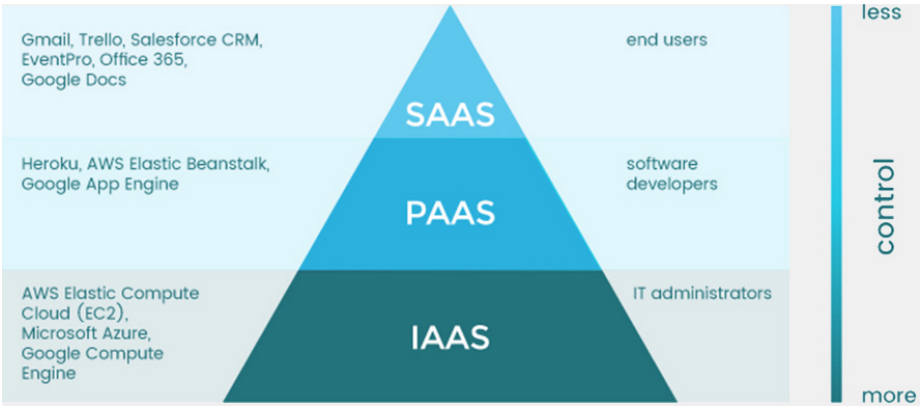

Figure 1: Triangle representation of Cloud Services [3]

what to install on top of the infrastructure, in terms of operating systems, middleware (.NET Runtime, JDK, etc.), and other software based on their needs. As stated earlier, it is thus the cloud service model that grants the greatest amount of flexibility for customization. The key strength of IaaS is that companies no longer have to incur up-front costs to acquire hardware, and importantly they also do not have to worry about over– or undershooting demand, something which could be very costly if the companies misjudge what their future needs will look like. With IaaS, these complications are abstracted away, which means companies can focus on more important matters. Most cloud service providers have IaaS services; Microsoft has Azure IaaS [6], Amazon offers EC2 [7], and Google has Google Compute Engine [8].

Furthermore, companies can use the concept of *Infrastructure-as-Code (IaC)* when specifying the infrastructure that they want. As opposed to interactively picking and configuring the desired infrastructure in e.g. a web interface, IaC provides a way of defining infrastructure using declarative code. Among other things, this approach enables version control of infrastructure and prevents something known as "configuration drift", where changes are made to environments without documenting it and without making sure changes are applied everywhere [9]. It also facilitates disaster recovery, since the declarative scripts are idempotent, meaning they produce the same result no matter how many times they are run. One drawback of IaaS is the cost, which can rise quickly since customers are usually billed per resource used. It's also important for the customer to know exactly what they have to provide: antivirus is one thing that might be overlooked [5].

Popular tools that make use of IaC are Terraform by HashiCorp, AWS CloudFormation by Amazon, and also Ansible, Puppet, Chef, and SaltStack [10].

4 PaaS

A PaaS gives the client the freedom to make any changes in the application but the customization of the development platform is limited. Only a few preset changes can be made. Moreover, the analytics of the application using the cloud service is available to the client. This includes information such as what component is being used or viewed the most and gives an insight into what should be changed in the application. The provider is responsible for storing the data but the client has access to it, this could be a problem if a large amount of data is transferred which lowers the performance. Unlike in IaaS, the client and the provider have the same amount of responsibility for the infrastructure. The provider is responsible for providing an up-to-date and functioning OS. Everything else is under the responsibility of the client, such as setting up the implementation, application monitoring, and data.

PaaS is often used when companies want a public cloud model[1] but cannot find a suitable one. Then, PaaS can be used to have the infrastructure and platforms on an external provider instead of having them on internal data centres. However, a security risk is that the provider has access to the data and the application since the OS is their responsibility [11], this could raise GDPR issues. Some examples of PaaS providers are Windows Azure, AWS Elastic Beanstalk, and Google App Engine.

5 SaaS

Historically, the SaaS cloud service model can be said to date back as far as the 1960s, when organisations with access to large mainframe computers offered a share of their computing capacity to large enterprises such as banks [12]. These days, SaaS refers to services where the software itself is the artefact of interest, and it is probably the kind of service model that most laypeople associate with the cloud. Most digital services used by individuals nowadays can be categorized as SaaS services. Some examples are Salesforce, a CRM service for managing customer relations; Slack, a messaging platform; and Google's suite of services such as Gmail and Google Drive.

As mentioned in the background, the SaaS cloud service model is generally targeted at end-users. Unlike PaaS and IaaS, where some control over the underlying architecture is afforded to the customer, in the SaaS model the only control the customer has is the ability to somewhat customize the use of the application [5], and even this ability may be limited. This brings both benefits and disadvantages. For the provider, it can be easier to manage the service exactly as envisioned, and updates can be applied without requiring user action. Conversely, this may be bad for the customer if, for instance, they are commit-

[1] In a public cloud model, anyone can access the service, and the provider is responsible for managing the systems [4]

ted to the way things used to work and are forced to spend time and money on re-learning how to use the service [5]. Another problem with SaaS services is that they tend to be hosted off-site and are accessed through the internet, which means that for applications where low latency is required, these services may be unsuitable [5].

6 Conclusion

Nowadays, cloud services are used for many applications and also in DevOps. Without these services, companies would need to spend more time on extraneous tasks just to get things up and running. The services provide a way to abstract away these details and instead allows companies to focus on making their business visions a reality. There are many providers and services which can be confusing. This is why having a clear picture of the three models, IaaS, PaaS and SaaS, helps to pick the right provider. IaaS offers the infrastructure and gives to the client the freedom to entirely customize it. The drawback is the cost. Meanwhile, PaaS offers the infrastructure as well as a development platform, but the client has limited customization choices. The providers have access to the database which can be a con for some companies. Finally, SaaS offers the whole spectrum where the customer can only customize the use of the application. This service cannot guarantee low latency and the service can update which can defer in the client's workflow.

7 References

[1] A. Suleman, "Council Post: The Relationship Between The Cloud And DevOps," Forbes. https://www.forbes.com/sites/forbestechcouncil/2017/07/21/the-relationship-between-the-cloud-and-devops/ (accessed Apr. 11, 2021).

[2] "History of the cloud — BCS." https://www.bcs.org/content-hub/history-of-the-cloud/ (accessed Apr. 11, 2021).

[3] "IaaS vs PaaS vs SaaS – cloud service models compared," Plesk, Oct. 24, 2019. https://www.plesk.com/blog/various/iaas-vs-paas-vs-saas-various-cloud-service-models-compared/ (accessed Apr. 10, 2021).

[4] O. Media, "Chapter 1. Introduction to the Cloud - The Basics of Cloud Computing." https://learning.oreilly.com/library/view/the-basics-of/9780124059320/xhtml/CHP001.html (accessed Apr. 10, 2021).

[5] "What is IaaS? Infrastructure as a Service — Microsoft Azure." https://azure.microsoft.com/en-us/overview/what-is-iaas/ (accessed Apr. 20, 2021).

[6] "Azure Infrastructure as a Service (IaaS) — Microsoft Azure." https://azure.microsoft.com/en-

us/overview/what-is-azure/iaas/ (accessed Apr. 20, 2021).

[7] "Amazon EC2." https://aws.amazon.com/ec2/?ec2-whats-new.sort-by=item.additionalFields.postDateTime
whats-new.sort-order=desc (accessed Apr. 20, 2021).

[8] "What is IaaS (Infrastructure as a Service)?," Google Cloud. https://cloud.google.com/learn/what-is-iaas (accessed Apr. 20, 2021).

[9] V. W. Team, "Configuration Drift: Why It Happens and How It Complicates Life for Incident Response Engineers," VictorOps, Nov. 07, 2019. https://victorops.com/blog/how-configuration-drift-affects-incident-response-engineers (accessed Apr. 20, 2021).

[10] M. Chan, "15 Infrastructure as Code Tools to Automate Deployments," Thorn Technologies, Apr. 03, 2018. https://www.thorntech.com/15-infrastructure-as-code-tools/ (accessed Apr. 20, 2021).

[11] O. Media, "Chapter 4. Cloud Service Models - The Basics of Cloud Computing." https://learning.oreilly.com/library/view/the-basics-of/9780124059320/xhtml/CHP001.html (accessed Apr. 14, 2021).

[12] OECD, OECD/G20 Base Erosion and Profit Shifting Project Addressing the Tax Challenges of the Digital Economy, Action 1 - 2015 Final Report. OECD Publishing, 2015.