INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




# DevOps Technologies for Tomorrow

**RÚBEN DOS SANTOS BARROS**
Outubro de 2016

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

# DevOps Technologies for Tomorrow


## Rúben dos Santos Barros


**A dissertation submitted in partial fulfillment of the requirements for the degree of Master of Science, Specialisation Area of Computer Systems**


**Supervisor: Ângelo Martins, PhD**
**Co-Supervisor: Tiago Boldt Sousa**


**Evaluation Committee:**
President:
Nuno Silva, PhD, DEI/ISEP

Members:

Porto, October 23, 2016

"*Bad behavior arises when you abstract people away from the consequences of their actions.*"

Jez Humble

# Abstract

DevOps (short for development and operations) is an approach based on lean and agile principles in which the departments of development, quality assurance, and operations collaborate to deliver software in a continuous manner that enables the business to seize market opportunities faster and reduce the time to include customer feedback.

Users and customers of today's Web applications and mobile apps running in the Cloud expect that fast feedback and features to their issues and requests respectively. Thus, it is a critical competitive advantage to be able to respond as quickly as possible. To achieve that, besides the necessary cultural and organizational changes, we need to use new tools to implement automations for the software workflow. Automation is the key to efficient collaboration and tight integration between development and operations. The DevOps community is constantly pushing new approaches, tools, and open-source artifacts to implement such automated processes. However, as all these proprietary and heterogeneous DevOps automation approaches differ from each other, it is hard to integrate and combine them to deploy applications in the Cloud.

With the recognition of the importance of DevOps, an explosion of technologies that address the subject was evident. However, a problem emerges; such diversity made it non-trivial for software teams to evaluate the wide range of existing tools and identify the best approach for them to practice DevOps.

In this dissertation, we thoroughly research DevOps, its concepts, and some of the most widely used tools, and gather the most relevant information as our defined goals. Consequently, we present a document with structured information regarding DevOps as well as a reliable DevOps Knowledge Map.

Throughout this document, we show that a DevOps approach brings many benefits to a wide range of different users and organizations, as it automates several processes while increasing the team's confidence and quality of life.

**Keywords:** DevOps, Software Development, Knowledge Map, Cloud Computing

# Resumo

DevOps, junção de *Development* (desenvolvimento) com *Operations* (operações) é uma abordagem baseada em métodos leves e ágeis em que os departamentos de desenvolvimento, controlo de qualidade e operações colaboram para entregar software de uma forma continua que permite o negócio de agarrar oportunidades de mercado mais rapidamente e reduzir o tempo da inclusão do feedback do cliente.

Os utilizadores e os clientes das aplicações web e mobile de hoje em dia esperam essa rapidez para o seu feedback e funcionalidades para os seus problemas e pedidos respetivamente. É, portanto, uma vantagem crítica de competitividade responder a esses pedidos o mais rápido possível. E para isso, para além da necessária mudança cultural e organizacional, precisamos de novas ferramentas para implementar automações para o fluxo de trabalho do nosso software. A automação é a chave para uma colaboração eficiente e uma integração mais profunda dos departamentos de desenvolvimento e operações. A comunidade de DevOps está constantemente a fornecer novas abordagens, ferramentas e artefactos *open-source* para implementar tais processos automáticos. Contudo, como todas essas abordagens proprietárias e heterogéneas de automação são diferentes de umas para as outras, torna-se complicado para integrar e combiná-las para meter aplicações na *Cloud*.

Com o reconhecimento da importância de DevOps, era evidente uma explosão de tecnologias para endereçar o assunto. Contudo, surge um problema; tanta diversidade faz com que não seja trivial para as equipas de software avaliar a ampla variedade de ferramentas e identificar a melhor abordagem para eles praticarem DevOps.

Nesta dissertação temos como objetivo, fazer uma pesquisa exaustiva de DevOps, os seus conceitos e algumas das ferramentas mais usadas e reunir a informação mais relevante. Consequentemente apresentámos um documento com informação estruturada sobre DevOps assim como um Mapa de Conhecimento de confiança acerca de DevOps.

Ao longo deste documento, mostramos que uma abordagem DevOps traz imensos benefícios para uma ampla variedade de utilizadores e organizações, graças à automação de diversos processos enquanto aumenta a confiança da equipa e a sua qualidade de vida.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

This chapter introduces the project, its context, and motivation, describe the contributions, and the outline of this document.

## 1.1 Context

Consider an organization which is working towards creating software. They had a very clear separation of responsibility between their development team (Dev), quality assurance (QA) team and operations (Ops) team / system administrators (SysAdmin). This exercise worked like a clock, controlled and accurate, and processes like product design, development, and support were predictable enough that companies and their employees scheduled their finances, vacations, surgeries, and mergers around product releases. At this point, operations took over, and their job was to maintain the software and the infrastructure running [1].

In the past, software operators were mainly responsible for the network, storage, and security of applications. This work was usually done manually through command-line code. Furthermore, the most proficient SysAdmin's had a portfolio of Perl scripts with a set of commands to automatically repair something in the system. However, these methods do not scale when we are talking about hundreds, thousands, or tens of thousands of servers (in the case of Google [2] and Facebook [3]). These methods are time-consuming, need the presence or action of the SysAdmin, and are even hard to manage that many machines manually and repeatedly.

However this separation comes with a price, and whilst the development teams strive for change, the operations teams strive for stability [4].

With the introduction of Cloud computing, the opportunity for smaller development teams to create competitive software was introduced, allowing them to keep their size while scaling at a global level. DevOps played a key role at allowing them such scalability without scaling the team size. Although there is not a concise way to describe DevOps, we believe it to be the description of a culture in which business owners and the development, operations, and quality assurance departments collaborate to deliver software in a continuous manner and encourages practices to evolve to meet that culture focusing on business instead of departmental objectives [5]. Thereby, modern applications, running in the cloud, still need to be resilient and fault tolerant, still need monitoring, still need to adapt to massive swings in load, etc. The teams need to become agiler, reduce problems like downtime, and time to deliver new content. Whereas the new sysadmin will not power down a machine, replace

a failing disk drive, reboot, and restore from backup; he'll write software to detect a misbehaving server instance automatically, destroy the bad instance, spin up a new one, and configure it, all without interrupting service.

The infrastructure does not go away — it moves into the code; and the people responsible for the infrastructure, the system administrators, and corporate IT groups evolve so that they can write the code that maintains the infrastructure. Rather than being isolated, they need to cooperate and collaborate with the developers who create the applications, this is the movement informally known as "DevOps. [6]

## 1.2 Motivation

DevOps involves a significant number of different topics such as Cloud, Virtualization, Monitoring, Automation, Log Management, and so forth. Consequently, professionals have to expand their areas of expertise, which can be exhilarating for some.

Since DevOps is relatively new, there is not a set of guidelines to follow and consequently implement it. Nonetheless, it is something that could help companies and even professionals to improve their product. Developers should be more operationally aware and build better, fault-tolerant and scalable software.

There are significant gains in using DevOps, and our motivation is to guide software teams to become more efficient, reach a larger amount of users with fewer resources.

## 1.3 Goals

We propose to thoroughly research DevOps and DevOps tools and capture a State of the art, identifying the tools categories and their key features to aggregate them. For each category, map features to tools that solve them, then discuss forces influencing the adoption of the available tools for each category. Then we will elaborate a DevOps Knowledge Map.

We want to identify which key questions software development teams should ask when starting a new project to determine the forces that will influence technological decisions.

Finally, we synthesized a set of guidelines for DevOps adoption, based on the projects characteristics (defined by the key questions), and test the application of those on a proof of concept project and publish our conclusions.

In summary, what body of knowledge should we collect and formalize to help software teams to practice DevOps? Which tools categories should we consider when elaborating the Knowledge Map? Which technologies should be adopted? How much will teams improve while using our Knowledge Map?

## 1.4 Contributions

By the end of this dissertation, the following contributions are expected:

- A literature overview concerning different DevOps categories as well as their tools;

- A Knowledge Map composed by the categories and tools studied;
- A set of guidelines on how to use the Knowledge Map.

With all of these in mind, we intend to help DevOps practitioner teams, to be able to determine the areas of intervention, their problems and possible solutions, including which tools should be adopted to help them achieve their goals.

While this dissertation is scoped for new software projects, we believe that the knowledge imparted could help teams with existing projects.

## 1.5 Outline

This dissertation is the final iteration of an extensive research process to aggregate a large quantity of spread information concerning different DevOps categories and their tools.

The chapter 2 starts by presenting the current state of software development, the Cloud, and then DevOps. This serves as, respectively, an introduction to some key concepts needed to understand this document and a base on which we based some of our studies.

The chapter 3 describes the problem, our work proposal, and the value analysis concerning that same work.

The chapter 4 is the culmination of our research, which starts with our Knowledge Map, which states and identifies the nine addressed DevOps categories and their tools, followed by a detailed description concerning the same categories and tools.

The chapter 5 presents the methodology used to validate our work as well as the results of that validation.

Finally, the chapter 6 identifies the main contributions of this thesis and suggests some work for the future.

# Chapter 2

# State of the Art

This chapter discusses existing work and research relevant to the context of the problem.

## 2.1 Traditional Software Development

We consider that an organization employs a traditional software development when they clearly separate their personnel by their specialty, those people perform rudimentary processes, and fulfill activities manually and repeatedly, and own one or a set of servers to deploy their products.

### 2.1.1 Departmentalization of People

Traditional organizations consist of multiple teams which divide their workload by type of work. Normally they consist of a Development department, an Operations department, and sometimes a Quality Assurance department [5].

The Development department, depending on the organization, has several roles, as can be seen bellow [7]:

- **Interaction Designer**, to ensure the "interaction goals" of the system are articulated and agreed by the stakeholders, and develops user personas;
- **Chief Engineer**, to determine the needs that the product must meet and continually oversees the development of the product to ensure that it is on target to meeting those needs, and to listen to the stakeholders and then negotiates with the project team to address their needs;
- **Architect**, to ensure the product or service under development achieves its performance and other qualitative requirements, and guides the interfacing and integration of the solution components of this project into the existing architectural landscape;
- **GUI Designer**, to guide look-and-feel decisions, help write and clarify user stories and personas, elaborate user interaction guidelines and standards;
- **Requirement Analysts**, for solicitation and elaboration of stakeholders needs and requirements;
- **Designer**, to find solutions for those requirements;
- **Coder**, to implement the user stories;

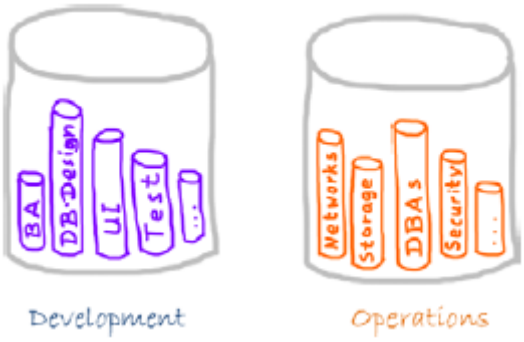



Figure 2.1: Development and operations are two distinct departments. Often, these departments act like silos because they are independent of each other [4]

- **Database Designer**, to ensure that a product meets production requirements, and plans product features to support e.g. data schema migration.

Someone in some organization might assume one or more of these roles. However, others can be a full dedicated team, which can result in many work stoppages when the product goes from one team to another.

The Quality Assurance department is in charge of approving the application before going to the clients. They have one or more **Quality Assurance** professional with the following responsibilities:

- Plan and create the tests for the application;
- Execute the tests, identify possible errors, and report them back to the Development for resolution.

The Operations department has a **System Administrator** and may or may not have many professionals working alongside him, and they have duties such as:

- Take the product to production and maintain it fully operational;
- Monitor and maintain the server, the network, and the database;
- This department has to be always prepared to resolve possible problems with all the previously mentioned components.

Splitting work areas appears to benefit the management as well. In addition to the specialized team, each department has its manager who fulfills the individual requirements needed for this particular department. Each department defines its goals based on the division of labor. The development department may be measured by its speed in creating new features, whereas the operations department may be judged by server uptime and application response time. Unfortunately. operations is considered to be successful if the metrics are stable and unchanging, whereas development is only applauded if many things change. Because conflict is baked into this system, intensive collaboration is unlikely. Development teams strive for change, whereas operations teams strive for stability. The conflict between development and operations is caused by a combination of conflicting motivations, processes, and tooling. As a result, the departments isolation evolve.

In a nutshell, the conflict between development and operations is as follow:

- **Need for change**: Development produces changes like new features, bug fixes, and work based on change requests, and wants that their changes go into production.
- **Fear of change**: Once the software is delivered; the operations department wishes to avoid making changes to the software to ensure stable conditions for the production systems.

However, as for the development departments, this process was improving with the introduction of Agile methods, which helped them to fulfill their objectives faster and better. The Agile movement has brought together programmers, testers, and business representatives. Conversely, operations teams are isolated groups that maintain stability and enhance performance. The conflict between the two groups can only be healed and the silos bridged by aligning the two groups' different goals. To do so, Agile methods must be applied to operations as well [4].

### 2.1.2 Infrastructure and Deployment

A while back, the usual decision to provide an application or services to clients was to buy one or multiple servers, mount the network, and work to maintain them online the maximum amount of time possible.

When the same application or service would be ready to be delivered, the operations team had the responsibility to install, configure, and maintain them working. All these processes were done manually, repeatedly, and sometimes in more than one server.

## 2.2 The Internet

The creation of the Internet has marked the foremost milestone towards achieving this grand 21st-century vision of 'computer utilities' by forming a worldwide system of computer networks that enables individual computers to communicate with any other computers located elsewhere in the world. This internetworking of standalone computers reveals the promising potential of utilizing the seemingly endless amount of distributed computing resources owned by various owners, which empowers computer processing and scaling [8].

Applications making use of these utility-oriented computing systems emerge simply as catalysts or market makers, which brings buyers and sellers together.

## 2.3 Cloud Computing

The cloud is not simply the latest fashionable term for the Internet. Though the Internet is a necessary foundation for the cloud, the cloud is something more than the Internet. The cloud is where someone goes to use technology when they need it, for as long as they need it, and not a minute more. They do not install anything on their desktops, and they do not pay for technology when they are not using it [9].

The cloud can be both software and infrastructure. It can be an application someone access through the Web or a server that they provision exactly when they need it. Whether a service is software or hardware, the following is a simple test to determine whether that service is a cloud service: "If someone walks into any library or Internet cafe and sit down at any computer without preference for operating system or browser and access a service, that service is cloud-based." [9]

Cloud computing empowers companies to consume computing resources as a utility, in the same way that electricity and water are consumed. This possibility removes the obligation to build and maintain a computing infrastructure in-house, which helps to maximize the effectiveness of the shared resources [10]. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. For example, a cloud computer facility that serves European users during European business hours with a particular application (e.g., email) may reallocate the same resources to serve North American users during North America's business hours with a different application (e.g., a web server). This approach helps maximize the use of computing power while reducing the overall cost of resources such as using less power, air conditioning, and rack space to maintain the system [11].

Thus, Cloud Computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased. Developers with innovative ideas for new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it. They need not be concerned about overprovisioning for a service whose popularity does not meet their predictions, thus wasting costly resources, or underprovisioning for one that becomes wildly popular, thus missing potential customers and revenue. Moreover, companies with large batch-oriented tasks can get results as quickly as their programs can scale, since using 1,000 servers for one hour costs no more than using one server for 1,000 hours [12].

Cloud Computing is associated with the following characteristics [13]:

- **On-demand self-service**: A client can unilaterally provision computing capabilities, such as server time and network storage, without requiring human interaction with each service provider.
- **Broad network access**: Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous client platforms (e.g., mobile phones, tablets, laptops, and workstations).
- **Resource pooling**: The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state or datacenter). Examples of resources include storage, processing, memory and network bandwidth.
- **Rapid elasticity**: Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the client, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

- **Measured service**: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth and active user accounts). Resource usage can be monitored, controlled and reported, providing transparency for the provider and consumer.

### 2.3.1 Cloud Business Models

A cloud provider is a company that delivers cloud computing based services and solutions to businesses and individuals. This service organization may provide virtual hardware, software, infrastructure and other related services.

Cloud providers deliver cloud solutions through on-demand, pay-as-you-go systems as a service to customers and end users. Cloud provider customers access to cloud resources through The Internet, and are billed only for resources and services used according to a subscribed billing method [14].

Depending on the business model, a cloud provider may provide various solutions, 2.2, such as:

- Infrastructure as a Service (IaaS): May include virtual servers, virtual storage, and virtual desktops/computers;
- Software as a Service (SaaS): Delivery of simple to complex software through the Internet;
- Platform as a Service (PaaS): A combination of IaaS and SaaS delivered as a unified service;
- A cloud provider also may be classified as a public cloud provider, private cloud provider, hybrid cloud provider or community cloud provider.

## 2.4 DevOps

The term DevOps was first used when Patrick Debois had to name a conference about the cooperation between Development and Operations. The conference took place in Belgium, October 2009, and was called DevOpsDays. The successful conference generated a lot of discussion in Twitter, which led to the creation of the hashtag #DevOps. Since then, the movement is known as DevOps [15].

DevOps (short for development and operations) is an approach based on a lean and agile principles in which business owners and the development, operations, and quality assurance departments collaborate to deliver software in a continuous manner that enables the business to seize more quickly market opportunities and reduce the time to include customer feedback [5].

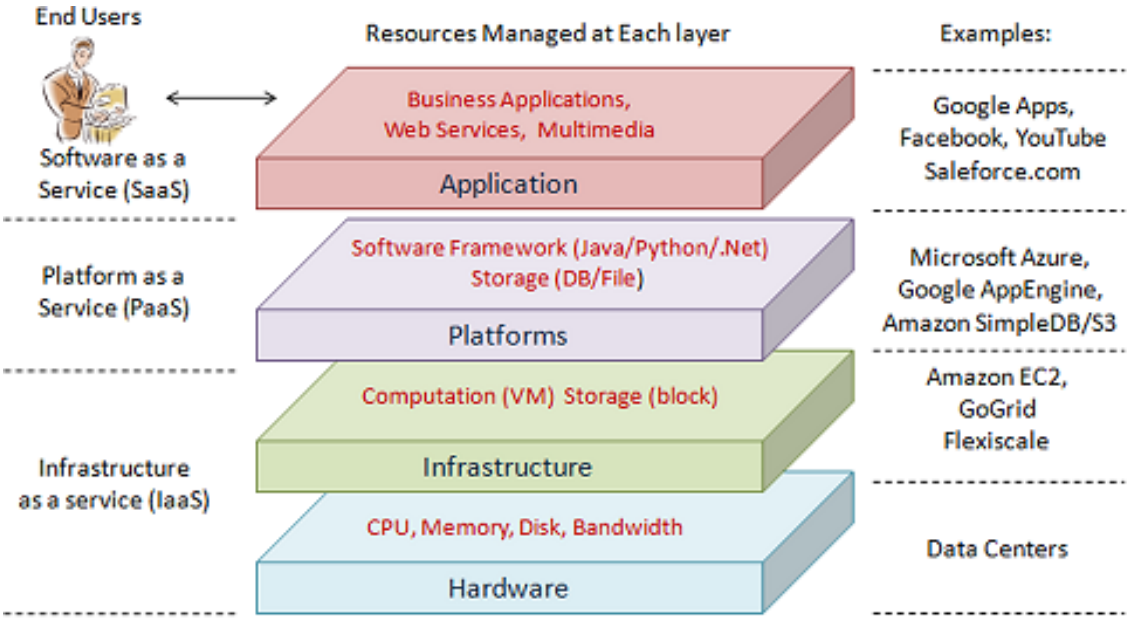



Figure 2.2: Cloud Providers present different infrastructures solutions for clients with different needs

### 2.4.1 DevOps in Numbers

Based on a research survey conducted by Stephen Elliot [16] during October and November of 2014 across multiple industries and with respondents across development, testing, and operations, he could identify critical DevOps metrics from 20+ Fortune 1000 organizations. The results show:

- For the Fortune 1000, the average total cost of unplanned application downtime per year is $1.25 billion to $2.5 billion.
- The average hourly cost of an infrastructure failure is $100,000 per hour.
- The average cost of a critical application failure per hour is $500,000 to $1 million.
- The average number of deployments per month is expected to double in two years.
- IT organizations that have tried to custom adjust current tools to meet DevOps practices have a failure rate of 80%, thus making tool replacement and/or addition a critical requirement.
- There is an expectation that DevOps-led projects will accelerate the delivery of capabilities to the customer by an average of 15–20%.
- The average cost percentage (per year) of a single application's development, testing, deployment, and operations life cycle considered wasteful and unnecessary is 25%.
- Over the next two years, DevOps teams will increasingly bring security, compliance, and audit teams into the project-planning cycle to embed some of these requirements in automated processes to reduce business and security risks.

Table 2.1 shows that 43% of the respondents are currently using DevOps practices, while another 40% are currently evaluating DevOps as a way to extend their existing agile investments across the split departments, include business executives earlier in the cycle, and deliver more business value. There is no doubt that this trend will continue, with more organizations adopting DevOps practices and organizing themselves around it. The business value is too much to ignore.

| | % of Respondents |
|---|---|
| Currently evaluating DevOps practices | 40.0 |
| Less than a year | 10.0 |
| 12-24 months | 13.3 |
| 25-48 months | 13.3 |
| More than 4 years | 6.7 |
| Don't know | 16.7 |

Table 2.1: DevOps Practices Timeline

Table 2.2 shows the average time to repair an infrastructure failure. Table 2.3 shows the average amount of time to fix an application failure. Over the past five years, the average time to repair on the infrastructure side has improved faster than on the application side. However, we expect DevOps practices, and the associated tools, to exponentially improve the average time it takes to repair application failures.

| | % of Respondents |
|---|---|
| 1-5 minutes | 9 |
| 11-59 minutes | 17 |
| 1-12 hours | 35 |
| 2-7 days | 17 |
| Don't know | 22 |

Table 2.2: Average Time to Repair an Infrastructure Failure

| | % of Respondents |
|---|---|
| 6-10 minutes | 4 |
| 11-59 minutes | 9 |
| 1-12 hours | 35 |
| More than 12-24 hours | 4 |
| More than 1-7 days | 13 |
| Don't know | 35 |

Table 2.3: Average Time to Repair an Application Failure

To examine application downtime impact per year across the Fortune 1000, Stephen Elliot conducted the following analysis: 1,000 companies multiplied by an average of 250 critical applications per company equals 250,000 total applications. 250,000 multiplied by an average of $500 to $1 million per hour of downtime cost equals $1.25 billion to 2.5 billion of unplanned application downtime costs per year.

Table 2.4 indicates the amount of time it takes to restore a service in production that has failed. Table 2.5 indicates how long it takes a code change to reach the customer.

| | % of Respondents |
|---|---|
| Less than an hour | 19 |
| 1 to <3 hours | 33 |
| 3 to <8 hours | 14 |
| 8-24 hours | 5 |
| Don't know | 29 |

Table 2.4: Average Time to Restore a Production Failure

| | % of Respondents |
|---|---|
| 0-60 minutes | 14 |
| 61 minutes - 24 hours | 10 |
| More than 1-6 days | 10 |
| 1-2 weeks | 24 |
| More than 2-4 weeks | 5 |
| More than 1-3 months | 10 |
| More than 3 months | 5 |
| Don't know | 22 |

Table 2.5: Timeline for Code Change Impact on Customers

### 2.4.2 Recognizing the Business Value of DevOps

DevOps applies agile and lean principles to the entire software supply chain. It enables a business to maximize the speed of its delivery of a product or service, from initial idea to production release to customer feedback to enhancements based on that feedback. Because DevOps improves the way that business delivers value to its customers, suppliers, and partners, it is an essential business process, not just an IT capability.

DevOps provides significant return on investment in three areas:

- **Enhanced customer experience**: This experience builds customer loyalty and increases market share. To deliver this experience, a business must continuously obtain and respond to customer feedback, which requires mechanisms to get fast feedback from all the stakeholders in the software application that is being delivered: customers, lines of business, users, suppliers, partners, and so on;
- **Increased capacity to innovate**: Modern organizations use lean thinking approaches to increase their ability to innovate. Their goals are to reduce waste and rework and to shift resources to higher-value activities;
- **Faster time to value**: Speeding time to value involves developing a culture, practices, and automation that allow for fast, efficient, and reliable software delivery through to production. DevOps provides the tools and culture required to facilitate efficient release planning, predictability, and success.

### 2.4.3 How DevOps Works

DevOps is not a button someone switches; it is not an application someone installs; it is not a script that someone attaches to their application. There is not a correct or incorrect

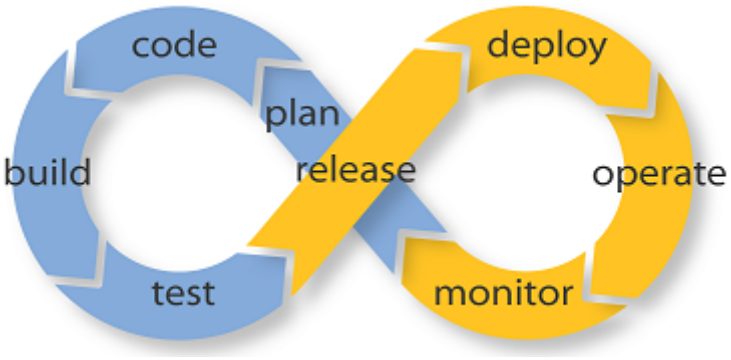



Figure 2.3: DevOps can create an infinite loop of release and feedback for all your code and deployment targets.

way of practicing DevOps. DevOps describes a culture and encourages practices to evolve to meet that culture.

The DevOps movement has produced several principles that have evolved over time and are still evolving. Several solution providers, including IBM, have developed their variants. All these principles, however, take a holistic approach to DevOps, and organizations of all sizes can adopt them. These principles are [5]:

- **Develop and test against production-like systems**: The goal is to allow development and quality assurance teams to develop and test against systems that behave like the production system so that they can see how the application behaves and performs well before it is ready for deployment;
- **Deploy with repeatable, reliable processes**: This principle allows development and operations to support an agile software development process all the way through to production. Automation is essential to create processes that are iterative, frequent, repeatable, and reliable, so the organization must create a delivery pipeline that allows for continuous, automated deployment and testing;
- **Monitor and validate operational quality**: Organizations typically are good at monitoring applications and systems. However, this principle moves monitoring earlier in the life cycle by requiring that automated testing be done early and often. Whenever an application is deployed and tested, quality metrics should be captured and analyzed to get early warning about operational and quality issues that may occur in production;
- **Amplify feedback loops**: This principle appeals organizations to create communication channels that allow all stakeholders to access and act on feedback.
  - Development may act by adjusting its project plans or priorities;
  - Production may act by enhancing the production environments;
  - Business may act by modifying its release plans.

### 2.4.4 The Adoption of DevOps

At its root, DevOps is a cultural movement; it is all about people. An organization may adopt the most efficient processes or automated tools possible, but they are useless without the people who eventually must execute those processes and use those tools. A DevOps culture is characterized by a high degree of collaboration across roles, focus on business instead of departmental objectives, trust, and high value placed on learning through experimentation [5].

The DevOps reference architecture shown in Figure 2.3 proposes the following four sets of adoption paths:

- **Plan**: This adoption path consists of one practice that focuses on establishing business goals and adjusting them based on customer feedback: Continuous Business Planning. DevOps help teams to set business objectives and continuously change them based on customer feedback. By identifying and eliminating waste in the development process, the team becomes more efficient but also addresses cost;
- **Develop**/**Test**: This adoption path involves two practices:
  - Collaborative development: Developers, software architects, QA practitioners, operations personnel, security specialists, business analysts, suppliers, partners, and lines-of-business owners work together by providing a common set of practices and a common platform they can use to create and deliver software. This practice also includes Continuous Integration, in which software developers continuously or frequently integrate their work with that of other members of the development team;
  - Continuous testing: This practice means testing earlier and continuously across the life cycle, which results in reduced costs, shortened testing cycles, and achieved continuous feedback on quality.
- **Deploy**: Continuous release and deployment take the concept of Continuous Integration to the next step. The practice that enables release and deploy also allows for the creation of a delivery pipeline. The goal is to release new features to customers and users as soon as possible;
- **Operate**: This Operate adoption path includes two practices that allow businesses to monitor how released applications are performing in production and to receive feedback from costumers:
  - Continuous Monitoring: Provides data and metrics to operations, QA, development, lines-of-business personnel, and other stakeholders.
  - Continuous customer feedback and optimization: This feedback allows different stakeholders to take appropriate actions to improve the applications and enhance customer experience.

### 2.4.5 Patterns and Practices

For a team to correctly practice DevOps is not trivial. Several steps and practices are critical to establishing successful DevOps culture in the organization; some are as follow [17]:

- Active Stakeholder Participation: developers, operations staff, and support people must work closely together on a regular basis;
- Automated testing: Adopt test-first approach. Automate testing so that the application is tested as often and early as possible;
- Integrated configuration management: Development teams will need to understand, and manage, the full range of dependencies for their product so that operations staff can understand the potential impact of a new release;
- Integrated change management: Development teams must work closely with operations teams to understand the implications of any technology changes. Integrated change management depends on the Active Stakeholder Participation, Integrated configuration management, and Automated testing.
- Continuous Integration: Is the discipline of building and validating a project. Continuous Integration enables the development of a high-quality working solution safely in small, measured steps by providing immediate feedback on code defects;
- Integrated deployment planning: Experienced development teams will do such planning continuously throughout construction with Active Stakeholder Participation.
- Continuous deployment: Extend the Continuous Integration till production. Continuous deployment enables development teams to reduce the time between a new feature being identified and being deployed into production;
- Production support: Development teams should ensure production fixes and regular releases both are taken care without issues;
- Application Monitoring: Development teams need to be aware of what is happening with the product;
- Automated dashboards: Code quality, build, and deployment dashboard should be real time.

All this, through a collaborative working team, which is key to DevOps.

Table 2.6 shows what we would expect from an evolving trend that is impacting budget allocations and the projects receiving funding. Automation is a key investment area across both development and operations teams, as the notion of continuous delivery remains a focal point.

| | % of Respondents |
|---|---|
| Automation | 60.0 |
| Continuous Delivery | 50.0 |
| Continuous Integration | 43.3 |
| Automated Testing | 43.3 |
| Application Monitoring/Management | 43.3 |
| IT Operations | 33.3 |
| Git Branch Builds | 26.7 |
| Log Analytics | 23.3 |

Table 2.6: Initiatives to Implement DevOps Projects [16]. Note: Multiple responses were allowed.

### 2.4.6 Technologies

In the midst of several technologies, we took into consideration the following technologies:

- **Cloud Infrastructure** or **Cloud Provider**: the final product will be built, deployed and managed in one of many available infrastructures;
- **Cloud Infrastructure Management**: abstracts CPU, memory, storage, and other resources, enabling fault tolerant and resilient distributed systems to be effortlessly built and run efficiently;
- **Virtualization**: refers to the act of creating a virtual operating system within the Infrastructure. This way, we can run multiple operating systems on a single piece of hardware. A Virtualized System is nothing more than a data file on a physical computer that can be moved or copied to another computer;
- **Supervision**: handles starting of tasks and services, such as the deployed application and its dependencies, during boot, stopping them during the shutdown and supervising them while the system is running;
- **Load Balancing**: help when there's a need to have more than one system running the same application; it has the capability of equally divide work between those systems;
- **Automation**: Automation tools help developers and operations to deploy and configure systems and applications to any physical, virtual, or cloud location, no matter the size of the infrastructure. These tools help orchestrate the resources in the Virtual System, or even create or destroy a similar system automatically;
- **Monitoring**: is primarily a type of security and surveillance software that may help Supervision and Automation. It observes and tracks the operations and activities of users, applications, and network services on a computer, a virtualized system or application [18];
- **Data Logging**: is the process of collecting and storing data over a period of time. This process allows development teams to, quickly, trace problems with their application to fasten the resolution time [19];
- **Scheduling**: is performed using job schedulers which have the ability to run services periodically at fixed times, dates, or intervals. This process automates system maintenance. However it has a general purpose making it useful for things like starting a new Virtualized System [20];
- **Service Discovery**: is a network protocol which allows automatic detection of devices and services offered by these devices on a computer network. If this service is correctly configured, with the help of Automation, it can detect new instances of the application running, and allows software agents to make use of one another's services without the need for continuous user intervention and configuration [21].

For each of these categories, there are many options with different features, requirements, and prices, which can confuse and divide teams into choosing one or another.

### 2.4.7 Technological Relations

As seen in Figure 2.4, we can observe that all the technologies are intertwined towards a common goal; the automatic, correct, and continuous functioning of the Application. The Cloud Infrastructure Management orchestrates the resources provided by the Cloud Provider and configures the Scheduler, the Supervision, and the Monitor. The Scheduler performs scheduled actions on the machine like an update or the execution of a script. The Virtual Machine will host the Application and manage it as well as other services with Supervision. Supervision will start the Application and any other dependencies concerning it or the system. The Monitor will monitor the correct functioning of the Virtual Machine and its services, like the Application. A Logging service will register the Application logs so as to help teams to quickly find out problems in their ecosystem. The Application registers itself in the Service Discovery so that Service Discovery knows where the Application is running. The Service Discovery queries the Application to know if it is still online.

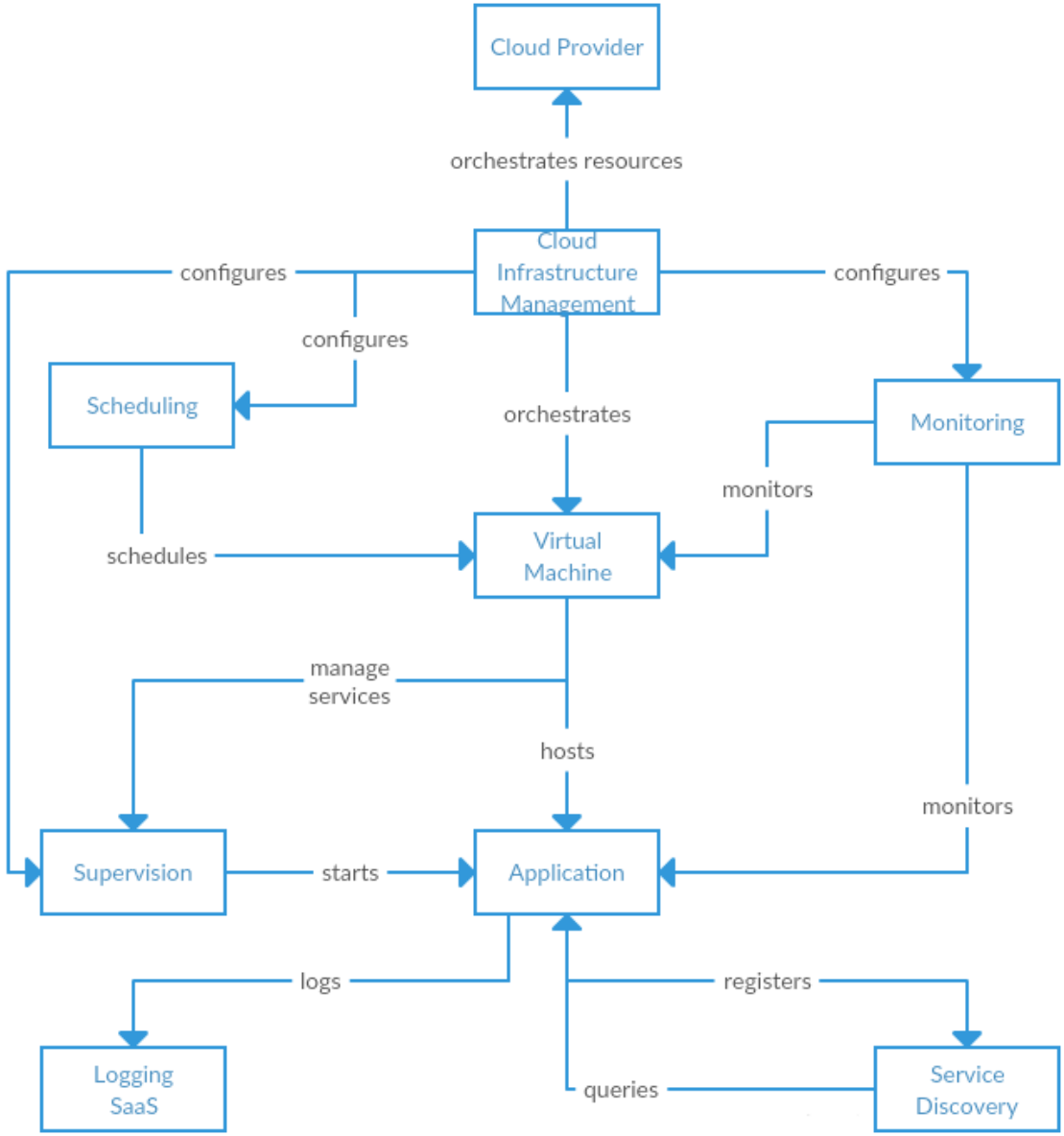



Figure 2.4: The relation between some of the presented technologies

# Chapter 3

# Problem Definition

The problem in question is that DevOps practices are relatively new, and we are only grasping the full potential and depth about it. Starting companies do not have the information to help them choose what they will need, losing valuable time and resources investigating and experimenting until deciding which tools appeals to them.

In this chapter, the problem is identified. The research questions are raised and the formalization of the proposed solution is presented. We also take a look at the value analysis of the dissertation.

## 3.1 Problem

With the recognition of the importance of DevOps, an explosion of technologies that address the subject was evident. However, a problem emerges; such diversity made it non-trivial for software teams to evaluate the wide range of existing tools and identify the best approach for them to practice DevOps. Since DevOps is a relatively new ideology, it is safe to assume that the vast majority of software development teams do not grasp the full range of areas that practicing DevOps involves. So the problem resides specifically in the need to determine the areas of intervention, their problems, and their possible solutions.

Our challenge is the capture of such information and the creation of a Knowledge Map that can be simply evaluated both when starting or while a software project to help teams moving towards DevOps.

## 3.2 Thesis Statement and Questions

We identified four questions that we want to answer at the end of this dissertation, these being as follow:

- **What body of knowledge should we collect and formalize in order to help software teams to practice DevOps?** - We want to research on a broad set of technologies to be able to elaborate a thorough and trustworthy Knowledge Map.
- **Which tools categories should we consider when elaborating the Knowledge Map?** - We need to ascertain which requirements might influence teams in choosing a tool over another.

- **Which technologies should be adopted?** - We want to help teams choosing technologies and to understand their requirements and features.
- **How much will teams improve while using our Knowledge Map?** - We need to learn and quantify how much a team improves to prove that our work is reliable.

**Thesis statement**

The author believes that providing the right educational content to software teams, they can become more efficient and effective managing the entire life-cycle of the software they produce through the adoption of DevOps.

## 3.3 Work Proposal

Our work proposal for the identified problem is a thorough research on the Internet and research articles, about DevOps tools so that we could ease the adoption process by software teams, by aggregating DevOps tools into categories and compare members of each category considering their key features. Identify which questions should be asked when starting a new project to identify the forces that will influence technological decisions.

Furthermore, we should be able to use the captured knowledge to elaborate a DevOps Knowledge Map. Such would determine the areas of intervention for DevOps practitioner teams, their problems and possible solutions to those, including how and which tools should be adopted to help them achieve their goals.

Then we could evaluate how forces can be leveraged to achieve optimal solutions for specific scenarios, and provide a concrete description of how to use the DevOps Knowledge Map to provide a solution for common DevOps problems, usable both by regular operation teams or DevOps practitioners.

Finally, we should be able to verify and validate the DevOps Knowledge Map, and provide the work to one or more software teams and measure how it improves their operations efficiency.

## 3.4 Value Analysis

### 3.4.1 The importance of a Value Proposition

The value proposition, which is the central element of the value model and the core of a business model helps an organization to present to people if and how its business will profit, which partners it needs, the nature of its key operations, and how it will acquire and retain customers. This value proposition is an overall view of the organization bundle of products and services that together represent value for a specific customer segment, and it describes the way an organization differentiates itself from its competitors while clarifying why customers buy from it and not another organization [22].

These products and services are the value offered by the organization, which is key to any business, and any business activity is about exchanging some tangible and/or intangible good or service and having its value accepted and rewarded by customers or clients, either inside the enterprise or collaborative network or outside. Besides, it becomes necessary to bear in

mind the perception of cost/benefit for the organization and the client, as a factor which influences the relation between the parts, providing a better knowledge of the customer and the market [23].

There is also the concept of perceived value, in which different clients perceive different value for the same product/service. Also, organizations involved in the purchasing process can have different perceptions of customers' value delivery. Furthermore, that value, as perceived by the producer, means something distinct from the value perceived by the user.

In our case, we want to provide the knowledge around the DevOps processes and tools to be used by different kinds of organizations. As it is a recent approach on how IT teams should refactor their workflows, there are not much certainty around the concept as a whole, and the fact that the technologies are growing at tremendous speeds, to consolidate their markets, it is hard to find up-to-date information as well as an aggregation concerning those tools.

### 3.4.2 The benefits/sacrifices of following a DevOps mindset through a longitudinal perspective

As mentioned before we aim to develop a Knowledge Map, to facilitate the adoption of DevOps, which will be our primary focus. The main objectives for this map are to provide software teams with the knowledge about several DevOps tools and their characteristics, advantages, and disadvantages, as to save an enormous amount of time researching about those tools and eventually choose one of them.

Regarding market positioning, we want to help organizations, more specifically software development teams and operations teams, to save time and money while achieving the same or even better results through our extensive study surrounding DevOps and its tools.

In a Longitudinal perspective of value becomes necessary to considerate the benefits and sacrifices for those teams before acquiring our Knowledge Map, the moment when they acquire it, the moment when the team finishes using it, and the use experience after using it. Nowadays there are still many teams managing the infrastructure manually, which is a tedious, error-prone, and slow task. However, when an organization thinks about adopting DevOps to improve their product/service, through the knowledge of our Knowledge Map, there is a pre-purchase moment in which the organization expects to save time in the future when using DevOps techniques and tools. Nevertheless, they know that there is a cost of learning and configuring those tools. At the point of trade, the team gets the Knowledge Map, and they study and choose the best tool for their case scenario. After getting and utilizing the Knowledge Map and learning and configuring the selected DevOps tools, besides saving much time managing the infrastructure, the software team is expected to have the capacity to update the product in a more convenient and repeatable manner. These processes should give more confidence to the team when updating their product/service, allowing them to reach their clients much faster. The product should become much more stable, improving the quality of the product, as well as increase exponentially the quality of life of every employee of the organization. Finally, it will be much easier to customize and change where the product and service will be deployed. Finally, after adopting those techniques and tools, the software team should realize that their product/service improved exponentially as well as the quality of their life in relation to when they were using the old fashioned techniques. Table 3.1 states the benefits and sacrifices before and after using our Knowledge Map.

| | Benefits | Sacrifices |
|---|---|---|
| **Ex-Ante** | 1. Save time managing infrastructure | 1. Time spent learning and configuring tools and processes |
| **Ex-Post** | 1. Save time managing infrastructure<br>2. Updating the product is more convenient and repeatable<br>3. Teams have more confidence when updating the product/service<br>4. Faster time-to-market<br>5. Product/Service becomes more stable<br>6. Quality of life of the organization's employees improves<br>7. Quality of the product/service improves<br>8. It is much easier to customize where the product/service is deployed | 1. Time spent learning and configuring tools and processes |

Table 3.1: Benefits and sacrifices of adopting the Knowledge Map

### 3.4.3 Possible business scenarios

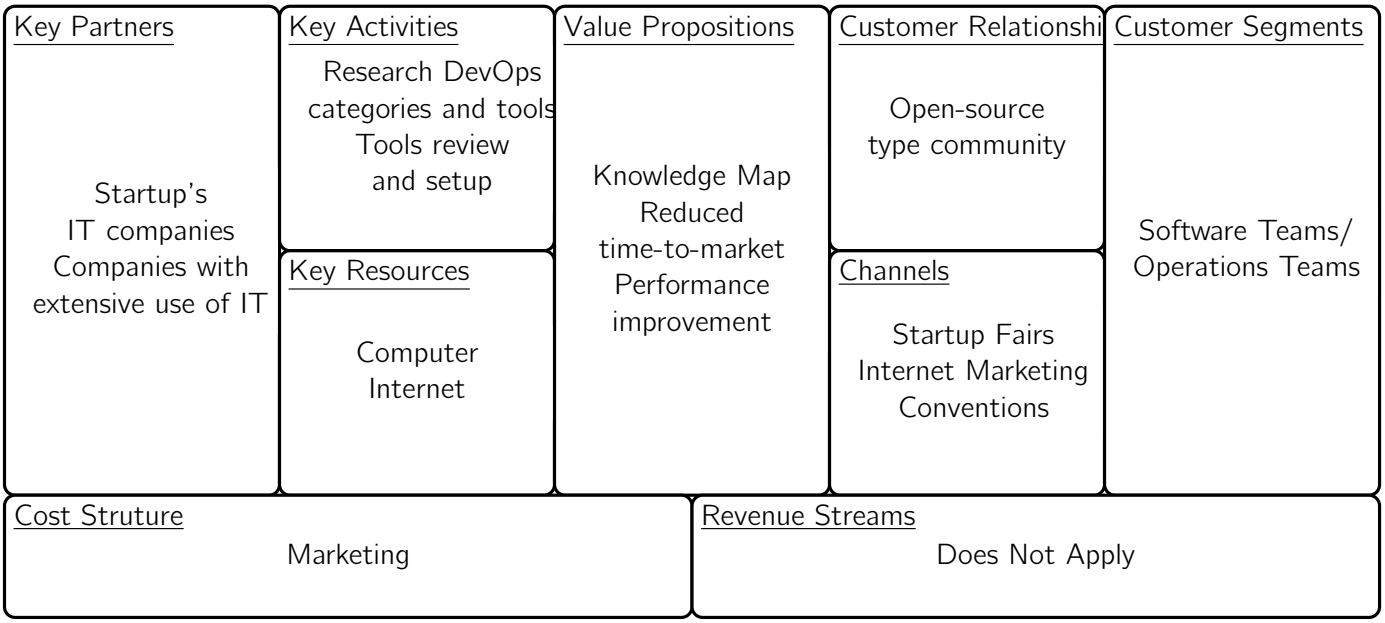
Regarding the business, we reach a win-win situation through an integrative negotiation, in which both parties collaborate to reach the maximum benefits possible. In this kind of negotiation, it is necessary to define the expectations and objectives beforehand and identify the indisputable points of the negotiation. On the other hand, we need to foresee the counter-offers and still be prepared for what the client expects. It is also important to be prepared to explain the reasons behind the maximum and minimum we will get or receive.

### 3.4.4 The business model Canvas

We developed a business model Canvas to give better insight about our project, which can be observed below. In this case, we believe that our Key Activities are the research of DevOps categories as well as the research and setup of some market leader tools. These activities aim to facilitate the adoption of DevOps processes and tools for Startup companies as well as for well-established IT companies and even companies that rely a lot on IT, which would be our Key Partners. As for the Value Propositions, we intend to provide a Knowledge Map with different concepts and tools for various scenarios for organizations to use, which then will significantly improve their performance, and consequently, will reduce their time-to-market.

We would like to freely provide our Knowledge Map at different startup fairs and conventions as well as through the internet, believing that our project is purely academic and open-source giving up any possible Revenue Streams. However, given the open-source approach, we expect to have a relationship with our costumers through an open-source type community to exchange information.

Finally, the Cost Structure for our business will be purely around the marketing of it.




## 3.5 Conclusions

In this chapter, the problem to be tackled in the next semester was formalized.

First, the problem itself and its scope were described and delimited. Then a possible approach towards our goals was discussed along with the value analysis of the dissertation.

# Chapter 4

# DevOps Technologies for Tomorrow

Having discussed the evolution of how we handle the delivery of applications and the value of delivering faster and automatically in chapter 2, in this chapter, we will present the technologies and tools studied and present a conclusion about them.

Fortunately or unfortunately the DevOps field is too valuable to pass on, and as such there are dozens of alternatives for each category, which we had to pass on, opting for the market leaders. Also, during the conception of this document, the technologies are continuously evolving and presenting more and better features for their ever demanding clients, which may cause some lack of accuracy in some statements.

## 4.1 Knowledge Map

The main focus of our project is presented through a Knowledge Map, which can be observed in Figure 4.1. All of its nodes present DevOps categories and their tools, which are addressed throughout this thesis, and in every single one of them is given a technological overview to guide software teams to take the most appropriate decision depending on the project Requirements when choosing to adopt one or more DevOps oriented technologies.

## 4.2 Infrastructure

When an organization prepares to release an application, one of the first and most important decisions is where that application will be deployed. A few years ago the decision was "easy" and involved the purchase of machines and hiring people to operate them, but nowadays cloud computing became accessible and a viable option.

### 4.2.1 In-House Solution

The decision to opt for an in-house solution would mean to a great investment right from the get-go, involving the hardware to run the application and the people to operate it. However, some organizations already have the means, but even though the decision is not that straightforward. The application might need better hardware or the possibility to scale easily, thanks to market demands [24].

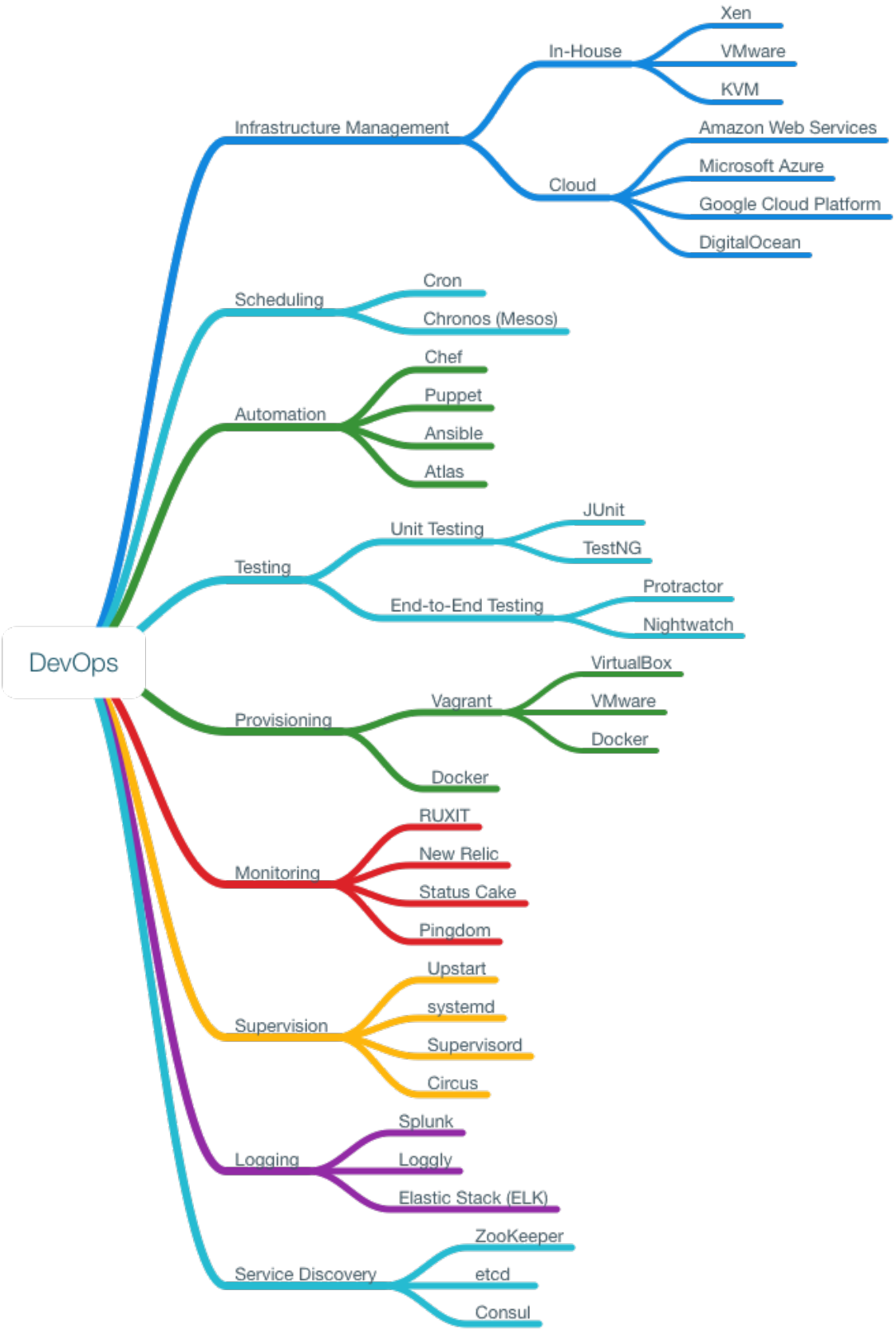



Figure 4.1: Knowledge Map with DevOps Tools

Organizations choosing the In-House option must have in mind the investment in hardware, the people to operate it, the future investment in hardware improvement, the network involving the infrastructure, the replication of the data, the security of the data, and much more [25].

#### 4.2.1.1 Xen

Xen was first developed at University of Cambridge Computer Laboratory, in 2003, and is now maintained by The Linux Foundation. Xen is intended for an enterprise utilization, and companies like Amazon and Linode are using it to provide cloud services and virtual machines for their clients [26].

Xen is available as a special Linux distribution called XenServer. However, it can be installed from other Linux distributions like centOS, Fedora, openSUSE, and Ubuntu. One of its features is its ability to host all the main operating systems like Linux, Windows, and Solaris. Xen is also capable of simulate x86, x64, and ARM architectures and, like other hypervisors, is capable of Full Virtualization, Paravirtualization, and Live Migration [27].

Since Xen is an open source project it does not provide a customer support service. However, Citrix, who developed XenServer, provides those services [28].

**Full virtualization**

With Full Virtualization the guest Operating System is unaware that it is in a virtualized environment. Therefore hardware is virtualized by the host Operating System so that the guest can issue commands to what it thinks is actual hardware but are just simulated hardware devices created by the host [29].

Full Virtualization is the only option that requires no hardware assist or Operating System assist in virtualizing sensitive and privileged instructions. With this kind of virtualization, we get the best isolation and security for virtual machines and simplifies migration and portability. However, there is a cost in performance and efficiency for the guest Operating System [30].

**Paravirtualization**

Paravirtualization is an enhancement of virtualization technology in which a guest Operating System is modified, prior to installation inside a virtual machine. With this modifications, the guest Operating system is aware that it is a guest and accordingly has drivers that, instead of issuing hardware commands, simply issue commands directly to the host Operating System. This virtualization method offers an increase in Performance and efficiency for the guest Operating System. A potential downside of this approach is that such modified Operating Systems cannot ever be migrated back to run on physical hardware.

VMware has employed this concept by making available VMware Tools, a package with enhanced device drivers that increase the efficiency of guest Operating Systems [31, 30, 32].

**Live Migration**

Live migration is the movement of a virtual machine from a physical host to another without turning it down, without any noticeable effect from the point of view of the end user. The most significant advantage of live migration is the fact that it facilitates proactive

maintenance, e.g. if an imminent failure is suspected, the potential problem can be resolved before the disruption of service occurs [33].

**Conclusion**

Xen is one of the most mature Hypervisors, but as an open source project it will force a team to lose lots of time learning it and setting it up, or the organization has to hire some consultant to help them setting it up.

Xen specializes in easily creating and destroying Virtual Machines in a cluster, and if a capability like that is what an organization needs, then Xen is a cheapest alternative to VMware [25].

#### 4.2.1.2 VMware

VMware started in 1998 and claimed to be the first to successfully virtualize the x86 architecture commercially [34]. VMware provides two kinds of virtualization, one called VMware Workstation, which can be installed on top of Windows, Linux, and Mac OS to virtualize all different types of Operating Systems, whilst the other is called VMware ESX and can be run directly on server hardware without an underlying Operating System [25].

VMware, as other virtualization software, is capable of Live Migration, Full Virtualization, and Paravirtualization on the ESXi Hypervisor version, moreover, VMware vSphere ESXi is capable of working as a Hypervisor. However, unlike XenServer, VMware does not virtualize the ARM architecture alongside the x86 and x64 [25].

VMware vSphere Hypervisor is a free product with a built-in management tool to create and provision virtual machines, and an advanced memory management. It requires a minimum of a single socket with two core CPU, 4GiB of RAM, and a single 1GiB network adapter. Nevertheless, VMware provides a simple vSphere and vSphere with Operations Management Edition as its paid versions. These versions offer pool computing and storages across multiple physical hosts, a centralized management of multiple hosts, perform the live migration of virtual machines, and many other premium features [35].

**Conclusion**

VMware vSphere is an all around product, full of features, but expensive. However, that can be what a large company is looking for. It is easier to manage and administrate than XenServer and has a much wider user base [36, 25]. VMware is growing at a very rapid pace thanks to its support and applications from third-party vendors.

#### 4.2.1.3 KVM

KVM, or Kernel-based Virtual Machine, is a virtualization infrastructure for the Linux kernel, which was merged into the Linux kernel in 2007, empowering the Operating System to turn itself into a hypervisor [37].

KVM supported x86 architectures from the get-go but later gained x64 and ARM support. However, KVM requires a processor with hardware virtualization extensions, making it unfeasible to use old and new CPUs without it [25].

Since KVM is based on Linux kernel, our host Operating System has to be Linux, but it can virtualize many versions of Linux, Solaris, Windows, Mac OS X, and even Android. With the backing of VirtIO, KVM can provide Paravirtualization for different Operating Systems [38].

Recently, Red Hat Enterprise Linux (RHEL) 5.4 included KVM support and is dropping Xen, employing much of the talent behind KVM, and introducing friction to companies that have cloned RHEL and invested heavily in Xen [39, 40].

**Conclusion**

Even though KVM is capable and efficient; it does not have a technical advantage over any of the others virtualization tools out there who are around much longer. One of the only benefits for KVM is that it is already built-in the recent Linux releases [25].

If an organization chooses to use RHEL, then KVM might be a valid option, but they need to keep in mind that setting up KVM can be difficult.

#### 4.2.1.4 In-House Solution Conclusion

When opting for an in-house solution for an application, the market provides us tools like Xen, VMware, or even KVM, each of these with different strengths. VMware and Xen are the most capable and mature technologies, being around for more than a decade. While VMware has some free tools to use, its main advantages come from the premium version, whereas Xen is an open-source project with all the inherent benefits and sacrifices with tools like that. Finally, KVM does not have any technical advantage on the technologies above, however, it is a very efficient tool that is built-in in the recent Linux releases [25].

| | Xen | KVM | VMware |
|---|---|---|---|
| x86 | ✓ | ✓ | ✓ |
| x86_64 | ✓ | ✓ | ✓ |
| ARM | ✓ | ✓ | ✗ |

Table 4.1: Architectural comparison between different infrastructure management software

| | Xen | KVM | VMware |
|---|---|---|---|
| Linux | ✓ | ✓ | ✓ |
| Windows | ✓ | ✓ | ✓ |
| Full Virtualization | ✓ | ✓ | ✓ |
| Paravirtualization | ✓ | ✓ | ✓(ESXi Hypervisor) |
| Live Migration | ✓ | ✓ | ✓ |
| Built-in | ✗ | ✓(recent linux) | ✗ |

Table 4.2: Feature comparison between different infrastructure management software

### 4.2.2 Cloud-based Solution

When opting to deploy our application in the cloud, it will save us from a massive investment in server hardware and people to operate it. Organizations can consume computing resources as a utility, and pay it like the water or electrical bill. Cloud computing allows organizations to scale applications easily depending on the market demands. However, even though an organization might save a lot in personnel and equipment, it may have to invest in people who knows how to operate in the cloud, or give some training to their current employees.

The whole Infrastructure as a Service (IaaS) is an invaluable service. The ease of deployment, maintenance, scalability, and pay-as-you-go model make Cloud Computing a wonderful choice for an organization's application [41, 42, 43, 44].

#### 4.2.2.1 Amazon Web Services

Amazon Web Services (AWS) alongside Microsft's Azure are the leading public cloud platforms providing worldwide coverage [42, 43]. Both services offer cloud computing as a utility and let organizations create virtual machines depending on their necessity.

Among all the services provided by AWS, the most central and best-known include Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3) [45].

AWS provides their services to companies like Pinterest, Netflix, adobe, Zynga, and many others [46].

**Locations**

Regarding geographic locations, AWS is positioned in Europe (Ireland and Frankfurt), North America (North Virginia, North California, and Oregon), South America (São Paulo), and Asia Pacific (Tokyo, Seoul, Singapore, Mumbai, and Sydney) [47]. Providing all these options an organization is sure to reach every possible client efficiently.

**Virtual Machines**

Amazon provides Virtual Machines through Amazon EC2 and has many different options regarding computer specifications. EC2 ranges from 1 to 128 Virtual CPUs, 0.5GiB to 1952GiB of RAM, and then varies regarding storage and network performance. The pricing of these machines varies depending the chosen specifications and Operating System [48].

Regarding Operating Systems, Amazon provides Ubuntu Server 14.04 LTS, SUSE Linux Enterprise Server 12 SP1, Red Hat Enterprise Linux 7.2, and Microsoft Windows Server 2012.

**Databases**

With Amazon EC2 it is possible to create a virtual machine and then use it as a database, but Amazon has services which provide the same behavior but much easier to set up, operate, and scale. These services are called Amazon Relational Database Service (Amazon RDS) for relational databases, and Amazon DynamoDB for nonrelational databases [49].

Amazon RDS has available in their catalog MySQL, PostgreSQL, Oracle Database, and Microsoft SQL Server. As for Amazon DynamoDB, the only option available is to create

a table with a primary key to inject data. Each of these services provides backup plans for every need.

#### Object Storage

One of the popular services provided by Amazon is Amazon S3 which lets an organization store all kinds of files to be accessed by an application. These files can range in size from one byte to five terabytes, but the quantity is unlimited. The price for this service is calculated from how much space is utilized, starting at $0.03 per GiB calculated monthly [50].

#### Other Services

Amazon has other interesting services, and we would like to highlight the EC2 Load Balancer to share traffic between different machines, and EC2 Container Service to run and manage Docker containers.

#### Free Tier

Regarding a free tier plan, Amazon offers the best experience. In the first twelve months, an account can have a virtual machine for 750 hours per month, which can be divided into multiple machines if the sum between them is equal or less than that time. These machines are called "t2.micro", with one vCPU and one gigabyte of RAM, a not so powerful machine, but good enough for some testing. An account also has access to five gigabytes of storage from Amazon S3, 500 megabytes per month of storage from Amazon EC2 Container Registry (store and retrieve Docker images), 25 gigabytes of storage from Amazon DynamoDB (NoSQL), and more from other services [51].

#### Conclusion

Amazon presents a service that at least equals to their sales one. Solid, efficient, reliable, scalable, and even low-cost are some of the adjectives that Amazon Web Services deserves. Amazon can help an organization start as a free tier account paying nothing and then go up as much as it needs with a corresponding price.

As shown in Figure 4.2, a downside of AWS is its web platform that can be a bit overwhelming and confuse for beginners, presenting many services and even more options for each. Their documentation is good and quite extensive, sometimes perhaps even too extensive.

### 4.2.2.2 Microsoft Azure

Microsoft's Azure, like Amazon Web Services, is a cloud computing platform for building, deploying, and managing applications and services. Even tough Azure is a product by Microsoft, the producer of Microsoft Windows; it provides not only Linux but also the Linux-based container, Docker. Azure uses a specialized Operating System called Microsoft Azure to create and run its virtual machines [52].

Azure provides their services to companies like BMW, Toyota, EasyJet, NBC Sports, Heineken, and many others [53].

#### Locations

Azure wins regarding geographic areas, being positioned in North America (Toronto, Quebec, Iowa, Illinois, Virginia, Texas, West Central US, and California), South America (São Paulo),

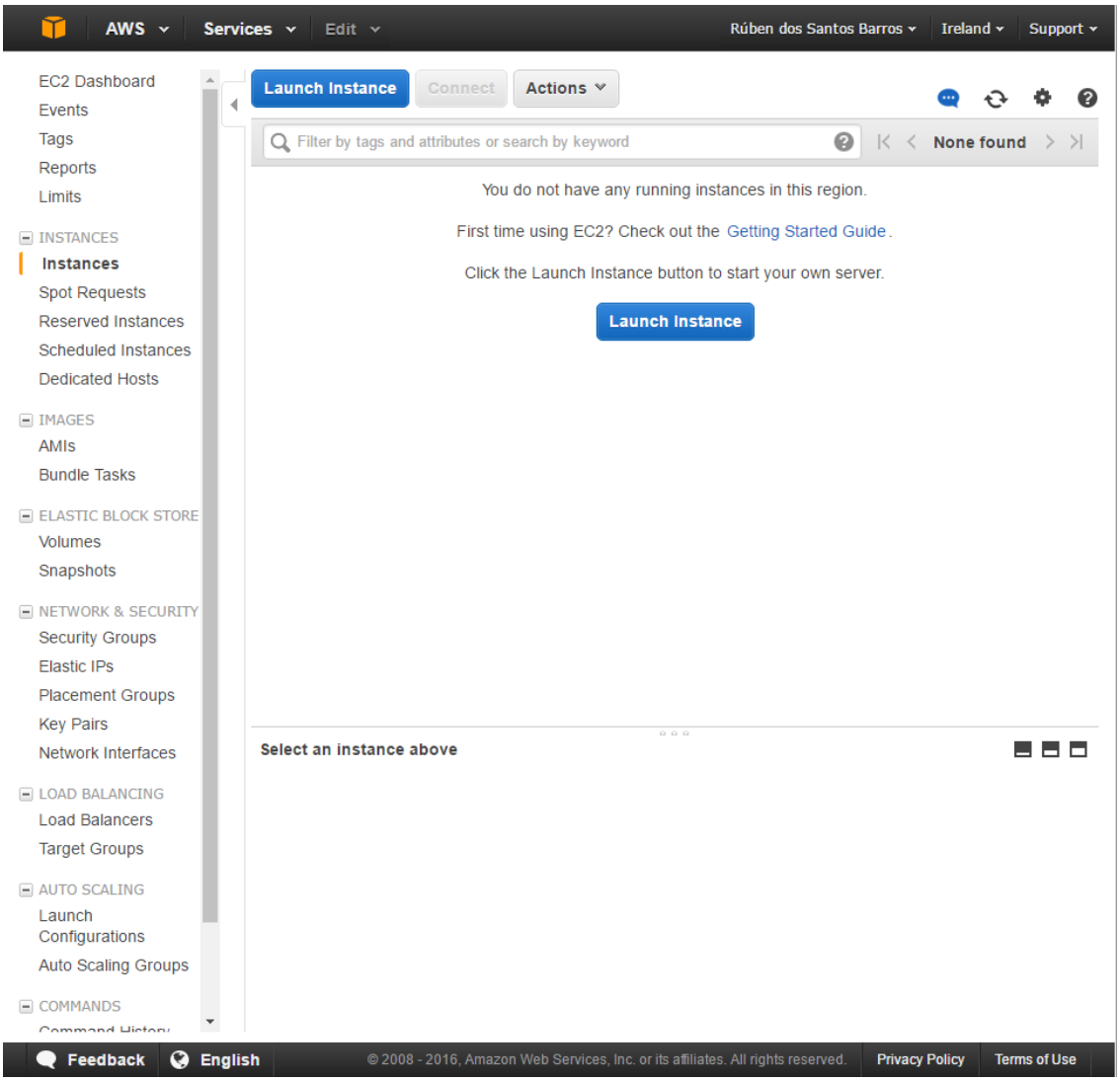



Figure 4.2: Example of Elastic Compute Cloud Dashboard

Europe (Ireland and Netherlands), Asia (Hong Kong, Singapore, Pune, Chennai, Mumbai, Shangai, Beijing, Tokyo and Osaka), and Oceania (New South Wales and Victoria) [54].

#### Virtual Machines

Azure provides several predefined virtual machines to teams as well as clean machines with different Operating Systems. As for clean machines, Azure provides CentOS, Ubuntu Server, Red Hat Enterprise Linux, openSUSE, Windows Server, Windows 10, and Windows 7. Then, Azure provides machines with preconfigured SQL Server, Oracle Database, MongoDB, JDK, Cassandra, LAMP, Visual Studio, WordPress, and much more [55].

Regarding the specifications of the virtual machines provided by Azure, they start at a quarter CPU core and 0.75 gigabytes of RAM and go to sixteen CPU cores and 112 gigabytes of RAM. These specifications are capped to much in comparison with AWS top tier machines. However, Azure will release machines with 32 CPU cores and 448 gigabytes of RAM [55].

#### Databases

As for databases, Azure provides a service to create an SQL Database and choose the Database Transaction Units (DTUs), which describe the relative capacity of the performance level, the maximum storage, ranging from two gigabytes to one terabyte. This service provides geo-replication for all plans and point-in-time restore from 7 to 35 days [56].

A smaller service grants the possibility to create MySQL Databases. These databases range capacity from 0.02 gigabytes to 10 gigabytes of space and from 4 to 40 total connections. However, only the top tier plans grant backups and geo-distribution.

#### Object Storage

The direct competitor to Amazon S3 is Azure's service Storage Accounts, to store and manage all kinds of files. The main difference is that Azure gives options regarding the replication type, providing zone-redundant storage (ZRS), locally-redundant storage (LRS), geo-redundant storage (GRS), and read-access geo-redundant storage (RA-GRS), and the access tier, Cool to store less frequently accessed data, and Hot for frequently accessed data. The chosen options will influence the paid price per gigabyte, ranging from $0.01 for LRS Cool to $0.061 for RA-GRS Hot [57].

#### Other Services

Like AWS, Azure also offers the possibility to create load balancers and define if the machines are at Azure's servers or in a different place.

#### Free Tier

Regarding the free tier plan, Microsoft offers $200 in Azure budget to try their service, in any way a user sees fit to meet its needs. Besides that, Azure offers some free services, like App Service, to quickly build and host web and mobile apps, Azure IoT Hub, which lets us exchange up to 3,000 messages and control up to 10 Internet of Things devices, and much more [58].

#### Conclusion

Microsoft's presents worthy concurrent to Amazon Web Services in the form of Azure. As seen in Figure 4.3, its web platform is much more user-friendly to those who are used to Microsoft Windows and has a proper documentation to support it. However, even though its user friendliness, Azure may be trickier and hard to find and change some important

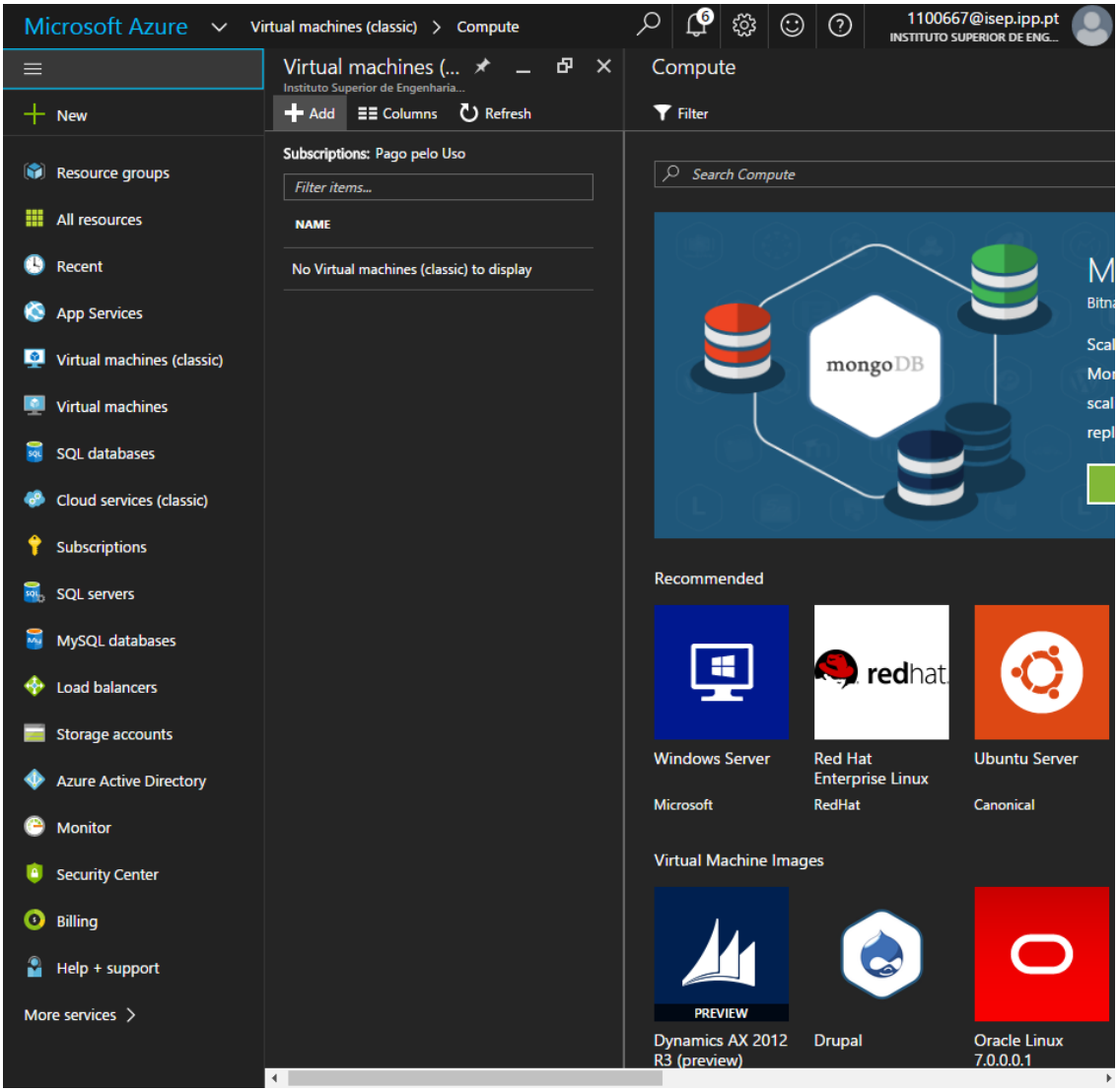



Figure 4.3: Example of Virtual Machines Dashboard

configurations. Another downside is its performance, which falls a little short to AWS while their prices are a bit higher.

#### 4.2.2.3 Google Cloud Platform

Google Cloud Platform (Google CP), is a cloud computing platform much like AWS and Azure. Google CP provides its hosting on the same infrastructure in which Google provides end-user products like Google Search and YouTube. Google CP is a part of a suite of enterprise solutions from Google for Work and provides a set of modular cloud-based services like hosting and computing, cloud storage, data storage, translation APIs and prediction APIs [59].

Google CP provides their services to companies like Coca-Cola, Spotify, Snapchat, and many others [60].

### Locations

Regarding geographic locations, Google CP is positioned in North America (Iowa, South Carolina, and Oregon), Europe (Belgium), and Asia (Taiwan) [61].

### Virtual Machines

Google's rival to Amazon EC2 is called Google Compute Engine which provides virtual machine instances. Google CPs specifications are ranging from a shared virtual CPU with 0.6 gigabytes of RAM, to 32 core CPU with 208 gigabytes of RAM. Regarding the available Operating Systems, Google CPs provides Debian, CentOS, FreeBSD, openSUSE, SUSE, Red Hat Enterprise Linux, and Windows Server [62].

Google provides a billing by the minute which is different from the usual hourly rates offered by AWS and Azure [63].

### Databases

On the topic of relational databases, Google provides Cloud SQL, a fully managed database service on top of MySQL. This service released its second generation, which Google states, is seven times faster and will scale up to twenty times of its capacity. The pricing for this service is calculated in how many instances are running per hour ($0.015 - $4.024 depending on the RAM), and per gigabyte ($0.17) [64].

As for the non-relational counterpart, Google offers Cloud Bigtable and Cloud Datastore. Regarding Cloud Bigtable, Google's Big Data database service is the same database that powers services like Search, Analytics, Maps, and Gmail. It is designed to handle massive workloads at consistent low latency and high throughput, being an excellent choice for both operational and analytical applications like Internet of Things, user analytics, and financial data analysis [65]. While Cloud Datastore, is a highly scalable NoSQL database for applications. Cloud Datastore automatically handles sharding and replication, depending on the application load [66, 67].

### Object Storage

Regarding Object Storage, Google CP provides a service called Cloud Storage to allow organizations to store data. Cloud Storage tags its price at $0.026 per gigabyte per month [68].

Google does not inform about maximum storage an account can get. However, they allow going up until petabytes, which is more than all the other cloud providers.

### Other Services

Google gave in to Docker success and included services like Container Engine, which is a type of Google Compute Engine but is meant to run a Docker container, and Container Registry to store and manage Docker images on Google CP [69].

Google also provides a service called Cloud Load Balancing, which can load balance between all Google's data centers, and App Engine to build and scale web applications and mobile backends.

### Conclusion

Google looks a bit behind in relation to AWS and Azure, and their prices are not top notch, even though having pay per minute for Google Compute Engine, which can be a great plus for certain organizations [42, 43].

Regarding the provided services, Google loses in quantity and, in a significant portion, quality. However, Google Big Data services are particularly strong, thanks to Google proficiency in that field. Another strong aspect is Google's Network which is quite fast and powerful [70].

In sum, Google excels at Big Data operations and their price plans might work with some projects, other than that, services like AWS and Azure are much more complete.

#### 4.2.2.4 DigitalOcean

DigitalOcean, founded on 2011 in New York, provides virtual servers and cloud storage. DigitalOcean focus is different than cloud platforms like AWS and Azure, in a way that their main motivation is the developers' needs, instead of an everything-to-all-people approach [71].

DigitalOcean provides their services to companies like HP, Docker, Atlassian, jQuery, Red Hat, and many others [72].

**Locations**

Unlike other cloud platforms, DigitalOcean worldwide presence is not their strong suit, but nonetheless, they are located in North America (San Francisco, New York, and Toronto), Europe (London, Amsterdam, Frankfurt), and Asia (Bangalore and Singapore) [73].

**Virtual Machines**

DigitalOcean names their virtual machines Droplets, and these Droplets have specifications ranging from one core CPU, half a gigabyte of RAM, and 20 gigabytes of SSD storage, going to 20 core CPU, 64 gigabytes of RAM, and 640 gigabytes of SSD storage. Regarding the Operating Systems, DigitalOcean provides several version of Ubuntu, FreeBSD, Fedora, Debian, CoreOS, and CentOS.

Furthermore, DigitalOcean has Droplets with the installed dependencies for applications like Cassandra, Django, ELK, GitLab, LAMP, MongoDB, Node.js, Ruby on Rails, and WordPress [74].

**Object Storage**

Recently, DigitalOcean presented Block Storage, which when creating a Droplet, provides an option to attach SSD space ranging from one gigabyte to sixteen terabytes, increasing the monthly price in $0.10 per gigabyte. DigitalOcean provides live resize for the Block Storage if there is a need to adjust space and there is even the possibility to move these Blocks between Droplets [75].

**Conclusion**

DigitalOcean is not an all-around product but is outstanding at what it does. Since network speed is one gigabyte per second and SSD storage is the default for DigitalOcean, their Droplets tend to outstand the ones that AWS provides with the same price. DigitalOcean also provides a pretty straightforward and fixed price plan [71].

A possible downside when opting for DigitalOcean is that it is missing services like load balancing and hosted databases. Since DigitalOcean only offers UNIX based Operating Systems, the user must be capable of managing and configuring it.

#### 4.2.2.5 Cloud-based Solution Conclusion

When opting for a cloud-based solution for an application, we can choose from Amazon Web Services, Microsoft Azure, Google Cloud Platform, or even DigitalOcean, each of these with different strengths and purposes. While AWS, Azure, and Google CP are the leading cloud providers [43, 42] with the most variety of Use Cases, services, server locations, and prices, DigitalOcean tries to differentiate itself with its user-friendly interface while aiming for a more packaged solution for small applications [71].

### 4.2.3 Preconditions identified

We identified a set of requirements to be met when adopting each infrastructure solution. Regarding an in-house solution, we identified that some prior knowledge about infrastructure management would greatly increase the adopting of the referred technologies. Whereas for a cloud-based solution, we found that knowing and testing an application would help to define the machine specifications to prevent oversubscriptions.

| | AWS | Azure | Google CP | DigitalOcean |
|---|---|---|---|---|
| Europe | ✓* 2 | ✓* 2 | ✓ | ✓* 3 |
| Asia | ✓* 4 | ✓* 9 | ✓ | ✓* 2 |
| North America | ✓* 3 | ✓* 8 | ✓* 4 | ✓* 3 |
| South America | ✓ | ✓ | ✗ | ✗ |
| Africa | ✗ | ✗ | ✗ | ✗ |
| Oceania | ✓ | ✓* 2 | ✗ | ✗ |

Table 4.3: Geographic comparison between different cloud providers

| | AWS | Azure | Google CP | DigitalOcean |
|---|---|---|---|---|
| PostgreSQL | ✓ | ✗ | ✗ | ✗ |
| SQL Server | ✓ | ✓ | ✗ | ✗ |
| NoSQL Service | DynamoDB | ✓ | Cloud BigTable | ✗ |
| Container Service | ✓ | ✓ | ✓ | ✗ |
| Function as a Service (FaaS) | ✓ | ✓ | ✓ | ✗ |
| Load Balancing Service | ✓ | ✓ | ✓ | ✗ |

Table 4.4: Service comparison between different cloud providers

## 4.3 Automation

In the nineteenth century appeared new manufacturing processes such as the replacement of hand production methods to machines, which radically improved efficiency, consistency, and productivity. This was called Industrial Revolution.

| | AWS | Azure | Google CP | DigitalOcean |
|---|---|---|---|---|
| Free VM | 1 year | 60 minutes daily | ✗ | ✗ |
| Base Linux VM | $10.08 | $14.40 | $4.03 | $10.80 |
| Base Windows VM | $13.68 | $12.96 | $18.43 | ✗ |
| Object Storage | $0.03 per GiB | $0.08 per GiB | $0.04 per GiB | ✗ |

Table 4.5: Price comparison between different cloud providers

Nowadays, a new revolution is happening, and it is called DevOps, which consists mainly in the Automation of processes. This evolution is not really about technology but more about trust and collaboration. We need to trust that a computer can perform functions otherwise done by humans, and collaborate better and faster with each other, so that we can achieve the same improvements that we got after the so-called Industrial Revolution [76, 77].

**Deployment Automation**

Continuous Delivery embraces automated deployments in various stages of the software delivery process and identifies manual deployments as one of the common release anti-Patterns. This practice makes use of computers strength to make releasing software a repeatable and reliable push-button activity [78].

Since time-to-market is important if not vital, a combination of Agile development practices, a sound suite of automated tests, and deployment automation allows the time minimization to deliver new features [78].

Quoting Jez Humble and Dave Farley's book "Continuous Delivery": "A deployment pipeline is, in essence, an automated implementation of your application's build, deploy, test, and release process.". This pipeline concept is an analogy to Lean Manufacturing, where a production line stops whenever the process detects a defect along its way, and the team takes corrective measures. The typical deployment pipeline initiates whenever a developer commits code to a software repository inside the version control system such as Git or Subversion. When the build automation server observes a change in the repository, it triggers a sequence of stages which exercise a build from different angles via automated testing. During these stages, the application is compiled, built, executed, and tested through Unit Tests, Integration Tests, End-to-End Tests, Capacity and Load Tests. After having successfully passed all these phases, the production environment is updated with the new version of the application [79].

Employing an Automated Deployment lets organizations change their applications' environments through script configuration, meaning that if arises the need to switch to a different infrastructure provider, the overhead of deploying to that target is minimal. The Deployment becomes much less error-prone since manual deployment involves human interaction and quoting Murphy's Law "If anything can go wrong, it will go wrong.". This process will also accelerate deployments through repeatability and reliability. Furthermore, the knowledge of how to release the application is captured in the system instead of an individual's brain. Ultimately, teams will have more time to develop software, and consequently, applications can be upgraded and released more frequently [80].

By enforcing an automated deployment process and treating infrastructure as code organizations make sure to end up with deployable code and working environments at the end of each iteration [78].

From a business perspective, deployment automation allows organizations to get changes, whether planned or unplanned, into production reliably and quickly. This process allows them to gather valuable user feedback as early as possible and develop and maintain software that is lean, and that offers features that their users like to use, thereby avoiding software bloat [78].

#### Continuous Integration

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily - leading to multiple integrations per day. Each integration is verified by an automated build, which includes testing the software to detect integration errors as quickly as possible. Many teams find that this approach leads to significantly reduced integration problems and allows a team to develop cohesive software more rapidly [81].

### 4.3.1 Chef

Chef, a configuration management tool, was released in 2009 by a company with the same name [82]. It was designed with a DevOps culture in mind, and in addition to automating the provisioning of server infrastructure, Chef can automate the provisioning of runtime environments, applications, and containers. With the release of Chef 11, the product was completely rewritten using Erlang programming language, which took the team to rename the core server API to Erchef. The team also replaced CouchDB for PostgresSQL. These changes resulted in a memory consumption reduced by a factor of 10 and thus allowed Chef to manage easily 10,000 clients, something like four times what Chef 10 could handle [83].

Even though Chef is written in Erlang, the configuration files are coded in Ruby. These configuration files are called recipes, which can belong to a cookbook with many recipes [84].

Chef provides their services to companies like Facebook, Nordstrom, Standard Bank, and many others [85, 86].

#### Operating Systems

Initially, Chef was designed to support Linux, but support for Windows has increased dramatically in recent years and is now supporting PowerShell DSC, IIS, and SQL Server. Chef even supports 64-bit Windows, meaning that Chef will be able to manage the upcoming Windows Nano Server [83].

#### Cloud Service Support

Chef integrates with major cloud providers such as Amazon Web Services, Microsoft Azure, Google Cloud Platform, and others [87].

#### Chef DK

When acquiring Chef, we get Chef DK from the get-go, which is a toolkit that contains some tools developed by the Chef community, including built-in testing tools like RuboCop and Foodcritic, the unit testing framework known as ChefSpec, and Test Kitchen, an integration tool for testing coded infrastructure. This software gives new users a streamlined workflow [88].

#### Chef Server and Chef Client

Chef uses the server-client model, which means the Chef Server works with the Chef Client to apply configurations to managed nodes. This process can be configured in a way that Chef Server sends a set of instructions (recipe) to a load balancer while a web server or a database server receive different sets of instructions. This process happens when the Chef Client runs, as it queries the server for the latest set of recipes, which are executed orderly so that consistent and repeatable results are seen. This process, by default, happens every 30 minutes, meaning it is a pull method, one not as good as a direct push method [89].

The Chef Client packs a tool called Ohai, which detects platform details, network usage, memory usage, kernel data, CPU data, and much more, giving Chef the capability to search and report on node configurations across the network.

**Chef Supermarket**

Chef provides an open source community site allowing users to browse, download, and share cookbooks [90].

**Setup**

To start using Chef, one must first deploy a Chef Server on Ubuntu 12.04, Ubuntu 14.04, or Red Hat Enterprise Linux from version 5 through version 7. Then we need to install the Chef Client in the desired nodes, which can have Debian, FreeBSD, Mac OS, Red Hat Enterprise Linux, Ubuntu, or Windows as their Operating System.

**Free Plan and Pricing**

Chef is an open source, meaning that it is available for free, but still has some premium plans for bigger projects which require more automation and support.

The free version of Chef comes with an easy to install Chef Server to manage more than 10,000 nodes, free support for eight hours during the work days for a month, and access to Chef Supermarket. This plan offers features like Chef Backend, to ensure Chef service is uninterrupted, Reporting, to capture and visualize the state of the nodes running the Chef Client, Management Console, a Web-based management console, and Analytics Platform, to get visibility into Chef Server and verify its compliance. These features are only available for systems with 25 nodes or less.

Regarding premium plans, Chef offers Hosted Chef, for a minimum of twenty nodes at an annual price of $72 per each node, offering a hosted Chef Server and high availability. The other available plan is called Chef Automate and comes with an annual price of $137 per each node, offering tools for workflow, compliance, visibility, and high availability, as well as a 24x7 support window [91].

**Conclusion**

Chef is highly mature and works at massive scale due to its adoption by Facebook, who also has contributed [86]. It resonates best for those who are familiar with version control tools, and unit and integration tests. The way that cookbooks and recipes can leverage the powerfulness of Ruby is a major advantage.

Even though Chef is a robust tool, the lacking of a proper push command, forcing a team to wait for the automatic pull is a significant disadvantage. The learning curve for Chef is also a major obstacle, and it might even include the learning of the Ruby programming language [84].

### 4.3.2 Puppet

In 2005 Puppet Labs released Puppet, a configuration management tool written in Ruby [92]. Puppet, like Chef, is another declarative tool, but instead of cookbooks and recipes, the configuration files are called Puppet Manifest. These can be written in Puppet's Declarative Language or Ruby DSL (domain-specific language) [84].

Puppet has enjoyed significant first-mover advantages about Chef, and though both have been market leaders since the early days of IT Automation, Puppet has a longer commercial track record and a larger install base [93].

Puppet provides their services to companies like Google, Intel, NASA, Twitter, Uber, Sony, and many others [94].

#### Service Support

Like Chef, Puppet is a renowned product, and as such it integrates smoothly with Amazon Web Services, Google Cloud Platform, Microsoft Azure, and even with private cloud software Open Stack. Furthermore, Puppet also mingles with Docker, to install, configure and manage containers, Atlassian, to collaborate in real time, Cisco, to apply automation practices to Cisco networks, Jenkins, to continuously deliver better software, and Splunk, to gain real-time infrastructure visibility and insights. All of these come in the form of Puppet modules. The Puppet Forge is a repository for these modules, which are written by the Puppet team and by the Puppet user community. These modules solve a wide variety of problems, and using them is encouraged as it can save a lot of time and effort [95].

#### Facter

Puppet discovers the system information through an intrinsic utility called Facter and compiles the Puppet manifests into a system-specific Catalog containing resources and resource dependencies, which are applied to the target systems. Any actions taken by Puppet are then reported [96].

#### Hiera

Puppet also comes with Hiera, a convenient hierarchical key/value store. This store allows users to override and set site-specific settings to Puppet modules without having to fork or modify them [97].

#### Marionette Collective

The Marionette Collective, also known as MCollective, is a Framework for building server orchestration or parallel job execution systems, allowing users to execute administrative tasks on clusters of servers programmatically [98].

#### Code Manager

The recent addition to Puppet, called Code Manager, uses r10k and the file sync service to stage, commit, and sync code, automatically managing the environment and modules. Code Manager starts by creating a control repository with branches for each environment, such as production, development, or testing. Then Puppetfiles are written for each environment, specifying which modules to install in each environment. Finally, when code is pushed to the control repository, Puppet syncs the code to the masters, so that all servers start running the new code at the same time, without interrupting agent runs [99].

### Setup

Regarding the setup process of Puppet, one must first choose the Puppet Enterprise Master to manage the Agents. These can be Red Hat Enterprise Linux (RHEL), CentOS, Ubuntu, or SUSE Linux Enterprise Server (SLES). About Puppet Enterprise Agents, the offer is much wider going from RHEL, CentOS, Fedora, Windows, Ubuntu, Debian, SLES, Solaris, Mac OS, and many others [100].

### Master/Agent

Puppet runs in an agent/master (client/server) architecture, where a Puppet master controls configuration information and managed agent nodes request only their configuration Catalogs. The Puppet agent runs as a background service and periodically sends facts to the Puppet master and requests a Catalog. Once received, the agent applies it by checking each resource the Catalog describes. If it finds any resources that are not in their desired state, it makes any changes necessary to correct them. After applying the Catalog, the agent submits a report to the Puppet master [101].

Puppet can make use of push commands through modules like MCollective, but it is not how Puppet was designed to operate [102].

### Free Plan and Pricing

Puppet provides a free plan to install Puppet Enterprise on ten nodes. This plan includes access to all product updates and almost every module available.

Apart from that, Puppet offers a standard package for a starting price of $120 per node, which includes support for bug fixing, a private knowledge base, and more. Finally, Puppet also offers a Premium support package with 24-7 and phone support for priority issues and free training. These two plans can only be purchased through contacting the company with some information about an organization means and requirements, and the final price will be presented accordingly [103, 104].

### Conclusion

Chef and Puppet are continuously growing and expanding their automation platforms in response to the needs of the DevOps community. Puppet Code Manager is a feature that represents that demand, which helps streamline the continuous integration/delivery pipeline [93].

Like Chef, Puppet has a robust community that continuously supports it through feedback and modules, which are available for everyone. MCollective is the major difference to Chef if an organization values push commands [84, 105].

Nevertheless, like Chef, a team needs to know or learn Ruby in addition to Puppet DSL [84].

Regarding price comparisons with Chef, Puppet limits their free usage with ten nodes but full features for all. Whereas Chef does not restrict the number of nodes, however, most features only hold out for 25 nodes. Another note is that although Chef paid edition starts at $72 per node, it requires a minimum of 20 nodes [104, 91].

### 4.3.3 Ansible

Ansible is a free-software platform, developed by Michael DeHann in 2012, for configuring and managing computers which combines multi-nodes software deployment, ad-hoc task execution, and configuration management [106]. Ansible is written in Python, but its modules can be written in any programming language, however, to express descriptions of systems, Ansible uses YAML [84].

Ansible's configuration, deployment, and orchestration are done through Playbooks. They can be used to manage configurations of and deployments to remote machines, and besides that, they can sequence multi-tier rollouts involving rolling updates, and can delegate actions to other hosts, interacting with monitoring servers and load balancers along the way. Playbooks are designed to be human-readable and are developed in YAML [107].

Ansible grants the safety of repeatable runs of tasks through its idempotency [108]. It uses "Facts", which is system and environment information it gathers, as "context," before running Tasks. Ansible uses these facts to check state and see if it needs to change anything to get the desired outcome, which makes it safe to run Ansible Tasks against a server over and over again [109].

#### Setup

Since Ansible is a Python project, the best way to get it is via pip, the Python package manager. However, it is also available in Ubuntu through APT, and in CentOS and Red Hat Enterprise Linux through Yum. Finally, Ansible can even be downloaded and compiled from its Git repository.

Ansible offers a minimal management service, using SSH to manage Unix nodes and PowerShell to work with Windows servers, without an agent installed on nodes. The only requirement for Ansible to administer a UNIX node is that it must have Python 2.4 installed, whereas to manage a Windows node it requires PowerShell 3.0 or later [110].

#### Interaction

Like Chef or Puppet, Ansible distinguishes two types of systems, a controller machine, and its nodes. The controller machine is where orchestration begins, as it controls the nodes over SSH. This machine must have an inventory file with IPs or domains of its nodes and an Ansible Playbook describing the state of a server or service. The Playbook will deploy Ansible modules that handle configuration to servers and remove them once the service is running [111].

With this type of interaction, when Ansible is not managing nodes, it does not consume resources because no agents or programs are executing Ansible in the background, decreasing the dependencies of a server.

#### Cloud Support

Since Ansible connects to nodes through SSH, it can act on any in-house solution or any cloud platform like Amazon Web Services, DigitalOcean, Google Cloud Platform, Microsoft Azure, and much more [112].

#### Ansible Tower

Ansible Tower is a web-based solution designed to be the hub for all Ansible automation tasks. Tower allows us to control access to who can access what and provides a graphical inventory to manage or sync with a wide variety of cloud sources [113, 114].

**Free Plan and Pricing**

Ansible is a free product available on GitHub for anyone to use it. However, Ansible Tower, the Web-UI version is provided in three different versions: Self-Support, which includes maintenance and upgrades but no support whatsoever, Standard, which includes support eight hours during weekdays, and Premium, with 24-7 support. The Self-Support version has an option for less than 100 nodes and costs $5,000 a year and another choice for less than 250 nodes costing $10,000 a year, more than that are only available in Standard and Premium. Standard costs $10,000 a year for less than 100 nodes, whereas the Premium version is tagged at $14,000 a year for less than 100 nodes. To get the pricing for more than 100 nodes for Standard and Premium, one must reach Ansible to negotiate [115].

**Conclusion**

The big drawback when using Ansible is its feedback. Ansible does not guarantee the successful running of a playbook in one or more nodes, and it does not provide a detailed report of what happened in the nodes. With this, a team might not be sure if everything went as planned in all machines, and if not, which machines got problems and why [116, 117].

Nevertheless, Ansible is still a powerful and lightweight tool making it perfect for running simple repetitive tasks such as restarting services, copying files, or running simple shell commands. As it does not need an agent on nodes, modules can be written in any programming language and the simplicity concerning its usage [116].

If the intended purpose of an organization is to manage a project with a just a few machines and wants to optimize the machines' performance, then Ansible might be a better option than more robust ones like Chef or Puppet.

Since Ansible is a somewhat recent project and thanks to its lack of feedback it has not groped some big projects, however, its powerfulness is more than sufficient to captivate the interest of several teams with several small projects.

### 4.3.4 Atlas

After releasing open-source tools like Vagrant, HashiCorp by Mitchell Hashimoto, has released a commercial platform to unite some of these tools called Atlas. Atlas integrates HashiCorp's Vagrant, Packer, Terraform, and Consul to create a version control system for infrastructure management, believing that it provides benefits like transparency, auditability, and collaboration in the same way as application code version control [118, 119].

HashiCorp describes Atlas as "a unified dashboard and workflow for developing, deploying, and maintaining applications on any public, private, or hybrid infrastructure" promoting automation, audit, and collaboration on infrastructure changes across the modern datacenter [119].

Even though Atlas is a HashiCorp product that integrates some of their tools, it can even handle software like Chef, Jenkins, and Docker [119].

**Vagrant**

As stated before, Vagrant provides virtual machine box builds for the creation of lightweight, reproducible, and portable development environments. The Atlas integration allows the creation, hosting, and distribution of Vagrant boxes throughout a team [120].

**Packer**

Packer, as an open-source tool, creates machine images or containers for platforms like Amazon EC2, DigitalOcean, Google CP, VirtualBox, VMWare, and Docker from a single source configuration. Packer can be run on Atlas to create and store versioned machine images to be used on those platforms [121].

**Terraform**

Through a common configuration, Terraform automates infrastructure provisioning and management across several cloud providers. The Atlas integration with Terraform allows the state to be stored in a single remote location, provisioning to be audited, and enables collaborative review and application of changes [122].

**Consul**

Consul provides service discovery, configuration via a highly available key/value store, and a set of orchestration primitives. Atlas integration enables cluster state visualization, node monitoring, and alerting on applications and associated infrastructure [123].

**Interaction**

The major difference between Atlas and runtime configuration management tools like Chef, Ansible, and Puppet is that Atlas embraces immutable infrastructure and build-time configuration. This way Atlas builds a deployable artifact such as an Amazon Machine Image, a Google Cloud Engine image, a Docker container and only then provisions a host using this fully configured artifact. With other tools functioning with runtime configuration, a host is provisioned, started, and then configured. The build-time configuration embraced by Atlas leads to faster deploys, identical configurations, and a more scalable design [124].

Organizations can use the complete HashiCorp toolset to deploy immutable infrastructure continuously. The workflow starts when a developer, who is working locally in a Vagrant environment, pushes the updated application code to a source control system such as GitHub. From here, Atlas starts a process in which Packer ingests the application code and merges it with its required dependencies stated directly in the Packer file, or some provisioning script from Puppet, Chef, Ansible, or even Shell, to create a deployable artifact. The Packer-built artifact is stored in Atlas' artifact registry, which Terraform references and deploys an instance using this artifact. An operator is notified of a pending change, which shows the difference in the artifact version for a deployed instance. The operator then reviews and confirms this infrastructure change which triggers Terraform to apply it. If a Consul agent was configured in the Packer build stage, it connects with Atlas and adds the instance to the Consul service registry. This agent will check the health of these nodes, and if something bad happens, it triggers alerts through email or applications such as Slack or PagerDuty [125].

**Pricing**

Atlas pricing is not disclosed, and the only way to know is through contact with HashiCorp giving some information about the project in hands. Nevertheless, Atlas provides a Standard

and a Private option which differs only with phone support and private installation options for the Private choice.

**Conclusion**

Atlas tries to solve the inherent problem of operations, which is moving an application from code in development to running in production through a safe, repeatable, and auditable process. It is not a tool, but nevertheless can compete with tools such as Chef, Ansible, or Puppet, or even work alongside them [124].

Atlas is one of the most immature tools regarding configuration management, so it does not have a very broad user base to support it. The decision to not disclose the pricing for the service and the lack of a free plan might hinder the interest of some teams.

Nevertheless, Atlas includes some great tools like Vagrant, Packer, and Terraform that are known, used, and contributed by the community. The way that Atlas uses Packer to implement Immutable Infrastructure is a big advantage comparing to other projects. However, all of these helpful tools can be integrated with other configuration management tools, and even those tools can be used without Atlas, being the dashboard with shared information the sovereign feature around Atlas.

### 4.3.5 Deployment Automation Conclusion

Quoting Jez Humble and Dave Farley's book "Continuous Delivery": "A deployment pipeline is, in essence, an automated implementation of your application's build, deploy, test, and release process." [79]. With that process in mind, the market provides us tools like Chef, Puppet, Ansible, and Atlas to manage it. While Puppet is the most mature tool, Chef has been growing exponentially due to its adoption by Facebook. Both of these technologies have a robust community which supports and contributes to its sustained growth. Whereas Ansible is a much more lightweight tool that supports scripting in the most variety of programming languages [84]. Finally, Atlas, which, although immature, is the only tool that encourages and supports from the get-go an immutable infrastructure.

### 4.3.6 Preconditions identified

We concluded that the knowledge about testing and testing tools is strongly advised since an effective deployment depends on a battery of successful tests, and a team should be acquainted with agile development processes since these tools use a very similar approach. Fulfilling these requirements would greatly increase the adoption process of a Deployment Automation tool.

## 4.4 Virtualization and Provisioning

The practice called Virtualization is especially useful in complex environments, like when an organization wants to have similar environments for Development, Quality Assurance, and Production. Whereas the practice called Provisioning is about making a server ready for service through activities like selecting an image to install (physical server) or an image to run (virtual server), installing and configuring middleware and applications, and configuring

| | Chef | Puppet | Ansible | Atlas |
|---|---|---|---|---|
| AWS, Azure, Google CP | ✓ | ✓ | ✓ | ✓ |
| Script Language | Ruby | Puppet DL Ruby DSL | YAML | K/V (json) |
| Linux | ✓ | ✓ | ✓ | ✓ |
| Windows | ✓ | ✓ | ✓(PS 3.0) | ✗ |
| Node Agent | ✓ | ✓ | ✗ | ✗ |
| Middleman Server | ✓ | ✓ | ✗ | ✓ |
| Push Commands | ✓(not natively) | ✓(MCollective) | ✓ | ✓ |
| Immutable Infrastructure | ✗ | ✗ | ✗ | ✓ |
| Free Plan | Unlimited nodes Services limited to 25 nodes | Limited to 10 nodes | ✓ | ✗ |
| Basic Plan | $72 per node (min. of 20 nodes) | $120 per node | Ansible Tower $5,000 year for 100 nodes | Price undisclosed |

Table 4.6: Base comparison between different configuration management tools

the network. This process is something that may need to be frequently done, so automating it will pay off over the long haul [126].

In sum, Virtualization will help setting up and testing anything in advance before running applications in productions, while Provisioning will configure a suitable environment with every requirement met to run an application [127, 126].

### 4.4.1 Docker

Docker, released in 2013 as an open-source project, aims to automate the deployment of applications inside containers. Quoting the Docker web pages: "Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries – anything that can be installed on a server. This guarantees that the software will always run the same, regardless of its environment." [128, 129].

Docker shares the same kernel as the Operating System, allowing the containers to start instantly and use less RAM. However, since Docker uses the kernel, when ran on Windows and Mac OS it needs a tool like boot2docker to simulate this kernel. Although, since Windows 10 Hyper-V, Docker can just start without any assistance [130, 129].

The main difference between a provisioned virtual machine and a provisioned container is that the virtual machine includes an entire guest operating system, which will consume more storage space and RAM, while containers share the same kernel. In addition to that, Docker is also safer and easier to deploy and use. Besides, it allows anyone to work on the same project with the same settings, regardless of the local host environment [131].

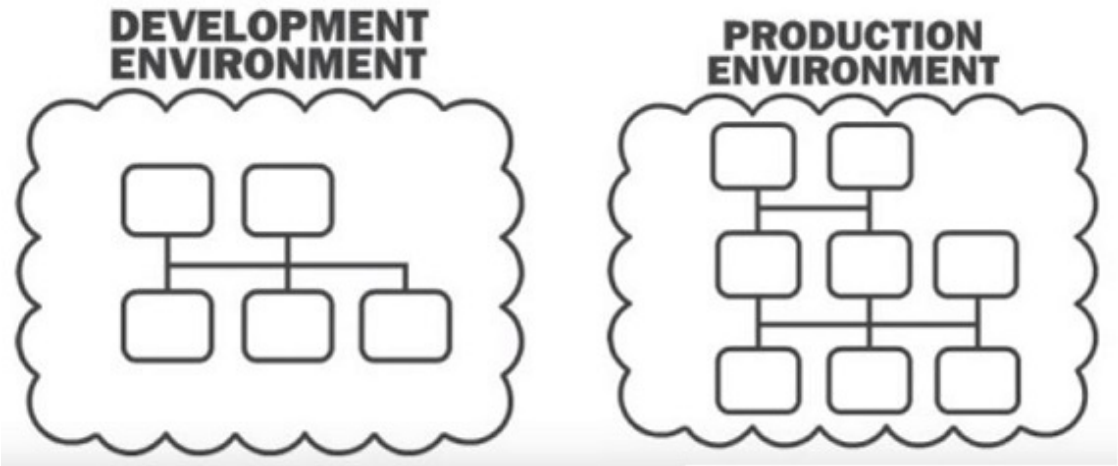



Figure 4.4: There should not be any differences between development and production environments

The main cloud providers, like AWS and Azure, already recognized Docker's value and provide services to store, run, and manage Docker images [131, 132].

**Conclusion**

Besides Docker being a truly lightweight way to run several layered applications on a single server, one of its benefits is the possibility to provision a server configuration and give a team the opportunity to develop and test on top of it. On top of that, Docker popularity reached the top contenders for DevOps automation like Chef, Puppet, Ansible, and many others, in addition to several cloud providers who are already prepared to host Dockered applications [131, 133].

Unfortunately, Docker is not perfect, and has its downsides, like it can not run Windows applications and sharing the kernel with the Operating System leaves the possibility to broke the host system.

### 4.4.2 Vagrant

Vagrant purpose is to create and configure virtual development environments. This means that Vagrant works on a higher-level wrapper for software such as VirtualBox, VMware, KVM and Linux Containers (Docker). Quoting Vagrant web pages "With an easy-to-use workflow and focus on automation, Vagrant lowers development environment setup time, increases development/production parity, and makes the "works on my machine" excuse a relic of the past." [120].

Vagrant is not a provisioning tool, but can work as such, since it is capable of launching a virtual machine or a container and run commands and scripts on them. Vagrant abstracts advanced configuration like networking and syncing folders no matter if VirtualBox, VMware or another tool was chosen [134, 135].

About host environment, Vagrant can be installed on Linux, Windows, and Mac OS, and in all cases, the workflow is the same. However, unlike Docker, Vagrant can start Operating

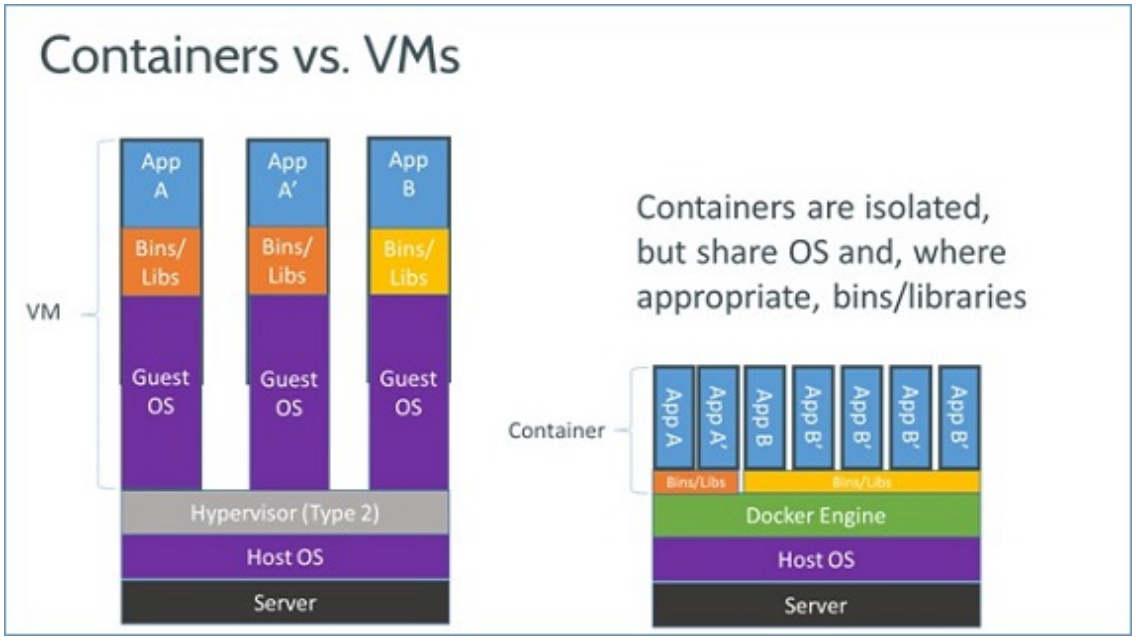



Figure 4.5: Architecture for Virtual Machines and Docker Containers [129]

Systems like Windows, which might be a plus for some projects. These systems come in the form of a "box", which can be used by anyone on any platform. The easiest way to use a box is to add it from the Vagrant catalog. Nevertheless, Hashicorp offers a tool called Packer to create machine and container images [135, 120].

**Docker**

Since Vagrant is not a direct Docker competitor, it recently added the possibility of spinning up containers and provides a good workflow for developing Dockerfiles. Vagrant also has the capacity to launch a Linux Virtual Machine on Windows and Mac OS to provide its kernel, or skip this part on a Linux host machine [136].

**VirtualBox/VMware**

Regarding VirtualBox is an open-source hypervisor developed by Oracle Corporation and released in 2007. VirtualBox is available for Linux, Mac OS, Windows, and Solaris, and can virtualize any version of Windows, Linux, BSD, Solaris, and limited virtualization of Mac OS. This product allows a machine to run multiple Operating Systems on a single machine, at the same time, facilitating the processes of testing, developing, demonstration, and then the deployment. When complemented with Vagrant, we can automate all processes through scripting, and use the straightforward abstract commands to help setting up configurations like network and synced folders [137, 138].

VMware provides Workstation Player to be used on Windows and Linux, and Fusion to be used on Mac OS. Both can virtualize any version of Windows and Linux, and allows a machine to run multiple Operating Systems, much like VirtualBox. When VMware is complemented with Vagrant it inherits the same benefits as VirtualBox [139].

**Automation Tools**

Vagrant becomes a great addition to automation tools like Chef, Puppet, and Ansible thanks to the possibility to create a development environment identical to the product environment, allowing operations to work with a faster feedback loop, reducing inconsistencies between development and production [135, 140].

**Conclusion**

Vagrant is a proficient tool for provisioning, even though it is not its primary focus. It is also not a rival to Docker but efficiently works alongside it. Vagrant is more abstract and general than Docker, and in the future, if a Docker competitor arises, Vagrant will run it as well [141].

In short, if an organization only develops and deploys on Linux, Docker will handle provision, test, and deploy of applications through containers. If an organization requires some abstraction, but even though knows the power behind containerization, Vagrant can handle Docker containers with almost none overhead. And finally, if an organization is not prepared for containers or needs a different Operating System than Linux, Vagrant is the most viable choice.

### 4.4.3 Provisioning and Virtualization Conclusion

Even though VirtualBox and VMware are widely used for virtualization, a software team should at least use a tool like Vagrant to handle the creation of those machines because it allows us to use it in different operating systems and through various virtualization technologies. As for Docker, it is a different and lightweight approach that, due to its high popularity, is already available in the major cloud providers as a service [131, 132]. Both these tools are capable of provision a server and provide development teams to work on a production-like system.

### 4.4.4 Preconditions identified

We found that a prior knowledge of the application and its requirements would significantly accelerate the process of creating a machine with the correct configurations for provisioning.

## 4.5 Testing

Software testing involves the execution of a software component to evaluate if it meets the requirements that guided its design and development, if it responds correctly to all kinds of inputs, and if it performs its functions within an acceptable time. The number of tests that one can create for an application might be infinite, and as a result, one must select tests that are feasible for the available time and resources [142].

Software Testing executes an application with the intent of finding problems and can be an iterative process as when one bug is fixed, others might surface. This process can provide objective independent information about the quality of software and risk of its failure to users [142].

**Test Automation**

The objective of Test Automation is to simplify as much of the testing effort as possible, through the automation of some repetitive but necessary tasks, or performing additional testing that would be difficult to do manually. When using a manual approach one might not always be effective in finding certain classes of defects, whereas an automatic approach can be run quickly and repeatedly, which can be cost-effective [143, 144].

Test Automation is a critical process for Continuous Delivery and Continuous Testing [145].

### 4.5.1 Unit Testing

The goal of Unit Testing is to isolate each part of the application and test if the individual parts are working correctly. These parts, or units, are small testable pieces of code like functions, classes, procedures, and interfaces. A method is created for each of these units and is tested to determine if code meets its design. Specifically, it means that when a set of inputs are given, it should return the proper values, and when invalid inputs are given, the application should handle failures gracefully [146, 147].

Unit Testing should be done as early as possible during the development phase, as it brings benefits like finding bugs at an early stage, helping in maintaining and changing the code, and simplifying the debug process [146].

#### 4.5.1.1 JUnit

JUnit first appeared in 2000 by Kent Beck and Erich Gama, while the Agile movement was in its early stages and Test Driven Development (TDD) was unknown. JUnit is an open source unit testing framework to write and run repeatable tests for Java [148].

During all of its life cycle, JUnit evolved and since its fourth release, besides the usual test ending with an assertion, it provides notations to different actions like BeforeClass, Before, After, and AfterClass. BeforeClass runs before the first method, Before runs before each test, After runs after each test, and AfterClass runs after the last test [149].

JUnit has been ported to languages like C (CUnit), C# (NUnit), C++ (CPPUnit), JavaScript (JSUnit), Microsoft .NET (NUnit), PHP (PHPUnit), Python (PyUnit), and many others.

**Conclusion**

A research survey in 2013 took 10,000 Java projects hosted on GitHub and found that it was present in 30.7% of them [150]. These numbers show the importance of Unit Testing and JUnit for Java projects.

#### 4.5.1.2 TestNG

TestNG, a testing framework for Java, was created by Cédric Beust, which was inspired by JUnit. It was designed to cover a wider range of test categories like functional, end-to-end, and integration, in addition to Unit Testing [151].

The best feature about TestNG is its annotations, which was later included in JUnit. This feature allows us to define things like how many times a test will run, how many threads will run this test, and which method will provide data to test. The Before and After annotations

are also well-though and more comprehensive since it covers suite, class, group, and method. Another feature is the ability to create groups of tests to only test one or some needed groups, and dependencies between tests [152, 149].

TestNG is also getting more updates, maintaining their code much often, and improving due to enormous feedback [149].

**Conclusion**

Since TestNG is more than just Unit Testing, some organizations are moving from JUnit when they need to write integration and end-to-end tests [149]. Like other testing tools, TestNG runs tests with Maven, Ant, Gradle, and more. It can be executed from Eclipse, IntelliJ IDEA, and some others IDE's. It is also easily integrated with other software tools like Jenkins, Sonar, and Mockito [151, 149].

A possible disadvantage is that TestNG does not create test classes before every test, meaning that we need to enclose creation of objects within some method [149].

Finally, TestNG seems more advanced in parameterize testing, dependency testing, and suite testing. It is meant for high-level testing and complex integration test and covers the entire core of JUnit functionality [153].

### 4.5.2 E2E Testing

End-to-end testing is a methodology used to test whether the flow of an application is performing as designed from the start to finish. The purpose of carrying out end-to-end tests is to identify system dependencies and to ensure that the right information is passed between various systems and system components. The entire application is tested in a real-world scenario such as communicating with the database, network, hardware, and other applications [154].

A simplified end-to-end testing of an email application might involve logging in to the application, accessing the inbox, composing an email, checking the sent items, and logging out of the application [154].

#### 4.5.2.1 Protractor

AngularJS is becoming immensely popular, and as such, Protractor appeared as an end-to-end test framework, since testing should be a mandatory process of any development workflow. Protractor provides a solid testing ground to cover the front-end of applications and can be written down as instructions for a human interacting with the application, as it is intended to run tests from a user's point of view [155].

Protractor allows testing Angular-specific elements without efforts like adding waits and sleeps to tests, as it can automatically execute the next step of a test the moment the web page finishes pending tasks [156].

**Setup**

Protractor needs Node.js to run and both can be installed through the npm packet manager. Protractor uses Jasmine test framework for its testing interface, requires Selenium Server to control browsers which requires Java Development Kit (JDK) [157].

#### Interaction

Protractor can send keyboard strokes to elements through their ng-model tag, click elements by their id, and assert model data and browser URL to ascertain the correct behavior of the application. Protractor is also capable of running the same tests one multiple browsers at the same time [158].

#### Conclusion

Protractor is a formidable testing tool for AngularJS capable of going through a complete application and find all kinds of errors. However, it might get a little tricky to setup, interact with third-party applications such as select2, datetimepicker, and modals.

### 4.5.2.2 Nightwatch

Nightwatch is a Node.js automated end-to-end framework for web-based applications and websites. It uses selenium's web driver to perform commands, assertions on DOM elements, and can run on several browsers [159].

Nightwatch has a built-in command-line test runner which can run the tests either sequentially or in parallel. Nightwatch, like TestNG, can define groups and tags to run some particular tests. Unlike Protractor, Nightwatch supports cloud testing providers like SauceLabs and BrowserStack [159].

#### Setup

Like Protractor, Nightwatch depends on Node.js, npm, and Selenium Server, which depends on Java Development Kit (JDK). Moreover, Nightwatch requires a configuration file with information like the tests source folder, output folder for reports, Selenium and browser information [160].

#### Interaction

Nightwatch is much different from Protractor in the way we write code. A test starts with a provided variable, connects actions with dots, and must be closed with the ".end()" call. Other than that, it has similar features as Protractor to set values, click buttons, assert content in the page, and functions like "before", "after", "beforeEach", and "afterWach" [161].

#### Conclusion

Since Protractor is attached to AngularJS, Nightwatch provides a way to test applications and web pages with plain JavaScript. The support for cloud testing providers and continuous integration tools like Jenkins are a great plus for a DevOps workflow.

## 4.5.3 Testing Conclusion

Since Testing software is an almost mandatory, repeatable, and time spending job, it is okay to think about ways to automate this process [143, 142]. The market presents us with several tools for different programming languages, and we opted to research about JUnit, TestNG, Protractor, and Nightwatch. While JUnit and TestNG are the unit testing framework for the Java programming language, which helps in finding bugs and simplifies

the debug process, Protractor and Nightwatch focuses on testing the client side, looking for compatibility and user interface problems.

### 4.5.4 Preconditions identified

Besides knowing the necessary about the programming language that we choose for our application, we do not think that one must know much else since this should be a team's top priority.

## 4.6 Scheduling

Scheduling is performed using job schedulers, which have the ability to run services in background periodically at fixed times, dates, or intervals. With this process, we can automate system updates, system maintenance, or even run any script we desire. We may even start or stop a virtualized machine at any predefined moment [162].

### 4.6.1 Cron

Cron is a time-based job scheduler for Unix systems and is most suitable for scheduling repetitive tasks to run periodically at fixed times, dates, or intervals. Typically automates system maintenance and administration [163, 164].

#### Setup

Setting up Cron is pretty straightforward process in which we need to put a script at /etc/crond.d or /etc/crontab with the desired configuration and the command or script to be executed [164].

#### Conclusion

There are not many alternatives to run scheduled jobs as smoothly as Cron, which is a part of the Unix operating system. As it is not a very demanding job, the simplicity of Cron is very useful and powerful to look for alternatives [165, 163].

### 4.6.2 Chronos

Chronos is intended to be a replacement for Cron as a distributed and fault-tolerant scheduler to run on top of Apache Mesos. It can be used to interact with systems such as Hadoop, developed by Apache, and can even schedule jobs that run inside Docker containers [166].

#### Setup

To setup Chronos, primarily we need to install dependencies like Apache Mesos, Apache ZooKeeper, and the Java Development Kit. Then we just need to download the package and install it on major Linux distributions or OS X [167].

#### Conclusion

Chronos has some advantages over Cron, like having more flexibility in job scheduling, and supporting the definition of jobs triggered by the completion of other jobs [168].

The downside of using Chronos are its dependencies, which can cause unneeded overhead to our machines. If we do not require a succession of jobs and can put everything we need in a script, then the best is to stick with Cron [169].

### 4.6.3 Scheduling Conclusion

Scheduling repeatable tasks is another way to ease the work of an operations team which handles hundreds, thousands, or even just one server. A truly lightweight tool like Cron or Chronos can be configured to run periodically to run any script, which helps any maintenance and system administration.

### 4.6.4 Preconditions identified

We believe that the use of a Scheduling tool only requires that we know and plan the scheduled actions to execute on the most appropriate moments.

## 4.7 Monitoring

Releasing an application is just the first step towards customer satisfaction. To extend or improve that satisfaction an organization needs to know as soon as possible if any component is behaving unexpectedly. These components range from system, network, and application, which we need to observe and get notified if any of these stopped functioning [170].

### 4.7.1 Server Monitoring

Cloud-based applications and microservices architectures are becoming the new normal. These highly dynamic application environments require a new approach to monitoring such as to learn in real-time how an organization's servers are holding out [171]. Information about CPU, RAM, and Disk usage might come in handy to prevent future crashes or even to plan an environment scaling up and down, depending on an organization's necessity.

A service like this allows all members of a software team to look at the same data in the same web interface, diminishing the debate over where the problem might be. Some of these services are prepared to send an email or even an SMS to warn a one or more members of the team when the server is not working properly [172].

A problem which may surface is the overhead these tools might add, but that is a business decision [173, 174].

### 4.7.2 Application Monitoring

Another service that aims to diminish the time a software team is looking for what is wrong within the application is an application monitoring tool. A tool like this will record the

performance of critical transactions of the application, evaluate the performance of specific code segments down to SQL statements, and identify the most significant transactions. All that information will help to spot when things like response times, call counts or error rates perform poorly [175, 176].

Application monitoring will also help a team to ascertain if it is the application, the server CPU, or even the database that is bottlenecking its performance, and will efficiently decrease a team's debugging and resolution processes [175, 176].

Like the Server monitoring, this tool will also lead to some performance overhead, but we can choose to apply this type of monitoring to just some nodes/servers as to save overall performance [174].

### 4.7.3 Real User Monitoring

In addition to the knowledge that we can obtain from monitoring our servers and the application itself; we may need to know what our users are 'feeling' from the application. Information like the page load time by Geolocation will help us decide if we require investing in servers near our target countries. We also get information about our user's movement throughout the application, helping us to understand if the application is as we intended, from its simplicity to the load time of each asset, giving us an insight on what transactions are the most important. Furthermore, we even get JavaScript errors and what is triggering them [177].

With a tool like this, there is no need to wait for a customer to call and report a problem. The team will know right away what is happening, and if the application is underperforming and where.

### 4.7.4 New Relic

New Relic is one of the most known and complete Software as a Service in the monitoring category [178]. It is one of the few services that can help an organization to monitor all aspects of its environment as is the case of Server Monitoring, Application Monitoring, and Real User Monitoring.

With New Relic we can watch and analyze the real utilization we are getting at all times so that we can adjust our service to our necessities. With this process a company like Miniclip, who had oversubscribed at Amazon Web Services, could learn their actual utilization and scale their deployment appropriately, saving nearly $500,000 in one year [179].

The company's mantra is "Life's too short for bad software.", since a service like this is intended to help an organization to know what is wrong, and even what may be a mistake, reducing resolution times and preventing downtimes from happening.

#### Server Monitoring

As for server monitoring, New Relic provides an easy to setup agent to keep track of our servers. The setup process is as simple as to log into the New Relic platform and choose the Operating System of the target machine. If we deploy to Red Hat, Ubuntu, Debian, or CentOS we need to copy a set of links and paste them directly on the machine or add them in some setup script to run when bootstrapping the machine. If the target is Windows

Server, we get an executable file to install the agent, and then insert the given license key [180, 181].

Through the New Relic platform we have access to the CPU and RAM usage, the Disk capacity and I/O utilization, Processes running, and Network traffic. We can then establish a set of rules to alert someone if any of these values are reaching alarming numbers.

**Application Monitoring**

New Relic Application Monitoring (APM) is ready to be included in a project if we are using some of the most popular languages like Ruby, PHP, Java, Microsoft.NET, Python, or Node.js. Each one is a little bit different to include in the project, but it will have a great impact in the future. We can identify key transactions in our application and follow its performance. We can learn which transactions are most time-consuming. We will know how much time is being spent on Database calls and track the exact SQL statements that are slowing the application [182].

**Real User Monitoring**

New Relic Real User Monitoring, also known as Browser, gives us information about every loaded asset, Ajax request, user interaction, JavaScript events and errors, and works in the majority of frameworks like AngularJs, React, EmberJS, and BackBoneJS. It also informs us on specific browser performance and divides them by versions. Besides, with the premium version, New Relic tells us the performance of our application by Geography [183].

There are two ways to install this tool, one by inserting a piece of JavaScript code into our application, and the other to use the Application Monitoring (APM) for the available platforms.

**HTTP Health Check**

New Relic provides a way to check if our machines are online (Synthetics) providing health checks from North and South America, Europe, Asia and Australia with intervals ranging from 1 minute to 1 day [184].

For the premium version we get API testing and even Chrome-based Selenium scripts to check if everything is work appropriately.

**Mobile Monitoring**

New Relic Mobile Monitoring is the most mature tool for mobile applications. It tells us all the reasons why our apps crashed and give us the insight to solve them quickly and easily. It will even include an automatic trail of the user's interactions leading up to the crash.

In New Relic platform we have access to HTTP response times and error rates, the average memory consumed, view count, and average interaction time with our application [185].

**Free Account**

New Relic APM free account called Lite, gives us access to Response Times and Error Rates, External Services Metrics and Database Metrics and SQL Traces. The Mobile Lite gives us access to Mobile Summary and Performance Metrics. The Real User Monitoring Lite provides the Browser page views and page load times. While, Synthetics Lite gives us our health checks worldwide-coverage. All of this information will be available daily, which prevents us to look back on our history [186].

#### Conclusion

New Relic is probably the most famous and complete monitoring tool available on the market [178, 187]. It is ideal for an established product with significant demands for availability and performance, while the Lite account could help small startups to improve their product gradually and become something better in the foreseeable future.

The team behind New Relic chose wisely to separate each kind of monitoring, but made no efforts to do multiple price plans to each kind expect for APM, making it difficult to go from Lite to Pro.

### 4.7.5 Ruxit

Ruxit, as well as New Relic, is another Software as a Service full of different features to help with the monitoring of our environment.

The big difference between Ruxit and New Relic is that the former provides application monitoring with zero configuration. Ruxit OneAgent automatically learns about our environment's normal performance through artificial intelligence. If our application breaks, Ruxit tells us where and why and shows us the chain of events that led up to the problem while identifying the cause [188].

Companies like, YAHOO!, Cisco, and LinkedIn choose Dynatrace's Ruxit as their all-around monitoring service [189].

#### Server and Application Monitoring

Ruxit's Server Monitoring Agent can be deployed to any Windows machine and installed from a provided executable file, or to any Linux machine through the execution of two commands provided at Ruxit's platform [190].

At Ruxit's platform, we can analyze CPU and RAM usage, the Network traffic, and all the information about the disk. We also get information about running Processes and how much CPU, RAM was used, and how much traffic they generated per second. The artificial intelligence powered agent will tell us the average duration of the actions performed by the user and how much those actions happen per minute. Ruxit also provides information about what kind of device are being used to access our application and even the browser at each type of device.

#### Real User Monitoring

As others Real User Monitoring features, we need to add a JavaScript snippet to our code. Then, through Ruxit's platform, we see the average time of active sessions and the average of how many actions we get per session. Another great information we get is the percentage of returning users and new users, the percentage of real users, synthetics and robots going through our application, and the Geolocation of our users [191].

#### HTTP Health Check

Ruxit's HTTP Health Check feature just needs the URL of our site. However, we can customize the device, the screen size, the network type, and the name of the user agent.

It is possible to check an application from North America, Europe, South America (São Paulo), Asia (Singapore), and Oceania (Sydney), repeating each 5, 10, or 15 minutes.

#### Mobile Monitoring

Mobile Monitoring at Ruxit is still in its Beta phase but is already available for Android and iOS. For Android, we need to add Ruxit to our Gradle script while on iOS, we need to modify our Podfile and our Info.plist file [192].

In the platform, we can see information regarding the number of users, new users, sessions, and their Geolocation. We also get information about the number of crashes.

#### Free Account and Pricing

Ruxit does not have a Free Account type, providing only a free trial period of 30 days.

Ruxit's pricing is scalable depending on the application needs. For Server and Application Monitoring we pay for each node, for Real User Monitoring, we pay for each 500 user sessions, and for HTTP Health Check we pay for each 100 web checks. Ruxit also offers a startup plan including monitoring for ten nodes, 2 million user sessions, 30 thousand web checks paying around $30,000 for 12 months [193].

#### Conclusion

Ruxit provides a well-polished interface resembling Windows eight tile screen, being completely customizable. Its artificial intelligence powered agent is an asset for users who do not want to spend much time configuring two or three different tools.

The lack of a free account type may be a negative factor to those who want or need a cheap solution.

### 4.7.6 Pingdom

Pingdom is around since 2007 and has earned the trust of companies like Spotify, Facebook, Twitter, and thousands of others. Pingdom does not try to be a jack of all trades, and instead focuses on the web performance of products while reducing false alerts through multiple probes around North America, Europe, and Asia Pacific [194].

#### HTTP Health Check

Depending on the chosen plan, an organization will get from 10 to 250 checks per minute from any location that they see fit.

To setup Pingdom's HTTP Health Check we need to add our application address at Pingdom's platform.

#### Real User Monitoring

Pingdom's Real User Monitoring goes from 100 thousand page views at one site to 5 million page views at 50 sites. It also gives us information about the loading performance of our assets and a possibility to filter by the browser, platform, and country.

To setup Pingdom's R.U.M. we need to retrieve a JavaScript snippet provided at Pingdom's platform and add it to our application.

#### Free Account

Pingdom abandoned its free account plan early in 2016, providing only a free trial period of 14 days. Previously created accounts moved to a free account plan with lots of limitations. This plan is not available for new accounts [195, 196].

#### Conclusion

If an organization only needs to check the performance and availability of an application, Pingdom is a pretty solid choice. It offers a great variety of price plans and has the possibility to change it at any time depending on the requirements.

One of the strong points about Pingdom regarding their competition is their SMS alerting option which is significantly better than to receive just an email.

### 4.7.7 StatusCake

StatusCake, like Pingdom, is most focused to check the availability and performance of an application [197].

Companies like Microsoft, Netflix, TomTom, and many others use and trust StatusCake in one or more of their services [198].

#### HTTP Health Check

Like all other services like this, StatusCake only requires that we go into their platform and add the URL of our domain, the time to notify us that our application is offline (0 minutes is a possible choice, but more time might prevent false alerts), the amount of servers to confirm the application downtime, how much time between checks, and the Geolocation of these checks [199].

Along HTTP Health Checks, StatusCake offers TCP, DNS, SMTP, SSH, PING, and PUSH checks for different kinds of services.

#### Real User Monitoring

Since mid-2016, StatusCake removed their option to monitor real usage of applications.

#### Free Account

StatusCake is one of the few to offer a Free Account plan with unlimited monitors and alerts, a minimum of 5 minutes between checks, and the location of those checks are made at random.

Other plans offer 1-minute check rate with eight selectable test locations, and constant second check rate with more than 60 selectable test locations. All these come with virus checking, SSL Monitoring, Domain Expiration Monitoring, Page Speed Monitoring and Real Browser Testing [199].

#### Conclusion

StatusCake is the only service to provide Health Checks from all continents with several available locations at each [200].

In contrast with Pingdom, StatusCake offers a free account type which for small projects might just be the necessary. Besides, another great asset is the possibility to send SMS to a predefined contact group, alerting that the application is not responding.

### 4.7.8 Monitoring Conclusion

After deploying an application our first worry is, or should be, if it is running, its performance, how are the servers handling it, and even how our clients are navigating through it. With all these in mind, the market provides us with full-scale options like New Relic and Ruxit, and some much more specific like Pingdom and StatusCake. While New Relic and Ruxit handle server, application, and user monitoring, tools like Pingdom and StatusCake specialize in checking if the server is operational.

### 4.7.9 Preconditions identified

We believe that most of the Monitoring tools are easy to setup and do not require any previous knowledge. However, we need to ascertain what is imperative to monitor in our application's ecosystem and know how to use that information to help us improve.

| | New Relic | Ruxit | Pingdom | StatusCake |
|---|---|---|---|---|
| North America | ✓ | ✓ | ✓ | ✓ |
| South America | ✓ | ✓ | ✓ | ✓ |
| Europe | ✓ | ✓ | ✓ | ✓ |
| Africa | ✓ | ✓ | ✓ | ✓ |
| Asia | ✓ | ✓ | ✓ | ✓ |
| Oceania | ✓ | ✓ | ✓ | ✓ |

Table 4.7: Geographic location comparison between different monitoring tools

| | | | | |
|---|---|---|---|---|
| Aplication Monitoring | ✓ | ✓ | ✗ | ✗ |
| Mobile Monitoring | ✓ | ✓ | ✗ | ✗ |
| Server Monitoring | ✓ | ✓ | ✗ | ✗ |
| R.U.M. | ✓ | ✓ | ✓ | ✗ |
| HTTP Health Check | $99 (10,000 checks) | $0.20 (100 checks) | $13 (430,000 checks) | $20 (430,000 checks) |
| R.U. Sessions | $199 (500,000 pageviews) | $0.20 (500 sessions) | 100,000 pageviews included | ✗ |
| Free Acount | ✓(1 day retention) | ✗ | ✗ | ✓ |

Table 4.8: Feature comparison between different monitoring tools

## 4.8 Supervision

The purpose of a Supervision tool is to hand over some important responsibilities usually done by the operations team to the computer. These tools handle the start of tasks, services, processes, and other dependencies for the proper functioning of an application. Supervision tools in addition to starting these processes, they can even keep watch over them and even

restart them if they stop. In the probable situation in which a computer needs to be turned off, these tools will attempt to stop the application and those processes gracefully to prevent the corruption of data [201].

### 4.8.1 Upstart

Upstart is the default supervisor tool for Ubuntu and was developed by Canonical. It was designed to bring the machine to a normal running state after power-on, or gracefully shutting down services before shutdown. As a result, the design is strictly synchronous [202].

Its use is simple, requiring only the writing of scripts with minor configurations using normal shell scripting. Even though normal supervision is easy to setup, if we need to add any log rotation we need to configure logrotate together with copytruncate option [203, 201].

### 4.8.2 Systemd

Released in 2010 by Red Hat's software engineers Lennart Poettering and Kay Sievers, systemd became the default supervisor tool for Red Hat and Fedora UNIX systems [202, 204].

Unlike upstart, it can handle its own log rotation. However, it uses the same configuration with shell scripting to customize the system boot and shutdown, as well as to ensure that an application is running, and if not, restart it [201].

### 4.8.3 Supervisor

Supervisor is a client/server system that allows its users to monitor and control some processes on UNIX-like operating systems. It does not replace supervision tools like Upstart or Systemd, but works alongside them to control processes related to a project, and starts like any other program at boot time. One of the things that differs Supervisor from the default supervision tools is its ability to start several instances of a program [201].

Supervisor has its own type of configuration for their scripts, these being written in Python programming language, unlike the shell scripted Upstart and Systemd [201].

### 4.8.4 Circus

Circus was developed by Mozilla to monitor and control processes and sockets. Like other supervision tools, it can be programmed via command-line, but as seen in Figure 4.6 it has a web interface to help manage the machine processes and even a way to manage them programmatically through its Python API. Circus can handle its own log rotation, and as Supervisor, it can start several instances of a program. However, like Supervisor, it cannot act as UNIX init [201].

As it is, Circus is an optimal choice if a project uses WSGI, like the Django framework [201].

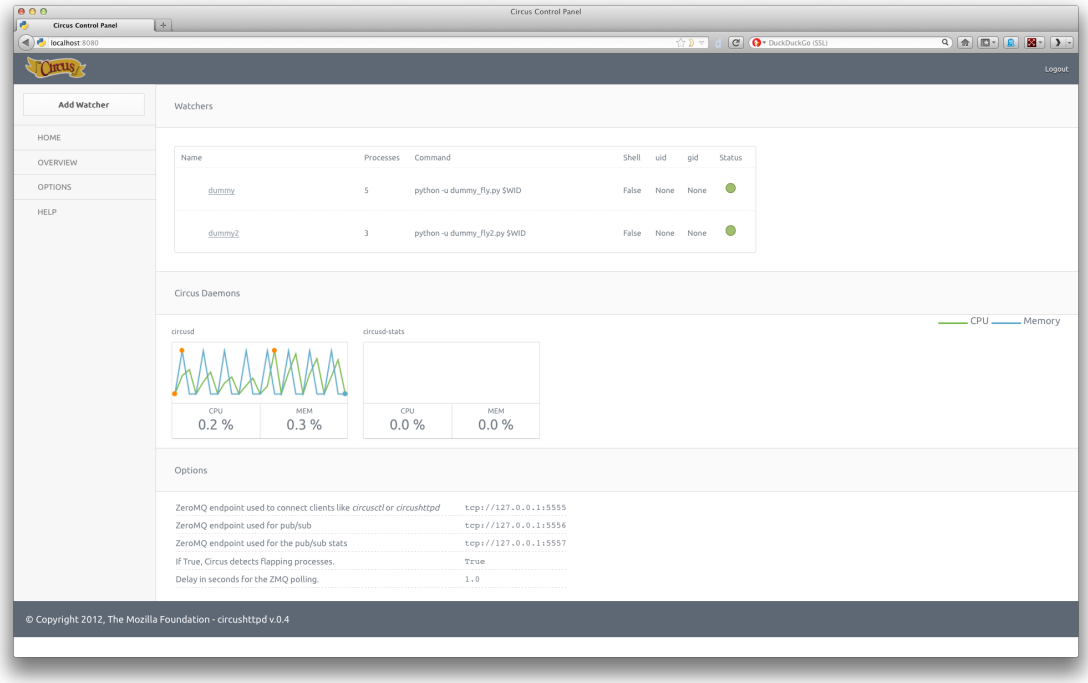



Figure 4.6: Example of Circus Dashboard [205]

### 4.8.5 Supervision Conclusion

Even if we are monitoring all aspects of our ecosystem, it will not resolve anything without human interaction, and for this reason, Supervision tools, in addition to handling the start and stop of applications and services, help us watch and restart them in the case of failure. Systems like Linux has tools like Upstart and Systemd already built-in. However, there are tools like Supervisor and Circus that are easy to configure and help keeping track of our application and its dependencies.

### 4.8.6 Preconditions identified

We concluded that a profound knowledge of the dependencies of our system and our application is crucial to help us setting up the Supervision tool for it to handle all those processes and tasks.

## 4.9 Logging

Logging or Log management, if done correctly, might bring significant benefits to an organization. Logging includes Log collection, in which a computer records everything happening with an application and system. Furthermore also includes Centralized Log aggregation, to facilitate teams reaching those logs, and Log rotation, to divide logs in multiple fields, generally daily or weekly files, facilitating the search for a specific log. Finally, it embraces

| | Upstart | systemd | Supervisord | Circus |
|---|---|---|---|---|
| Act as UNIX init | ✓ | ✓ | ✗ | ✗ |
| Log Rotation | ✓ (logrotate + copytruncate) | ✓ | ✓ | ✓ |
| Host | Ubuntu | Red Hat/Fedora | UNIX | UNIX |
| Starts several instances of a program | ✗ | ✗ | ✓ | ✓ |
| Script Language | Configuration + Shell | Configuration + Shell | Configuration | Python |
| GUI | ✗ | ✗ | ✗ | ✓ |

Table 4.9: Base comparison between different supervision tools

Log analysis and reporting, to check for compliance, system troubleshooting, security, and understanding user behavior [206, 207].

With all information of what's happening on a server, a team can parse it, analyze it, and take actions accordingly to improve several aspects of their product ecosystem.

### 4.9.1 Splunk

Splunk specializes in searching, monitoring, and analyzing machine-generated big data. It captures, indexes and correlates real-time data in a searchable repository from which it can generate graphs, reports, alerts, dashboards, and visualizations [208].

Splunk self-asserts itself as a leader in Operational Intelligence and claims to look closely to Machine Data to give insights that can help a company to become more productive, profitable, competitive, and secure [209].

Splunk motto transcribes as "You see servers and devices, apps and logs, traffic and clouds. We see data – everywhere."

#### Machine Data

Everything since logs, configurations, data from APIs, message queues, change events, the output of diagnostic commands, and much more are considered Machine Data. From Application Logs we get information about user activity, fraud detection, and application performance. Configuration Files tells us how an infrastructure has been set up, debugging failures, backdoor attacks, and time bombs [209].

#### Operational Intelligence

Operational Intelligence aims to turn machine data into valuable insights, giving a real-time understanding of what's happening across our environment, which will help us make informed decisions [210].

#### Setup

Splunk can be installed to Windows from an MSI executable file to Linux from a deb, rpm, or tgz file, to Solaris from a pkg.Z, tar.Z, or p5p file, and to Mac OS from a dmg, or tgz

file. After installation, Splunk will be available through port 8000, and we are able to upload log files, monitor files and ports, and even forward data from Splunk forwarder [211].

Splunk is prepared to analyze files from Tomcat's catalina, log4j, log4net_xml, log4php, ruby_on_rails, mysql_d, json, csv, apache, and many others.

**Logs Repository**

Splunk offer organizations the possibility to save their logs locally or in the cloud. If we choose to use local storage, we need to have the means to manage and secure them. However, if we opt for cloud storage it will be pricey and we have to be careful to what information we send, but then we do not have to worry about managing that information [212].

**Free Account and Pricing**

Splunk provides a free account type which gives us an indexing volume of 500MiB per day, universal real-time indexing of machine data, ad-hoc search and reports across real-time and historical data, highly customizable and interactive dashboards and much more.

Paid plans range from an indexing volume of 1GiB per day to more than 100GiB per day for the enterprise edition, while the cloud edition ranges from 5GiB per day to more than 20GiB per day. Both these plans will give us monitoring and alerts for individual and correlated real-time events, automatic discovery of patterns in our data, and much more that the free plan will not [213].

**Conclusion**

Splunk will provide the most complete and easy to use way to analyze an organization's logs. The free account type is a superb way to start, which then we can evolve to a startup account type, and if the organization wants even more advanced options, Splunk offers an enterprise account type.

### 4.9.2 Loggly

Through a cloud-based service, Loggly will mine tons of log data in real-time to reveal what matters to an organization, giving the insight to produce better code and deliver a better user experience [214, 215, 216, 217].

**Setup**

Loggly opts to use existing open standards like Syslog and HTTP instead of forcing the installation of an agent. It is capable of analyzing any text-based log like Ruby, Java, Python, JavaScript, PHP, Apache, Nginx, IIS, Tomcat, MySQL, and many others [218].

Loggly provides a way to go through our log data using regex statements.

**Logs Repository**

Loggly only offers a cloud-based repository. This option makes it hard for Loggly to persuade some companies to subscribe to their services, since leaving any information outside of their facilities might be a risk that those companies are not ready to cross [216].

**Free Account and Pricing**

Loggly has a free account type which will retain 200MiB of logs per day, during seven days. However, with Pro Plan, Loggly provides flexible overage protection, so that our logs can be available even if we occasionally exceed our subscription [219].

**Conclusion**

Loggly specializes in the field of finding operation problems for us to fix. It will not give us much insight on what we could do to improve something, but that may not be what we need. It is far lighter than other solutions and will give us what a Logger analyzer should give and nothing more.

### 4.9.3 Elastic Stack (ELK Stack)

The open source project ELK stack is composed by Elasticsearch for searching and analyzing data, Logstash which collects data, and Kibana to visualize and interact with the data [220].

Together, these three different open source products are commonly used in log analysis, business intelligence, security and compliance, and web analytics [221].

**Elasticsearch**

As stated before, Elasticsearch will search and analyze data in real time through a RESTful API which can be distributed, scalable, and highly available [222].

Elasticsearch is based on the Lucene search engine and uses JSON document-oriented to store complex real world entities.

**Logstash**

Logstash will collect, enrich, and transport our data from various schemas and formats. We can easily add plugins for custom data sources [223].

**Kibana**

The last piece of the stack, Kibana, was architected to work with Elasticsearch and lets us explore and visualize our data through a web platform [224].

**Logs Repository**

Being an open source project, we can choose where to store our logs, either locally or in the cloud.

**DIY or Outsourcing**

As the typical open source project, we have the possibility to assemble it ourselves, losing some time to learn it, mount it, and making it an asset.

Although, there are companies, like logit.io, which use the ELK stack and provide it to other companies as a service. Logit.io is customizable regarding their prices, letting us choose from 0.5GiB per day to 50GiB per day in 0.5GiB installments, and data retention from 1 to 30 days [225].

**Conclusion**

If we have the time or have a big team of developers we may try the ELK stack as our logger service, or if we are familiar with the ecosystem but do not have the time to mount it, solutions like logit.io, which host ELK as a service, might be a good bet [226, 227, 228, 229].

### 4.9.4 Logging Conclusion

Through monitoring our ecosystem, we can act quickly when something unexpected happens. However, thanks to logging we can improve our knowledge concerning the same ecosystem, boosting the application performance, and finding some problems otherwise undetectable, which might prevent backdoor attacks and time bombs.

The market provides us all different type of choices, beginning with the most enterprise one being Splunk, the ideal choice to save, analyze, and receive insight about an ecosystem logs. A fairly lighter option would be Loggly, a cloud-based solution for teams who only wants to keep track over their logs in a centralized location. Finally, we can opt for an open-source solution called Elastic Stack, composed of ElasticSearch, Logstash, and Kibana to store, analyze, and visualize our logs.

### 4.9.5 Preconditions identified

We found that a prior knowledge of the whereabouts of the system and application logs that we want to keep and analyze is essential to fasten the adoption and avail its benefits.

| | Splunk | Loggly | ELK Stack |
|---|---|---|---|
| Free Plan | ✓ (500MiB per day) | ✓ (200MiB per day + 7 days retention) | Open Source |
| Basic Plan | $170 (1GiB per day) | $55 (1GiB per day + 7 days retention) | ✗ |
| Cloud based | Splunk Cloud | ✓ | ✓(Outsource) |
| In-House Solution | Splunk Enterprise Splunk Light | ✗ | ✓ |

Table 4.10: Base comparison between different logging tools

## 4.10 Service Discovery

Service Discovery helps to keep track of all the services in a distributed system so that they can be found by other services as well as people operating those services. These systems are complex and need features like storing metadata about a service, health monitoring, varying query capabilities, real-time updates, and much more. At the bottom of it is a coordination mechanism for services to announce themselves and find others without configuration.

The main benefit is the "zero configuration" which replaces the hardcoded addresses by a service name. This technique becomes useful thanks to modern architectures in which nodes are created and destroyed frequently [230] [231].

### 4.10.1 ZooKeeper

ZooKeeper intends to assist distributed application with a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services [232].

ZooKeeper nodes store their data in a hierarchical name space, much like a file system or a tree data structure. Clients can read from and write to the nodes and in this way have a shared configuration service [233].

#### Setup

To setup a ZooKeeper server we need to download the JAR file. Then write a configuration file with tickTime, which is the basic time unit in milliseconds used by ZooKeeper, dataDir, which is the location to store the in-memory database snapshots, and finally the clientPort, which is the port to listen for client connections [231].

#### Client-side

From a client perspective, we need to connect to the server through its IP and port that we chose in the configuration file. We can do this with JAVA or C, which is lighter for our nodes [231].

#### Node Health Check

The server confirms that our nodes are working through ping commands [231].

#### Conclusion

ZooKeeper was a sub-project of Hadoop, later becoming a top-level project in its own right. It is probably the most capable service discovery tool, but it is not that straightforward to configure and manage it. The clients must be programmed to withstand most adversities that are difficult to write and often result in debugging challenges [231].

### 4.10.2 etcd

etcd is a distributed, consistent key-value store for shared configuration and service discovery, with a focus on being simple, secure, fast, and reliable, and written in Go [234].

#### Setup

We can setup etcd in several operating systems like Linux, Mac OS, and Windows. etcd also provides a Docker image with the latest release.

To start etcd just run the binary file provided and the port 2379 will be open for client communication and port 2380 for server-to-server communication. Finally, we need to set a key and share it between our nodes [231].

#### Combining with Registrator and confg

Registrator is a tool that automatically registers and deregisters services by inspecting containers as they are brought online or stopped.

confd act as a configuration management tool, which keeps configuration files up-to-date, and can even reload applications when configuration files change [231].

#### Conclusion

etcd was first provided by CoreOS but then expanded to multiple systems. It is easy to deploy, setup and use, the encryption and authentication by private keys are very safe, and the documentation is good. If combined with the right tools, it will solve most of our service discovery needs [234, 231].

### 4.10.3 Consul

Consul has multiple components, but as a whole, it is a tool for discovering and configuring services in our infrastructure. It provides features such as service discovery, health checking, key/value store and multi-datacenter.

Consul bases on running an agent which is responsible for health checking the services on the node as well as the node itself. These agents talk to one or several Consul servers that store and replicate the data. These servers elect a leader by themselves.

When some service, application, or node needs to discover something they can query the Consul servers or any of the Consul agents, which will forward queries to the servers automatically.

#### Gossip Protocol

Consul uses a gossip protocol, more precisely two different gossip pools, to manage and broadcast messages to the cluster. One pool for all members of a datacenter communicates, and other for servers to perform cross datacenter requests [231].

#### Node Health Check

Consul can check if a node is healthy through ping commands and even from HTTP requests to check for availability [231].

#### Embedded Service Discovery System

With Consul there is no need to build or use a third-party service discovery system since it is already a feature, unlike ZooKeeper and etcd [231].

#### Conclusion

Free, easy to setup, and powerful. Consul is in many cases better than etcd and ZooKeeper, as it was designed with services architecture and discovery in mind.

What distinguishes Consul from the rest it is the support for multiple data centers a health checking without added third-party tools which may cause unnecessary overhead to our ecosystem. Moreover, of course, the way Consul propagates knowledge about the cluster using gossip makes it easier to set up than the competition [231, 235].

### 4.10.4 Service Discovery Conclusion

In a big and complex ecosystem, it might be too tricky and chaotic to handle it manually, and for that, we use Service Discovery tools to automatically store metadata about our servers and helping those to find their service pool. We have some options in the market, ranging from mature, complete and complex like ZooKeeper, easier to manage and capable

of working in multiple Operating Systems like etcd, and even a new, powerful, and using modern communication protocols like Consul.

### 4.10.5 Preconditions identified

Regarding a System Discovery tool, we believe that a strong knowledge of the ecosystem and network is imperative to implement it correctly. This kind of tool requires that we have the knowledge to do the actions manually and only then test the tool extensively before using it in production.

## 4.11 Thesis Questions Discussion

The contents of this chapter allowed us to answer the following three out of four questions for our thesis.

### 4.11.1 What body of knowledge should we collect and formalize in order to help software teams to practice DevOps?

As seen throughout this chapter, we presented all aspects which constitute a DevOps work cycle, and several tools concerning each one. The decision regarding the choice of the researched tools lies in the fact that majorly they are the market leaders, as of the others we believe that they depict a different approach on how those leader's tools work.

Moreover, we believe that teams have the necessity to automate some aspects of their jobs in which Automation, Scheduling, and Service Discovery can prove themselves useful. We also found that through Virtualization and Provisioning we can prevent the usual mistake of "it worked on my machine," the lack of knowledge concerning the production environment by the developers, and allows us to create or expand an environment easily. Finally, they need to guarantee that their application is always working, and for that, they can use Testing to check the application correctness, Monitoring to be aware in real-time of the application ecosystem, and Supervision to handle possible failure of the application. In the possible case of failure, teams can use a Monitoring or Logging solution to know what went wrong with the application or the ecosystem, which helps the team to restore those quickly.

### 4.11.2 Which tools categories should we consider when elaborating the Knowledge Map?

When developing the Knowledge Map and researching about the categories and respective tools, we found some aspects which are influencing the market to adopt one tool or another. Like everything in business, we concluded that the price is a major concern when choosing the Infrastructure, the Automation, the Monitoring, and the Logging tools. Another concern is the learning curve for some In-house Infrastructure solutions, for the Automation and the Testing tools. The other concern is some specificity about the category, such as the type of virtualization we want when deciding about the Virtualization and Provisioning tool, the aspect we want to test when choosing a Testing tool, the aspect we want to monitor

when selecting a Monitoring tool. Finally, we found that teams only change the built-in Supervision tool for some cases, and only adopt Service Discovery when the application architecture demands it. For Scheduling, we did not found any concern in particular to change the built-in solutions.

### 4.11.3 Which technologies should be adopted?

Each case is different. Therefore there is not a single solution. To answer our question, we identified the forces that influence the choice of one tool over another as to help teams to use those to identify which is their best option for their case.

# Chapter 5

# Validation

Any scientific work must be validated to guarantee that it was executed impartially and is replicable. In this chapter, we describe the planned strategy for our validation.

## 5.1 Methodology

The methodology we chose to validate our thesis is a quasi-experiment. This type of validation uses naturally pre-existing groups, which in our case can be a team composed of developers and operators in an organization.

The process used to validate our Knowledge Map had four main steps. The first step was to find multiple organizations that were not using any DevOps process and were willing to change their way of work. Then we would interview them, while identifying some metrics, about the current state of how they were delivering their applications, and how much time they were taking in some specific steps of that process. After that, we would provide these organizations with our Knowledge Map and give them none to minimal support to choose and adopt some of the tools present in the Map. Finally, we would interview the organization again to gather the same indicators. If those indicators improved, then we would be able to validate the gain of using our Knowledge Map.

With this work, we consider that the time and customer satisfaction our main focus, and will try to improve both of them.

## 5.2 INESC TEC

We choose to do our validation at INESC TEC. INESC TEC is a research and development institute founded in 1985 and located on the campus of the Faculty of Engineering of the University of Porto, which brings together more than 650 researchers, of which more than 270 have PhDs. INESC TEC was created to act as an interface between the academic world, the world of industry and services and the public administration in Information Technologies, Telecommunications and Electronics (ITT&E).

At INESC TEC, projects have different durations, ranging from a couple of months to a few years, different team sizes, ranging from one or two people to a big and international team. These teams are self-organizing and do not follow any particular set of rules when developing a project.

Most often, the infrastructure used is allocated on their buildings, and managed by the system administrators or the developers. Lately, they started to embrace some cloud services like Azure and Microsoft 365.

### 5.2.1 The team

The team we were able to enroll in the experiment, provide the Knowledge Map and support was composed of two people. One was a recently master graduated student from Medical Informatics, while the other has about ten years experience in software development, including some experience working with Microsoft's Kinect at INESC TEC. Both had some experience in web front-end development, especially in JavaScript, but no prior knowledge of agile development, the Java programming language, and unit testing.

## 5.3 Objectives and Metrics

Since INESC TEC is different from a traditional organization with established objectives and demands, they want to provide their researchers with the latest trends in the market, to attract students and recent graduates in evolving a few steps further before hitting the job market. The goal was to set some guidelines for one team, and if the results were satisfactory, then pass those to the upcoming projects at INESC TEC to improve and standardize the infrastructure management processes.

The main objectives concerning the development process that we intended to reach were:

- Ensure that the development environment is as identical as possible to the production environment;
- Reduce the time that someone is spending testing the product through automation of this process.

Regarding the production process we identified the following objectives:

- Reduce the time for a project to go from development to deployed through the provisioning and virtualization of its infrastructure;
- Reduce the time spent and error making when deploying the product through automation and repeatability;
- Reduce or completely remove downtime when the product is updated through load balancing;
- Drastically increase the deploys done through a controlled and continuous integration process.

Finally, for the maintenance process we identified the following objectives:

- Reduce the time to notice and find an error through the integration of application, server, and real user monitoring tools;
- Ensure that in the event of an application failure, a supervision tool will try and restart it, resuming normal service.

With these objectives in mind, the following metrics were collected:

- Time that someone is testing the product after an iteration;
- Time between having the updated source code to a build of the project;
- Time to prepare the infrastructure and update the application;
- Downtime required to upgrade versions;
- Number of deploys within a month;
- How does the team notice errors;

Even though the main interests behind adopting DevOps might be purely financial and technical, there are some other benefits for the team such as:

- The sense of evolution through the learning of new processes and tools;
- Increase the trust level through a trustworthy and reliable lifecycle;
- Reduce the time and concern of manually monitoring the servers thanks to email and SMS notifications when something is misbehaving.

These objectives bring different kinds of metrics, such as:

- Motivation to adopt DevOps processes and tools;
- Difficulty to learn those processes and tools;
- Quantitative estimate of the stress when updating the project;
- Quantitative estimate of the concern for the well-being of the servers.

## 5.4 Initial State

Before the implementation, the team was interviewed, which helped us gather the following initial values for each metric, noting that we conducted some quantitative questions marked with an asterisk and assumed that 1 was the worst value and 5 was the best value:

| Metric | Observed Value |
|---|---|
| Parity between the development environment and the production environment * | 4 |
| Time for the team to test the product | 90 minutes |
| Time waiting to build the project | 10 minutes |
| Time that someone provisions the infrastructure | 5 minutes |
| Downtime required to upgrade versions | 5 minutes |
| Quantity of deploys within a month | Uncertain |
| Qualitatively, how do they notice errors | User feedback and manual testing |
| How to handle an application or service failure | Tomcat handles application failure |
| Time to notice an application error | Depends on User Feedback |
| Trust level when updating the project * | 4 |
| Concern for the well-being of the servers * | 3 |
| Motivation to learn DevOps * | 1 |
| Difficulty presumed to learn DevOps * | 2 |

Table 5.1: Initial metrics identified

## 5.5 Quasi-Experiment

This quasi-experiment was done in a period of four weeks in one of the projects being developed at INESC TEC. Every couple of weeks we gathered, Skyped, or exchanged emails to define the goals for the next week and review what happened during the prior week, in an attempt to evaluate their evolution.

The project was composed of a backend API alongside a frontend website. The backend API was developed using Spring Framework which was connected to a PostgreSQL database, whereas the frontend was developed using AngularJS Framework.

The application, was a simple contact management platform with a login page and pages to create, list, edit, and remove contacts.

**Backend Service (Spring)**

The team started by developing the backend service with Play Framework using Scala as the programming language. However, they did not adapt well to the language and the available libraries. Therefore they opted to change and use **Spring Framework** as their backend service.

The Spring framework is an open source application framework for the Java Platform. This framework requires Maven on the development computer, and Java Development Kit, Tomcat, and Apache to be installed on both the development and production computers.

**Frontend Service (AngularJS)**

We opted to use the first version of **AngularJS** because of its huge amount of support and libraries available all-around. AngularJS is a JavaScript open source web application framework developed by Google.

We opted to use AngularJS generator provided by Yeoman to handle the injection of dependencies like Jquery and organize the code for production with uglify, minify, and many others.

**Infrastructure**

Regarding the infrastructure, INESC TEC provided the team with **five virtual machines hosted in-house**, where they could deploy their ecosystem.

We decided to use two machines for database replication, two machines for the application, and finally, the last machine would be responsible for the load balancing of the user traffic.

**Provisioning and Virtualization**

To reproduce the production system with dependencies and configurations on the development environment, we opted to use **Vagrant**. It would install and start the dependencies to run the Spring server, as well as the server itself, and provide the pages with the AngularJS code to the users.

**Unit Testing**

The team was not familiar with unit testing, as such, we opted for an easy approach of **JUnit**. The main objective was for them to learn the basics and realize the benefits of TDD development.

JUnit is the unit testing framework for the Java programming language used by the Spring framework. We chose it instead of TestNG because their coordinator had major background and documentation about it.

#### E2E Testing

The team was not familiar with end-to-end testing as well, however, since they opted for AngularJS, the only available option was **Protractor**.

Protractor depends on several libraries like nodeJS, jasmine, and selenium, however, the most recent versions are capable of installing all those dependencies.

The tests consisted of trying to:

- An incorrect login;
- A successful login;
- Creating a contact;
- Checking the existence of that contact;
- Editing that contact;
- Checking if the contact was edited;
- Removing that contact;
- Checking if the contact was removed;
- Logging out.

#### Deployment

Since the project was small, we opted to use **Ansible**, a simpler Deployment solution, to connect to the different servers through SSH, and update them to the desired version of our project or even to update one or several dependencies. The team developed different scripts for the servers with the service and the machines with the databases.

## 5.6 Discussion

We were not able to validate the full DevOps adoption by the team. Even though we approach them earlier, they had some unfinished projects at INESC TEC which prevented them to commit to our project's needs completely. The fact that they had almost no knowledge of agile development, unit testing, and Java programming language did not help because we had to spend a lot of time training them in the basics.

After conducting and analyzing the initial interview, we took some conclusions. Their motivation to learn DevOps was awfully low which might suggest a similar commitment to the project. Another aspect that worried us was the team trust level when updating the project, which we believe was very high for someone with little background in operations and testing knowledge. Regarding the question about the parity between the development and the production environment, we believe that they were not aware of its implication since their answer was a four, although they were using Windows 10 to develop and Linux for production. Finally, their concern for the well-being of the servers was somewhat low, since

they did not have a monitoring solution in mind and their only way of finding out that an error was occurring was user feedback and the ability of Tomcat to handle the application.

Even so, since we did not have many options we proceeded. However, we encountered the first problem when we advised the team to use Play Framework with the programming language Scala because of its easiness to build and deploy an application package. However, the team was not ready for such complexity and took the initiative to adopt the Spring Framework with the programming language Java. The team was able to learn and adopt AngularJS easily though they were creating and managing everything manually, even though we had advised them about tools to automate several processes.

Then, the team learned Vagrant and created a virtual machine able to run their application. After that, the team had to learn the basics of testing, and as such, they focused on JUnit to test the backend of their application, and protractor to test the frontend counterpart.

Finally, the team was able to configure virtual machines hosted on their development environment through Ansible. However, they did not have the time to test their configurations aginst the infrastructure provided by INESC TEC.

We intended to include a New Relic agent to monitor the application as well as the servers. A supervisor to guarantee that Tomcat was always online as well as the application and databases. And we wanted to integrate the application with Loggly to handle the application's and system's logs.

### 5.6.1 Lessons Learned

The validation did not went as planned and did not provided the results needed. However, it provided valuable information so that we were able to identify some base competencies and preconditions that we believe mandatory for a successful DevOps adoption by another team. This preconditions were added to chapter 4 sections and are a major contribution of this thesis.

In this specific case, we believe that the team members should know about the **programming language** that they were going to use, and since they were using a **framework**, they should have tried it and deploy it manually with ease before attempting DevOps. They should have a large experience in automated **testing** because all DevOps processes depend on the correctness of the tests on the application and infrastructure. Lastly, we think that a strong knowledge of **agile development** is imperative since DevOps extends that knowledge for managing the infrastructure.

# Chapter 6

# Conclusions

This chapter presents our findings and future work. A quick overview of this document is presented. Considerations are made, and future work is discussed.

## 6.1 Contributions

This dissertation main contribution focuses on the Knowledge Map and the extensive research conducted concerning DevOps categories and some of its tools. Through this study, we want to facilitate and accelerate the adoption of those tools by software teams with a document with updated and structured information regarding a set of DevOps tools. This work can also be seen as a comparing point between new and updated tools for further investigation.

## 6.2 Considerations

Throughout this dissertation, we were able to answer three out of four questions of this thesis. We were not able to fully validate our findings. However, we were able to define a minimum set of requirements to be met by a software team for a successful DevOps adoption.

We respond briefly to our thesis questions as follows:

**What body of knowledge should we collect and formalize in order to help software teams to practice DevOps?**

As stated and expanded in chapter 4, we presented all aspects which constitute a DevOps work cycle, and several tools concerning each one. The decision regarding the choice of the researched tools lies in the fact that majorly they are the market leaders, as of the others we believe that they depict a different approach on how those leader's tools work.

**Which tools categories should we consider when elaborating the Knowledge Map?**

We found that the most common aspects that organizations have in mind when choosing a tool over another is its price, its learning curve, and its features.

**Which technologies should be adopted?**

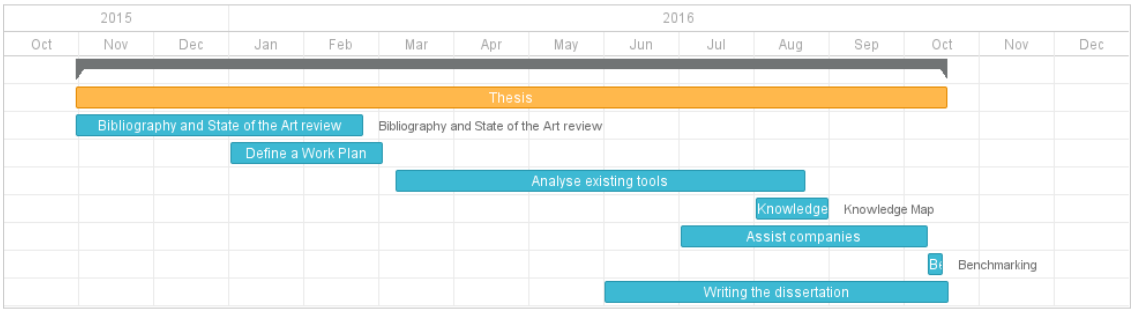



Figure 6.1: Gantt diagram representing the work done over the time

We concluded that every case is special and has different requirements, and organizations have distinct possibilities. As such, we believe that there is no right answer for any scenario, but there is a good fit for different situations.

**How much will teams improve while using our Knowledge Map?**

Unfortunately, as stated in chapter 5, we were not able to learn and quantify a team's improvements throughout the conception of this dissertation.

## 6.3 Work Schedule

After deciding that our research would focus on DevOps categories and their tools, in November 2015 we started by researching what DevOps was. First, we gathered some significant bibliography; we study it, and then drafted a state of the art document until mid-February 2016. Within that time a little after that we defined a work plan to follow in the following months. As such, we started to gather the market leader DevOps tools and some other interesting ones. We analyzed them, researched about them, set them up, and tried some basic functioning. Due to the quantity and complexity of categories and tools, this process lasted from March to August. However, we started to write the final document of this dissertation around June, which lasted until October 2016.

While we were doing both the research on the tools and writing the dissertation, we also developed the Knowledge Map with DevOps concepts and tools and approached INESC TEC regarding the collaboration. There were some talks about it in mid-May, and we established a partnership around June. However, the team was finished some unfinished projects and had a long vacation during August, which only allowed them to start on our project around 15 September. As explained in chapter 5, this prevented them to fully adopt DevOps and us from gathering the needed metrics for our validation. However, it allowed us to realize that there are a couple of conditions and basic knowledge to be met before diving in the DevOps ocean.

The schedule can be observed in the Gantt diagram in Figure 6.1.

## 6.4 Future Work

### 6.4.1 Research update

As stated at the beginning of chapter 4, DevOps and tools are continuously evolving and providing more and better features. Therefore, we believe that in the near future, some of the information presented in this dissertation concerning those tools will be outdated, and as such, a possible work for the future would be to revisit all of those.

### 6.4.2 Validation

In the future, we should validate this thesis using the methodology described in chapter 5, as to guarantee that this work was executed impartially and replicable.

### 6.4.3 Take the study to a wider population

The extensive research about DevOps and it practices indicates us the great benefits of adopting some of its processes and tools. In this dissertation, we were not able to completely validate a single case, which led us to research about the requirements on which a team should master before attempting to adopt a process or tool. As such, in the future, we would like to take this study and validation to a wider, able, and willing population to actually validate the benefits brought by the adoption of DevOps.

## 6.5 Conclusion

Several market leaders' organizations have noticed the enormous benefits of adopting DevOps processes and tools. This becomes apparent on small organizations as well as it allows them to work and deliver faster and better.

This document is the culmination of tremendous efforts to thoroughly research on several different tools to pave the way for organizations interested in adopting those and want more centralized knowledge.

The absence of validation for this dissertation is our greatest sorrow, which, unfortunately, was difficult right from the start when we did not find many companies who were not using DevOps and had an interest in adopting it, mostly thanks to the fear of changing their current processes. The team that we found to try and adopt DevOps were not able to do it during the conception of this document, which enlightened us with some conditions that a team must met to fasten their adoption process.

Nonetheless, we are happy with our research and the final state of this dissertation and all the gathered information about a new approach to developing software which is quickly taking the market and benefiting those who use it.

# Glossary

**Agile** Agile software development refers to a collection of development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.

**Analytics** Google Analytics is a freemium web analytics service offered by Google that tracks and reports website traffic.

**Apache** Apache is a web server software.

**API** Acronym for Application Programming Interface.

**APT** The Advanced Package Tool is a free software which handles the installation and removal of software on the Debian-based distributions.

**AWS** Acronym for Amazon Web Services.

**Built-in** Computer programs that are built into an operating system and are available automatically when it is installed on a machine.

**Cassandra** Apache Cassandra is a free and open-source distributed database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure.

**centOS** CentOS is a Linux distribution that attempts to provide a free, enterprise-class, community-supported computing platform which aims to be functionally compatible with its upstream source, Red Hat Enterprise Linux.

**CoreOS** CoreOS is an open-source lightweight operating system based on the Linux kernel and designed for providing infrastructure to clustered deployments, while focusing on automation, ease of application deployment, security, reliability and scalability.

**CouchDB** Apache CouchDB is open source database software that focuses on ease of use and having an architecture that "completely embraces the Web".

**CPU** Acronym for Central Processing Unit.

**Debian** Debian is a Linux computer operating system distribution.

**Dev** Development.

**DOM** Acronym for Document Object Model.

**DSL** Acronym for Domain Specific Language.

**FaaS** Acronym for Function as a Service.

**Fedora** Fedora is an operating system based on the Linux kernel, developed by the community-supported Fedora Project and sponsored by Red Hat.

**Framework** In Object-Oriented Programming, frameworks are reusable designs of all or part of a software system described by a set of abstract artifacts and the way they collaborate.

**FreeBSD** FreeBSD is a free Unix-like operating system descended from Research Unix via the Berkeley Software Distribution.

**Git** Git is a version control system that is used for software development and other version control tasks.

**GUI** Acronym for Graphical User Interface.

**IaaS** Acronym for Infrastructure as a Service.

**IDE** Acronym for Integrated Development Environment.

**IIS** Acronym for Internet Information Services.

**JDK** Acronym for Java Development Kit.

**LAMP** Acronym for Linux, Apache, Mysql, and PHP.

**Linode** Linode, LLC is an American privately owned virtual private server provider company based in New Jersey, United States.

**Linux Foundation** The Linux Foundation is a non-profit technology trade association chartered to promote, protect and advance Linux and collaborative development.

**Mac OS** Mac OS is the current series of Unix-based graphical operating systems developed and marketed by Apple Inc. designed to run on Macintosh computers.

**Microsoft SQL Server** Microsoft SQL Server is a relational database management system developed by Microsoft.

**MongoDB** MongoDB is a free and open-source cross-platform document-oriented database program.

**MySQL** MySQL is an open-source relational database management system.

**Node.js** Node.js is an open-source, cross-platform JavaScript runtime environment for developing a diverse variety of tools and applications.

**NoSQL** A NoSQL database provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases.

**openSUSE** openSUSE, formerly SUSE Linux and SuSE Linux Professional, is a Linux-based project and distribution sponsored by SUSE Linux GmbH and other companies.

**Ops** Operations.

**Oracle Database** Oracle Database is an object-relational database management system produced and marketed by Oracle Corporation.

**PaaS** Acronym for Platform as a Service.

**Pattern** In software, it is a recurrent, recognized good solution for a recurrent architectural, design or implementation problem.

**Performance** General measure that may mean short response time, high throughput, low utilization of computing resources, etc.

**PHP** PHP is a server-side scripting language designed primarily for web development but is also used as a general-purpose programming language.

**pip** pip is a package management system used to install and manage software packages written in Python.

**PostgreSQL** PostgreSQL is an object-relational database with an emphasis on extensibility and standards-compliance.

**PowerShell** PowerShell is a task automation and configuration management framework from Microsoft, consisting of a command-line shell and associated scripting language built on the .NET Framework.

**QA** Quality Assurance.

**RAM** Acronym for Random Access Memory.

**Requirement** It is a statement that identifies a necessary attribute, capability, characteristic, or quality of a system in order for it to have value and utility to a user.

**RHEL** Red Hat Enterprise Linux is a Linux distribution developed by Red Hat and targeted toward the commercial market.

**SaaS** Acronym for Software as a Service.

**Solaris** Solaris is a Unix operating system originally developed by Sun Microsystems which was later acquired by Oracle.

**SSD** Acronym for Solid State Drive.

**SSH** Acronym for Secure SHell.

**Use Case** In software engineering, it is a description of an intended system's behavior as the respond to an outside request or interaction.

**vCPU** Acronym for virutal Central Processing Unit.

**VM** Acronym for Virtual Machine.

**Windows** Windows is a metafamily of graphical operating systems developed, marketed, and sold by Microsoft.

**WordPress** WordPress is a free and open-source content management system based on PHP and MySQL.

**YAML** YAML is a human-readable data serialization language that takes concepts from other programming languages.

**Yum** yum is a software package manager that installs, updates, and removes packages on RPM-based systems.

# Bibliography


[1] J. Roche, "Adopting DevOps Practices in Quality Assurance Merging the art and science of software development," *Magazine - Communications of the ACM*, pp. 1–8, 2013.

[2] R. Miller, "Google Data Center FAQ," 2012. [Online]. Available: http://www.datacenterknowledge.com/google-data-center-faq-part-3/

[3] R. Miller, "Facebook DataCenter, Servers and Infrastructure FAQ," 2016. [Online]. Available: http://www.datacenterknowledge.com/the-facebook-data-center-faq/

[4] M. Huttermann, *DevOps for Developers*, 2012, vol. 1.

[5] S. Sharma and B. Coyne, *DevOps For Dummies*, 2015, vol. 1, no. 2. [Online]. Available: http://public.dhe.ibm.com/common/ssi/ecm/ra/en/ram14026usen/RAM14026USEN.PDF?

[6] M. Loukides, "What Is DevOps?" p. 15, 2012. [Online]. Available: http://shop.oreilly.com/product/0636920026822.do

[7] flowchainsensei, "The Many Roles in Software Projects," 2011. [Online]. Available: https://flowchainsensei.wordpress.com/2011/02/25/the-many-roles-in-software-projects/

[8] R. Buyya, R. Buyya, C. S. Yeo, C. S. Yeo, S. Venugopal, S. Venugopal, J. Broberg, J. Broberg, I. Brandic, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. June 2009, p. 17, 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1528937.1529211

[9] G. Reese, *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud*, 2009.

[10] M. Rouse, "What is Cloud Computing?" 2015. [Online]. Available: http://searchcloudcomputing.techtarget.com/definition/cloud-computing

[11] Business Cloud, "Cloud Computing Solution," 2015. [Online]. Available: http://www.business-cloudd.info/cloud-computing-solution/

[12] M. Armbrust, A. Fox, R. Griffith, A. Joseph, and RH, "Above the clouds: A Berkeley view of cloud computing," *University of California, Berkeley, Tech. Rep. UCB*, pp. 07–013, 2009. [Online]. Available: http://scholar.google.com/scholar?q=intitle:Above+the+clouds:+A+Berkeley+view+of+cloud+computing{#}0

[13] N. Computer and S. Division, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology," 2011.

[14] Techopedia, "What is a Cloud Provider?" 2015. [Online]. Available: https://www.techopedia.com/definition/133/cloud-provider

[15] F. Paul, "The Incredible True Story of How DevOps Got Its Name," 2014. [Online]. Available: https://blog.newrelic.com/2014/05/16/devops-name/

[16] S. Elliot, "DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified," no. December, 2015. [Online]. Available: http://info.appdynamics.com/rs/appdynamics/images/DevOps-metrics-Fortune1K.pdf

[17] S. W. Ambler, "Top 10 Practices for Effective DevOps," 2013. [Online]. Available: http://www.drdobbs.com/architecture-and-design/top-10-practices-for-effective-devops/240149363

[18] Techopedia, "What is Monitoring Software?" 2015. [Online]. Available: https://www.techopedia.com/definition/4313/monitoring-software

[19] Techopedia, "What is Data Logging," 2015. [Online]. Available: https://www.techopedia.com/definition/596/data-logging

[20] Techopedia, "What is Job Scheduling?" 2015. [Online]. Available: https://www.techopedia.com/definition/7882/job-scheduling

[21] L. Sun, H. Dong, F. K. Hussein, O. K. Hussein, and E. Chang, "Cloud service selection: State-of-the-art and future research directions," *Journal of Network and Computer Applications*, p. 17, 2013.

[22] A. Osterwalder, "MODELLING VALUE PROPOSITIONS IN E-BUSINESS," 2003.

[23] S. Nicola, E. P. Ferreira, and J. J. P. Ferreira, "A Quantitative Model for Decomposing & Assessing the Value for the Customer," *Journal of Innovation Management*, vol. 1, pp. 104–138, 2014.

[24] S. Zaharoff, "Business application hosting on-premises vs. cloud options," 2014. [Online]. Available: http://searchdatacenter.techtarget.com/feature/Business-application-hosting-on-premises-vs-cloud-options

[25] S. Shirinbab, L. Lundberg, and D. Ilie, "Performance Comparison of KVM, VMware and XenServer using a Large Telecommunication Application," 2014.

[26] I. Pratt, "Xen," 2003. [Online]. Available: https://sourceforge.net/p/xen/mailman/message/5533663/

[27] Xen, "Xen Project Release Features - Xen," 2016. [Online]. Available: https://wiki.xenproject.org/wiki/Xen{_}Project{_}Release{_}Features

[28] Citrix, "XenServer - Support and Self-Help Resources - Citrix," 2016. [Online]. Available: https://www.citrix.com/products/xenserver/support.html

[29] Dougvj, "What is the difference between Full, Para and Hardware assisted virtualiazation? - Stack Overflow," 2014. [Online]. Available: http://stackoverflow.com/questions/21462581/what-is-the-difference-between-full-para-and-hardware-assisted-virtualiazation

[30] VMware, "VMware Understanding Full Virtualization, Paravirtualization, and Hardware Assist."

[31] M. Rouse, "What is paravirtualization? - WhatIs.com," 2006. [Online]. Available: http://searchservervirtualization.techtarget.com/definition/paravirtualization

[32] VMware, "Virtualization Overview."

[33] M. Rouse, "What is live migration?" 2006. [Online]. Available: http://searchservervirtualization.techtarget.com/definition/live-migration

[34] VMware, *VMware company history*, 2016. [Online]. Available: http://www.vmware.com/company/mediaresource/milestones.html

[35] VMware, "Server Virtualization with VMware vSphere | VMware India," 2015. [Online]. Available: http://www.vmware.com/in/products/vsphere

[36] S. Lohr, "VMware market share more than 80%," *The New York Times.*, 2009. [Online]. Available: http://www.nytimes.com/2009/08/31/technology/business-computing/31virtual.html?pagewanted=2{&}{_}r=2{&}partner=rss{&}emc=rss

[37] KVM, "Linux 2 6 20 - Linux Kernel Newbies," 2007. [Online]. Available: https://kernelnewbies.org/Linux{_}2{_}6{_}20{#}head-bca4fe7ffe454321118a470387c2be543ee51754

[38] I. Habib, "Virtualization with KVM," *Linux Journal*, 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1344217

[39] J. Brockmeier, "KVM or Xen? Choosing a Virtualization Platform | Linux.com | The source for Linux information," 2010. [Online]. Available: https://www.linux.com/news/kvm-or-xen-choosing-virtualization-platform

[40] RedHat, "RHEL 5.4 Release Notes," 2009. [Online]. Available: https://access.redhat.com/documentation/en-US/Red{_}Hat{_}Enterprise{_}Linux/5/html/5.4{_}Release{_}Notes/index.html?

[41] D. Linthicum, "Try again, cloud contenders: Amazon, Google, and Microsoft have won | InfoWorld," 2014. [Online]. Available: http://www.infoworld.com/article/2608773/iaas/try-again--cloud-contenders--amazon--google--and-microsoft-have-won.html

[42] J. Panettieri, "Cloud Market Share 2016: AWS, Microsoft, IBM, Google - ChannelE2E," 2016. [Online]. Available: https://www.channele2e.com/2016/02/04/cloud-market-share-2016-aws-microsoft-ibm-google/

[43] J. Tsidulko, "Keeping Up With The Cloud: Top 5 Market-Share Leaders - Page: 1 | CRN," 2016. [Online]. Available: http://www.crn.com/slide-shows/cloud/300079669/keeping-up-with-the-cloud-top-5-market-share-leaders.htm

[44] Cloud Spectator, "VENDOR BENCHMARK 2016 NORTH AMERICA REPORT TOP 10 CLOUD Price-Performance Analysis of the Top 10 Public IaaS Vendors," 2016.

[45] C. Babcock, "AWS S3, Data Transfer Among Its Most Popular Services: Report - InformationWeek," 2016. [Online]. Available: http://www.informationweek.com/cloud/infrastructure-as-a-service/aws-s3-data-transfer-among-its-most-popular-services-report-/d/d-id/1326180

[46] Amazon, "Amazon Web Services (AWS)," 2016. [Online]. Available: https://aws.amazon.com/pt/

[47] Amazon, "Infraestrutura global," 2016. [Online]. Available: https://aws.amazon.com/pt/about-aws/global-infrastructure/

[48] Amazon, "Servidor de nuvem e hospedagem do Elastic Compute Cloud (EC2) – AWS," 2016. [Online]. Available: https://aws.amazon.com/pt/ec2/

[49] Amazon, "Amazon Relational Database Service (RDS) – AWS," 2016. [Online]. Available: https://aws.amazon.com/pt/rds/

[50] Amazon, "Amazon Simple Storage Service (S3) – Armazenamento na nuvem," 2016. [Online]. Available: https://aws.amazon.com/pt/s3/

[51] Amazon, "Serviços em nuvem gratuitos – Nível gratuito da AWS," 2016. [Online]. Available: https://aws.amazon.com/pt/free/

[52] S. Munaf and R. Lopez, "What technology is Azure running on? Is it on Hyper-V or some other kind of virtualization technology? - Quora," 2015. [Online]. Available: https://www.quora.com/What-technology-is-Azure-running-on-Is-it-on-Hyper-V-or-some-other-kind-of-virtualization-technology

[53] Microsoft, "Customer and Partner Success Stories for Microsoft Azure," 2016. [Online]. Available: https://azure.microsoft.com/en-us/case-studies/

[54] Microsoft, "Azure Regions - Microsoft Azure," 2016. [Online]. Available: https://azure.microsoft.com/en-us/regions/

[55] Microsoft, "Virtual Machines - Linux and Azure virtual machines - Microsoft Azure," 2016. [Online]. Available: https://azure.microsoft.com/en-us/services/virtual-machines/

[56] Microsoft, "SQL Database – Cloud Database as a Service - Microsoft Azure," 2016. [Online]. Available: https://azure.microsoft.com/en-us/services/sql-database/

[57] Microsoft, "Azure Storage - Secure cloud storage - Microsoft Azure," 2016. [Online]. Available: https://azure.microsoft.com/en-us/services/storage/

[58] Microsoft, "Pricing - App Service - Microsoft Azure," 2016. [Online]. Available: https://azure.microsoft.com/en-us/pricing/details/app-service/

[59] Google, "Google Cloud Computing, Hosting Services APIs - Google Cloud Platform," 2016. [Online]. Available: https://cloud.google.com/

[60] Google, "Companies Using Google Cloud Services - Google Cloud Platform," 2016. [Online]. Available: https://cloud.google.com/customers/

[61] Google, "Global Locations - Regions & Zones - Google Cloud Platform," 2016. [Online]. Available: https://cloud.google.com/about/locations/

[62] Google, "Compute Engine - IaaS - Google Cloud Platform," 2016. [Online]. Available: https://cloud.google.com/compute/

[63] Google, "Pricing — Price Performance Leadership - Google Cloud Platform," 2016. [Online]. Available: https://cloud.google.com/pricing/

[64] Google, "Cloud SQL - MySQL Relational Database Service - Google Cloud Platform," 2016. [Online]. Available: https://cloud.google.com/sql/

[65] Google, "Bigtable - Scalable NoSQL Database Service - Google Cloud Platform," 2016. [Online]. Available: https://cloud.google.com/bigtable/

[66] Google, "Bigtable - Scalable NoSQL Database Service - Google Cloud Platform," 2016. [Online]. Available: https://cloud.google.com/bigtable/

[67] Google, "Datastore - NoSQL Schemaless Database - Google Cloud Platform," 2016. [Online]. Available: https://cloud.google.com/datastore/

[68] Google, "Cloud Storage - Online Data Storage - Google Cloud Platform," 2016. [Online]. Available: https://cloud.google.com/storage/

[69] Google, "Container Engine - Google Cloud Platform," 2016. [Online]. Available: https://cloud.google.com/container-engine/

[70] A. Colangelo, "Google Cloud vs AWS: A Comparison | Cloud Academy," 2014. [Online]. Available: http://cloudacademy.com/blog/google-cloud-vs-aws-a-comparison/

[71] G. Pollock, "DigitalOcean vs. AWS," 2016. [Online]. Available: https://www.upguard.com/articles/digitalocean-vs-aws

[72] DigitalOcean, "Customers on DigitialOcean How companies deploy and scale on DO," 2016. [Online]. Available: https://www.digitalocean.com/customers/

[73] DigitalOcean, "DigitalOcean Cloud computing designed for developers," 2016. [Online]. Available: https://www.digitalocean.com/

[74] DigitalOcean, "Compute on DigitalOcean - Droplets: fast, resizable cloud servers," 2016. [Online]. Available: https://www.digitalocean.com/products/compute/

[75] DigitalOcean, "Storage on DigitalOcean - Highly available Block Storage," 2016. [Online]. Available: https://www.digitalocean.com/products/storage/

[76] E. Minick, "DevOps - ITs Automation Revolution," 2012. [Online]. Available: http://pt.slideshare.net/Urbancode/devops-its-automation-revolution/6

[77] S. Malvik, "DevOps Is Not About Trust. - DevOps.Town," 2016. [Online]. Available: http://devops.town/devops-is-not-about-trust/

[78] M. Etmajer, "Continuous Delivery 101: Automated Deployments," 2014. [Online]. Available: http://apmblog.dynatrace.com/2014/11/18/continuous-delivery-101-automated-deployments/

[79] J. Humble and D. Farley, *Continuous Delivery - Reliable Software Releases through Build, Test, and Deployment Automation*, 2010. [Online]. Available: https://books.google.pt/books?id=6ADDuzere-YC{&}dq=continuous+delivery{&}lr={&}hl=pt-PT{&}source=gbs{_}navlinks{_}s

[80] C. Smith, "The 5 Big Benefits of Automated Deployment - Redgate Software," 2015. [Online]. Available: http://www.red-gate.com/blog/database-lifecycle-management/5-big-benefits-automated-deployment

[81] M. Fowler, "Continuous Integration," 2006. [Online]. Available: http://www.martinfowler.com/articles/continuousIntegration.html

[82] J. Robbins, "Announcing Chef - Chef Blog," 2009. [Online]. Available: https://blog.chef.io/2009/01/15/announcing-chef/

[83] A. Bertram, "Review: Chef 12 fires up devops | InfoWorld," 2016. [Online]. Available: http://www.infoworld.com/article/3108545/data-center/review-chef-12-fires-up-devops.html

[84] P. Venezia, "Review: Puppet vs. Chef vs. Ansible vs. Salt | InfoWorld," 2013. [Online]. Available: http://www.infoworld.com/article/2609482/data-center/data-center-review-puppet-vs-chef-vs-ansible-vs-salt.html?page=4

[85] Chef, "Customer Stories - Chef," 2016. [Online]. Available: https://www.chef.io/customers/

[86] J. Jackson, "Facebook uses a seasoned Chef to keep servers simmering," 2014. [Online]. Available: http://www.pcadvisor.co.uk/news/software/3495788/facebook-uses-a-seasoned-chef-to-keep-servers-simmering/

[87] Chef, "Cloud Management - Chef," 2016. [Online]. Available: https://www.chef.io/solutions/cloud-management/

[88] Chef, "Install the Chef DK - Chef Docs," 2016. [Online]. Available: https://docs.chef.io/install{_}dk.html

[89] A. Gale, "Introducing Chef Server - Getting started with Chef," 2013. [Online]. Available: http://gettingstartedwithchef.com/introducing-chef-server.html

[90] Chef, "Welcome - The resource for Chef cookbooks - Chef Supermarket," 2016. [Online]. Available: https://supermarket.chef.io/

[91] Chef, "Chef Pricing - Chef," 2016. [Online]. Available: https://www.chef.io/pricing/

[92] Puppet, "Puppet FAQ | Puppet," 2016. [Online]. Available: https://puppet.com/product/faq

[93] Upguard, "Puppet vs. Chef Revisited," 2015. [Online]. Available: https://www.upguard.com/articles/puppet-vs.-chef-revisited

[94] Puppet, "Customers - Puppet," 2016. [Online]. Available: https://puppet.com/resources?types[type{_}customer{_}story]=type{_}customer{_}story

[95] Puppet, "Installing modules - Documentation - Puppet," 2016. [Online]. Available: https://docs.puppet.com/puppet/latest/reference/modules{_}installing.html?{_}ga=1.259566921.278374008.1449699253{#}about-puppet-module

[96] Puppet, "Documentation index - Documentation - Puppet," 2016. [Online]. Available: https://docs.puppet.com/facter/3.3/

[97] K. Anderson, "A Configuration Management Rosetta Stone: Configuring Sensu With Puppet, Chef, Ansible and Salt - Postmodern Sysadmin," 2015. [Online]. Available: http://www.xkyle.com/a-configuration-management-rosetta-stone-configuring-sensu-with-puppet-chef-ansible-and-salt/

[98] Puppet, "Marionette Collective - Documentation - Puppet," 2016. [Online]. Available: https://docs.puppet.com/mcollective/

[99] Puppet, "Managing code with Code Manager - Documentation - Puppet," 2016. [Online]. Available: https://docs.puppet.com/pe/latest/code{_}mgr.html

[100] Puppet, "Installing Puppet agent on Linux - Documentation - Puppet," 2016. [Online]. Available: https://docs.puppet.com/puppet/latest/reference/install{_}linux.html

[101] Puppet, "Overview of Puppet's architecture - Documentation - Puppet," 2016. [Online]. Available: https://docs.puppet.com/puppet/4.6/reference/architecture.html

[102] Fred, "puppet - Configuration management: push versus pull based topology - Server Fault," 2014. [Online]. Available: https://serverfault.com/questions/568187/configuration-management-push-versus-pull-based-topology/568372{#}568372?newreg=b878ae8c815e4c7e914a7e12c2d9a869

[103] Puppet, "Puppet open source projects - 40+ including Beaker, Facter and Hiera," 2016. [Online]. Available: https://puppet.com/product/open-source-projects

[104] Puppet, "Puppet Enterprise pricing," 2016. [Online]. Available: https://puppet.com/product/pricing

[105] F. Dog, "puppet - Configuration management: push versus pull based topology - Server Fault," 2014. [Online]. Available: https://serverfault.com/questions/568187/configuration-management-push-versus-pull-based-topology/568372{#}568372?newreg=b878ae8c815e4c7e914a7e12c2d9a869

[106] M. DeHaan, *An Interview with Ansible Author Michael DeHaan*. Colo and Cloud, 2012. [Online]. Available: http://www.coloandcloud.com/editorial/an-interview-with-ansible-author-michael-dehaan/

[107] Ansible, "Playbooks - Ansible Documentation," 2016. [Online]. Available: http://docs.ansible.com/ansible/playbooks.html

[108] M. Maas, *Ansible in Depth*, 2016. [Online]. Available: http://cdn2.hubspot.net/hub/330046/file-480366556-pdf/pdf{_}content/Ansible{_}in{_}Depth.pdf?t=1390852822000

[109] Server for Hackers, "An Ansible Tutorial - Servers for Hackers," 2014. [Online]. Available: https://serversforhackers.com/an-ansible-tutorial

[110] Ansible, "Getting | Started," 2014. [Online]. Available: http://docs.ansible.com/intro{_}getting{_}started.html

[111] G. Carvalho, "Puppet vs Ansible?" 2016. [Online]. Available: http://gutocarvalho.net/blog/2016/06/02/puppet-vs-ansible/

[112] Ansible, "Cloud Modules - Ansible Documentation," 2016. [Online]. Available: http://docs.ansible.com/ansible/list{_}of{_}cloud{_}modules.html

[113] Ansible, "Ansible Tower - Ansible.com," 2016. [Online]. Available: https://www.ansible.com/tower

[114] Upguard, "Ansible vs. Ansible Tower," 2016. [Online]. Available: https://www.upguard.com/articles/ansible-vs.-ansible-tower

[115] Ansible, "Ansible - Tower Pricing," 2016. [Online]. Available: https://www.ansible.com/pricing

[116] J. McCoy, "Ansible Review: It's like the diet version of Chef," 2015. [Online]. Available: https://www.trustradius.com/reviews/ansible-2015-10-06-11-36-56

[117] Upguard, "Top 5 Best and Worst Attributes of Ansible," 2014. [Online]. Available: https://www.upguard.com/articles/top-5-best-and-worst-attributes-of-ansible

[118] M. Hashimoto, "Atlas - HashiCorp," 2014. [Online]. Available: https://www.hashicorp.com/blog/atlas.html

[119] Market Wire, "HashiCorp Launches Atlas to Power DevOps Application Delivery on Any Infrastructure – Public, Private, and Hybrid," 2014. [Online]. Available: http://www.marketwired.com/press-release/hashicorp-launches-atlas-power-devops-application-delivery-on-any-infrastructure-public-1975584.htm

[120] Vagrant, "About - Vagrant by HashiCorp," 2016. [Online]. Available: https://www.vagrantup.com/about.html

[121] Packer, "Packer by HashiCorp," 2016. [Online]. Available: https://www.packer.io/

[122] Terraform, "Terraform by HashiCorp," 2016. [Online]. Available: https://www.terraform.io/

[123] Consul, "Consul by HashiCorp," 2016. [Online]. Available: https://www.consul.io/

[124] D. Bryant, "HashiCorp Publicly Release Atlas, a Version Control System for Infrastructure," 2015. [Online]. Available: https://www.infoq.com/news/2015/07/hashicorp-atlas-public-release

[125] Atlas, "Continuous Deployment of Immutable Infrastructure - Help | Atlas by HashiCorp," 2016. [Online]. Available: https://atlas.hashicorp.com/help/intro/use-cases/continuous-deployment-of-immutable-infrastructure

[126] V. Partington, "Deployment automation vs. server provisioning," 2010. [Online]. Available: https://blog.xebialabs.com/2010/12/20/deployment-automation-vs-server-provisioning/

[127] G. Bevin, "DevOps Introduction and Guide | zeroturnaround.com," 2013. [Online]. Available: https://zeroturnaround.com/rebellabs/pragmatic-devops-virtualization-provisioning-with-vagrant-chef/

[128] Docker, "Docker - Build, Ship, and Run Any App, Anywhere," 2016. [Online]. Available: https://www.docker.com/

[129] S. Vaughan-Nichols, "What is Docker and why is it so darn popular? | ZDNet," 2014. [Online]. Available: http://www.zdnet.com/article/what-is-docker-and-why-is-it-so-darn-popular/

[130] D. Merkel, "Docker - a lightweight Linux containers for consistent development and deployment," *Linux Journal*, 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2600241

[131] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An updated performance comparison of virtual machines and Linux containers," *2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 171–172, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7095802

[132] C. Pahl, "Containerization and the PaaS Cloud," *IEEE Cloud Computing*, vol. 2, no. 3, pp. 24–31, 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7158965

[133] Datadog, "8 surprising facts about real Docker adoption - Datadog," 2016. [Online]. Available: https://www.datadoghq.com/docker-adoption/

[134] M. Peacock, *Creating Development Environments with Vagrant*, 2015. [Online]. Available: https://books.google.pt/books?hl=pt-PT{&}lr={&}id=v{_}MwBwAAQBAJ{&}oi=fnd{&}pg=PP1{&}dq=vagrant+hashicorp{&}ots=5TIJ1y7ZRG{&}sig=bVuKu2jMux7j8PI4R0NUtp6s5uo{&}redir{_}esc=y{#}v=onepage{&}q=vagranthashicorp{&}f=false

[135] M. Hashimoto, "Stronger DevOps Culture with Puppet and Vagrant | Puppet," 2012. [Online]. Available: https://puppet.com/blog/stronger-devops-culture-puppet-and-vagrant

[136] S. Lowe, "Using Docker with Vagrant - Scott's Weblog - The weblog of an IT pro specializing in virtualization, networking, open source, and cloud computing," 2015. [Online]. Available: http://blog.scottlowe.org/2015/02/10/using-docker-with-vagrant/

[137] Vagrant, "VirtualBox Provider - Vagrant by HashiCorp," 2016. [Online]. Available: https://www.vagrantup.com/docs/virtualbox/

[138] VirtualBox, "Oracle VM VirtualBox," 2016. [Online]. Available: https://www.virtualbox.org/

[139] Vagrant, "VMware Provider - Vagrant by HashiCorp," 2016. [Online]. Available: https://www.vagrantup.com/docs/vmware/

[140] C. Wolfe, "OmniTI - Our Experiences with Chef: Using Vagrant and Chef to Enable Quality Assurance," 2013. [Online]. Available: https://omniti.com/seeds/seeds-our-experiences-with-chef-enabling-software-quality-assurance

[141] M. Hashimoto and S. Hykes, "Should I use Vagrant or Docker for creating an isolated environment? - Stack Overflow," 2014. [Online]. Available: http://stackoverflow.com/questions/16647069/should-i-use-vagrant-or-docker-for-creating-an-isolated-environment

[142] R. Patton, *Software Testing*, 2001.

[143] B. Marick, *When Should a Test Be Automated?* StickyMinds.com, 1998. [Online]. Available: http://www.stickyminds.com/sitewide.asp?Function=edetail{&}ObjectType=ART{&}ObjectId=2010

[144] E. Dustin, J. Rashka, and J. Paul, *Automated software testing : introduction, management, and performance*. Addison-Wesley, 1999.

[145] M. Rouse, "What is automated software testing? - Definition from WhatIs.com," 2014. [Online]. Available: http://searchsoftwarequality.techtarget.com/definition/automated-software-testing

[146] I. Foundation, "What is Unit testing?" 2016. [Online]. Available: http://istqbexamcertification.com/what-is-unit-testing/

[147] A. Kolawa, "Unit Testing Best Practices," 2009. [Online]. Available: http://www.parasoft.com/unit-testing-best-practices

[148] JUnit, "JUnit - About," 2016. [Online]. Available: http://junit.org/junit4/index.html

[149] T. Kaczanowscy, "Why TestNG?" 2016. [Online]. Available: http://kaczanowscy.pl/tomek/sites/default/files/testng{_}vs{_}junit.txt.slidy{_}.html{#}(1)

[150] T. Weiss, "We Analyzed 30k GitHub Projects - Here Are The Top 100 Libraries," 2013. [Online]. Available: http://blog.takipi.com/we-analyzed-30000-github-projects-here-are-the-top-100-libraries-in-java-js-and-ruby/

[151] TestNG, "TestNG - Welcome," 2016. [Online]. Available: http://testng.org/doc/index.html

[152] M. Stone, "java - JUnit vs TestNG - Stack Overflow," 2008. [Online]. Available: https://stackoverflow.com/questions/6658/junit-vs-testng

[153] A. Zhitnitsky, "JUnit vs TestNG: Which Testing Framework Should You Choose? - Takipi Blog," 2016. [Online]. Available: http://blog.takipi.com/junit-vs-testng-which-testing-framework-should-you-choose/

[154] Techopedia, "What is an End-to-End Test? - Definition from Techopedia," 2016. [Online]. Available: https://www.techopedia.com/definition/7035/end-to-end-test

[155] yearofmoo, "Advanced Testing and Debugging in AngularJS," 2013. [Online]. Available: http://www.yearofmoo.com/2013/09/advanced-testing-and-debugging-in-angularjs.html

[156] Protractor, "Protractor - end to end testing for AngularJS," 2016. [Online]. Available: http://www.protractortest.org/{#}/

[157] Protractor, "Protractor - end to end testing for AngularJS," 2016. [Online]. Available: http://www.protractortest.org/{#}/protractor-setup

[158] Protractor, "Tutorial - Protractor," 2016. [Online]. Available: http://www.protractortest.org/{#}/tutorial

[159] Nightwatch, "Nightwatch.js | Node.js powered End-to-End testing framework," 2016. [Online]. Available: http://nightwatchjs.org/

[160] Nightwatch, "Getting Started - Installation - Nightwatch," 2016. [Online]. Available: http://nightwatchjs.org/getingstarted{#}installation

[161] Nightwatch, "Developer Guide - Nightwatch," 2016. [Online]. Available: http://nightwatchjs.org/guide

[162] K. Aida, "Effect of Job Size Characteristics on Job Scheduling Performance," 2000.

[163] CogNiTioN, *Newbie Introduction to cron*. Unixgeeks.org, 1999. [Online]. Available: http://www.unixgeeks.org/security/newbie/unix/cron-1.html

[164] Ubuntu, *Ubuntu Cron Howto*. Help.ubuntu.com, 2016. [Online]. Available: http://help.ubuntu.com/community/CronHowto

[165] wdt, "crontab.guru - the cron schedule expression editor," 2016. [Online]. Available: http://crontab.guru/

[166] Chronos, "Chronos - Fault tolerant job scheduler for Mesos," 2016. [Online]. Available: https://mesos.github.io/chronos/

[167] Chronos, "Chronos - Setting Up and Running Chronos," 2016. [Online]. Available: https://mesos.github.io/chronos/docs/

[168] H. Shoff, "Chronos - A Replacement for Cron - Airbnb Engineering," 2013. [Online]. Available: http://nerds.airbnb.com/introducing-chronos/

[169] I. C., "torque - Outgrowing cron - what is the next scheduler? - Server Fault," 2011. [Online]. Available: http://serverfault.com/questions/277821/outgrowing-cron-whats-the-next-scheduler

[170] M. Kavis, "Nagios is not a monitoring strategy - DevOps.com," 2014. [Online]. Available: http://devops.com/2014/04/15/nagios-monitoring-strategy/

[171] Dynatrace, "Dynatrace Ruxit - Cloud native monitoring | Dynatrace," 2016. [Online]. Available: https://www.dynatrace.com/platform/offerings/ruxit/

[172] Ruxit, "Dynatrace SaaS and Managed (Ruxit) - Cloud native monitoring - Dynatrace," 2016. [Online]. Available: https://www.dynatrace.com/platform/offerings/ruxit/

[173] T. Doherty, "Will NewRelic server monitoring affect my server performance? - Infrastructure Monitoring - New Relic Online Technical Community," 2014. [Online]. Available: https://discuss.newrelic.com/t/will-newrelic-server-monitoring-affect-my-server-performance/4051/2

[174] S. Mappic, "The Real Overhead of Managing Application Performance - Application Performance Monitoring Blog | AppDynamics," 2011. [Online]. Available: https://blog.appdynamics.com/apm/the-real-overhead-of-managing-application-performance/

[175] Ruxit, "APM - Dynatrace," 2016. [Online]. Available: https://www.dynatrace.com/capabilities/application-performance-management/

[176] New Relic, "Application Performance Monitoring - New Relic APM," 2016. [Online]. Available: https://newrelic.com/application-monitoring

[177] C. Lever, "Benefits of Using RUM with Synthetic Monitoring - Rigor," 2014. [Online]. Available: http://rigor.com/blog/2014/12/benefits-using-rum-w-synthetic

[178] ITCentralStation, "Application Performance Management (APM) Vendors - IT Central Station," 2016. [Online]. Available: https://www.itcentralstation.com/categories/application-performance-management-apm{#}top{_}rated

[179] New Relic, "Miniclip Keeps the Fun Going with New Relic | New Relic Resource Center," 2016. [Online]. Available: https://newrelic.com/case-studies/miniclip

[180] New Relic, "New Relic Servers for Linux - New Relic Documentation," 2016. [Online]. Available: https://docs.newrelic.com/docs/servers/new-relic-servers-linux/getting-started/new-relic-servers-linux

[181] New Relic, "Installing Servers for Windows - New Relic Documentation," 2016. [Online]. Available: https://docs.newrelic.com/docs/servers/new-relic-servers-windows/installation-configuration/installing-servers-windows

[182] New Relic, "Welcome to New Relic APM - New Relic Documentation," 2016. [Online]. Available: https://docs.newrelic.com/docs/apm/new-relic-apm/getting-started/welcome-new-relic-apm

[183] New Relic, "Real User Monitoring (RUM) - New Relic Browser," 2016. [Online]. Available: https://newrelic.com/browser-monitoring

[184] New Relic, "Application Monitoring - New Relic Synthetics | New Relic Synthetics," 2016. [Online]. Available: https://newrelic.com/synthetics

[185] New Relic, "Mobile App Monitoring for iOS and Android - New Relic Mobile," 2016. [Online]. Available: https://newrelic.com/mobile-monitoring

[186] New Relic, "New Relic Pricing for Application Performance Monitoring - New Relic APM," 2016. [Online]. Available: https://newrelic.com/application-monitoring/pricing

[187] garethr, "Buy vs Build your Monitoring System," 2014. [Online]. Available: http://www.morethanseven.net/2014/02/16/buy-vs-build-your-monitoring-system/

[188] Ruxit, "APM powered by artificial intelligence - Dynatrace," 2016. [Online]. Available: https://www.dynatrace.com/capabilities/artificial-intelligence/

[189] Ruxit, "Customers - Dynatrace," 2016. [Online]. Available: https://www.dynatrace.com/company/customers/

[190] Ruxit, "Dynatrace Help - How do I install Dynatrace OneAgent," 2016. [Online]. Available: https://help.dynatrace.com/get-started/install/how-do-i-install-dynatrace-agent/

[191] Ruxit, "Real user monitoring (RUM) - Dynatrace," 2016. [Online]. Available: https://www.dynatrace.com/capabilities/real-user-monitoring/

[192] Ruxit, "Mobile app performance monitoring - Dynatrace," 2016. [Online]. Available: https://www.dynatrace.com/capabilities/mobile-app-monitoring/

[193] Ruxit, "Pricing - Simple, flexible, and predictable - Dynatrace," 2016. [Online]. Available: https://www.dynatrace.com/pricing/

[194] Pingdom, "Why choose Pingdom for monitoring," 2016. [Online]. Available: https://www.pingdom.com/company/why-pingdom

[195] Deathstarr, "Other - Pingdom no longer free - The Admin Zone," 2015. [Online]. Available: https://theadminzone.com/threads/pingdom-no-longer-free.137844/

[196] Pingdom, "Pingdom - Website Monitoring Made Easy," 2015. [Online]. Available: https://www.pingdom.com/planfree?mkt{_}tok=3RkMMJWWfF9wsRonv6jKZKXonjHpfsX56e4qWqW1lMI{%}2F0ER3fOvrPUfGjI4DScNlI{%}2BSLDwE

[197] StatusCake, "StatusCake - The Alternative to Pingdom," 2016. [Online]. Available: https://www.statuscake.com/alternative-to-pingdom/

[198] StatusCake, "StatusCake - Website Monitoring - Website Monitoring and Downtime Alerts," 2016. [Online]. Available: https://www.statuscake.com/

[199] StatusCake, "Pricing - StatusCake - Website Monitoring," 2016. [Online]. Available: https://www.statuscake.com/pricing/

[200] StatusCake, "What Locations Do You Have - Knowledge Base," 2016. [Online]. Available: https://www.statuscake.com/kb/knowledge-base/what-locations-do-you-have/

[201] T. Viethai, "Daemon Showdown - Upstart vs. Runit vs. Systemd vs. Circus vs. God - centosvn," 2014. [Online]. Available: http://centos-vn.blogspot.pt/2014/06/daemon-showdown-upstart-vs-runit-vs.html

[202] G. Noronha, "Entenda o systemd vs upstart - Gustavo Noronha (kov)," 2014. [Online]. Available: https://blog.kov.eti.br/2014/02/entenda-o-systemd-vs-upstart/

[203] Upstart, "upstart - event-based init daemon," 2016. [Online]. Available: http://upstart.ubuntu.com/

[204] M. Saunders, *Linux 101: Get the most out of systemd*. Linux Voice, 2015. [Online]. Available: https://www.linuxvoice.com/linux-101-get-the-most-out-of-systemd/

[205] Mozilla, "The Web Console - Circus 0.13.1 documentation," 2016. [Online]. Available: http://circus.readthedocs.io/en/latest/for-ops/circusweb/

[206] K. Kent and M. Souppaya, "Guide to Computer Security Log Management Recommendations of the National Institute of Standards and Technology," 2002.

[207] M. Rothman, "Leveraging Log Data for Better Security," 2007. [Online]. Available: http://www.eventtracker.com/newsletters/leveraging-log-data-for-better-security/

[208] D. Harris, "Gigaom - How Splunk Is Riding IT Search Toward an IPO," 2010. [Online]. Available: https://gigaom.com/2010/12/17/how-splunk-is-riding-it-search-toward-an-ipo/

[209] Splunk, "Splunk makes machine data accessible, usable and valuable to everyone." 2016. [Online]. Available: http://www.splunk.com/en{_}us/resources/machine-data.html

[210] Splunk, "Turn machine data into insights to see into your business and IT systems - Splunk," 2016. [Online]. Available: https://www.splunk.com/en{_}us/resources/operational-intelligence.html

[211] Splunk, "Manual - Splunk Knowledgebase," 2016. [Online]. Available: http://docs.splunk.com/Documentation/Splunk/6.5.0/Installation/Whatsinthismanual

[212] Splunk, "Splunk Product - Splunk," 2016. [Online]. Available: https://www.splunk.com/en{_}us/products.html

[213] Splunk, "Pricing - Splunk," 2016. [Online]. Available: https://www.splunk.com/en{_}us/products/pricing.html

[214] Loggly, "Loggly Index," 2016. [Online]. Available: https://www.loggly.com/

[215] A. Williams, *Loggly, A Splunk Competitor, Raises $10.5M For Cloud-Centric Approach To Log Management*, 2013. [Online]. Available: http://techcrunch.com/2013/09/03/loggly-a-splunk-competitor-raises-10-5m-for-cloud-centric-approach-to-log-management/

[216] J. Novet, *Loggly grows as it convinces more companies to keep their logs in the cloud (exclusive)*, 2014. [Online]. Available: http://venturebeat.com/2014/04/07/loggly-momentum-exclusive/

[217] J. Gold, *Cisco one of investors in log management startup Loggly*, 2013. [Online]. Available: http://www.networkworld.com/article/2169586/lan-wan/cisco-one-of-investors-in-log-management-startup-loggly.html

[218] Loggly, "Logging Setup - Loggly," 2016. [Online]. Available: https://www.loggly.com/docs/logging-setup/

[219] Loggly, "Loggly Pricing - Compare Loggly Plans and Features - Loggly |," 2016. [Online]. Available: https://www.loggly.com/plans-and-pricing/

[220] S. Banon, *The Future of Compass & ElasticSearch*, 2010. [Online]. Available: http://thedudeabides.com/articles/the{_}future{_}of{_}compass

[221] A. Yigal, "The Complete Guide to the ELK Stack - Logz.io." [Online]. Available: http://logz.io/learn/complete-guide-elk-stack/

[222] Elastic, "Elasticsearch," 2016.

[223] Elastic, "Logstash," 2016.

[224] Elastic, "Kibana," 2016.

[225] Logit, "Hosted ELK as a Service, using Kibana 4, Logstash 2 and ElasticSearch 2," 2016. [Online]. Available: https://logit.io/

[226] K. MacInnis, *StumbleUpon | Developer Blog*. StumbleUpon.com, 2011. [Online]. Available: http://www.stumbleupon.com/blog/dev/searching-for-serendipity/

[227] H. Karau and A. Alix, "foursquare now uses Elastic Search (and on a related note: Slashem also works with Elastic Search)! | Foursquare Engineering Blog," 2012. [Online]. Available: http://engineering.foursquare.com/2012/08/09/foursquare-now-uses-elastic-search-and-on-a-related-note-slashem-also-works-with-elastic-search/

[228] G. Horanyi, *Needle in a haystack - Using Elasticsearch to run the Large Hadron Collider of CERN*. medium.com, 2014. [Online]. Available: https://medium.com/@ghoranyi/needle-in-a-haystack-873c97a99983

[229] N. Craver, *What it takes to run Stack Overflow*, 2013. [Online]. Available: http://nickcraver.com/blog/2013/11/22/what-it-takes-to-run-stack-overflow/

[230] M. Abis, "6 Questions To 4 Experts on," 2015.

[231] V. Farcic, "Service Discovery: Zookeeper vs etcd vs Consul," 2015. [Online]. Available: https://technologyconversations.com/2015/09/08/service-discovery-zookeeper-vs-etcd-vs-consul/

[232] Gavin, "Apache Software Foundation," 2012. [Online]. Available: https://cwiki.apache.org/confluence/display/ZOOKEEPER/Index;jsessionid=ACEBE76A2997331C51BA52DC862206FE

[233] B. Reed, "ProjectDescription - Apache ZooKeeper - Apache Software Foundation," 2012. [Online]. Available: https://cwiki.apache.org/confluence/display/ZOOKEEPER/ProjectDescription

[234] H. Netto, L. C. Lung, M. Correia, and A. F. Luiz, "Replicação de Máquinas de Estado em containers no Kubernetes: uma Proposta de Integração."

[235] Consul, "Consul vs. ZooKeeper, doozerd, etcd - Consul by HashiCorp," 2016. [Online]. Available: https://www.consul.io/intro/vs/zookeeper.html

[236] S. P. Mirashe and N. V. Kalyankar, "Cloud Computing," *Communications of the ACM*, vol. 51, no. 7, p. 9, 2010. [Online]. Available: http://arxiv.org/abs/1003.4074

[237] T. B. Sousa, "DevOps Patterns for Software Orchestration on Public and Private Clouds," 2015.

[238] H. Lim, S. Babu, J. Chase, and S. Parekh, "Automated control in cloud computing: challenges and opportunities," *Proceedings of the 1st workshop on Automated control for datacenters and clouds*, p. 13–18, 2009. [Online]. Available: http://portal.acm.org/citation.cfm?id=1555271.1555275

[239] D. T. Connors, "Software development methodologies and traditional and modern information systems," *ACM SIGSOFT Software Engineering Notes*, vol. 17, no. 2, pp. 43–49, 1992.

[240] F. Erich, C. Amrit, and M. Daneva, "Report: DevOps Literature Review," no. October, p. 27, 2014.

[241] P. Debois, "Agile infrastructure and operations: How infra-gile are you?" *Proceedings - Agile 2008 Conference*, pp. 202–207, 2008.

[242] Saugatuck Technology, "Why DevOps Matters : Practical Insights on Managing Complex & Continuous Change," no. October, 2014.

[243] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[244] D. Namiot and M. Sneps-Sneppe, "On Micro-services Architecture," *International Journal of Open Information Technologies*, vol. 2, no. 9, pp. 24–27, 2014. [Online]. Available: http://injoit.org/index.php/j1/article/view/139

[245] Techopedia, "What is Virtualization?" 2015. [Online]. Available: https://www.techopedia.com/definition/719/virtualization

[246] Docker, "What is Docker," 2015. [Online]. Available: https://www.docker.com/what-docker

[247] P. Labs, "Puppet Enterprise," 2015. [Online]. Available: https://puppetlabs.com/puppet/puppet-enterprise

[248] Chef Docs, "All about chef," 2015. [Online]. Available: https://docs.chef.io/

[249] Core Blog, "Core Blog," 2015. [Online]. Available: https://coreos.com/blog/

[250] Kubernetes, "What is Kubernetes?" 2015. [Online]. Available: http://kubernetes.io/v1.1/docs/whatisk8s.html

[251] Apache Mesos, "Apache Mesos," 2015. [Online]. Available: http://mesos.apache.org/

[252] Airbnb Engineering, "SmartStack: Service Discovery in the Cloud," 2015. [Online]. Available: http://nerds.airbnb.com/smartstack-service-discovery-cloud/

[253] E. Griffith, "What is Cloud Computing?" 2015. [Online]. Available: http://www.pcmag.com/article2/0,2817,2372163,00.asp

[254] J. Stella, "An introduction to immutable infrastructure - O'Reilly Media," 2015. [Online]. Available: https://www.oreilly.com/ideas/an-introduction-to-immutable-infrastructure

[255] S. Bigelow, "What is immutable infrastructure? - Definition from WhatIs.com," 2016. [Online]. Available: http://searchitoperations.techtarget.com/definition/immutable-infrastructure

[256] A. Aram, *FaaS, PaaS, and the Benefits of the Serverless Architecture*. InfoQ, 2016. [Online]. Available: https://www.infoq.com/news/2016/06/faas-serverless-architecture

[257] T. Janczuk, "What is Function as a Service (FaaS)?" 2016. [Online]. Available: https://tomasz.janczuk.org/2016/06/what-is-function-as-a-service.html

[258] Dougvj, "operating system - What is the difference between Full, Para and Hardware assisted virtualiazation? - Stack Overflow," 2016. [Online]. Available: http://stackoverflow.com/questions/21462581/what-is-the-difference-between-full-para-and-hardware-assisted-virtualiazation

[259] Dynatrace, "Dynatrace Ruxit - Cloud native monitoring | Dynatrace," 2016. [Online]. Available: https://www.dynatrace.com/platform/offerings/ruxit/

[260] M. Rouse, "What is paravirtualization? - Definition from WhatIs.com," 2006. [Online]. Available: http://searchservervirtualization.techtarget.com/definition/paravirtualization

[261] G. Schade, "New Relic APM Review By Guido Schade, Senior Unix Engineer + Managing Dir | IT Central Station," 2014. [Online]. Available: https://www.itcentralstation.com/product{_}reviews/new-relic-apm-review-31513-by-guido-schade

[262] J. Kowall, "Ruxit is an impressive new SaaS delivered APM offering - Jonah Kowall," 2014. [Online]. Available: http://blogs.gartner.com/jonah-kowall/2014/10/07/ruxit-is-an-impressive-new-saas-delivered-apm-offering/

[263] A. Balkan, "Service Discovery: Zookeeper vs etcd vs Consul | Technology Conversations," 2015. [Online]. Available: https://technologyconversations.com/2015/09/08/service-discovery-zookeeper-vs-etcd-vs-consul/

[264] F. Paul, "Gene Kim Explains Why DevOps Matters," 2015. [Online]. Available: https://blog.newrelic.com/2015/06/24/gene-kim-why-devops-matters/

[265] J. Lyman, "DevOps mixing dev, ops, agile, cloud, open source and business - 451 CAOS Theory," 2010. [Online]. Available: https://blogs.the451group.com/opensource/2010/03/03/devops-mixing-dev-ops-agile-cloud-open-source-and-business/

[266] J. Hand, "The IT Culture War: The Struggle to Adopt DevOps | WIRED," 2015. [Online]. Available: http://www.wired.com/insights/2015/03/culture-war-struggle-adopt-devops/

[267] Anil, "DevOps Adoption and Challenges," 2016. [Online]. Available: https://anilkalose.wordpress.com/2016/01/08/devops-adoption-and-challenges/

[268] D. Placette, "51 Best DevOps Tools for DevOps Engineers | ProfitBricks Blog," 2015. [Online]. Available: https://blog.profitbricks.com/51-best-devops-tools-for-devops-engineers/

[269] E. Wilinski, "DevOps Best Practices: Finding the Right Tools," 2014. [Online]. Available: https://blog.newrelic.com/2014/06/02/devops-tools/

[270] M. Jones, "Dev or Ops?" 2010. [Online]. Available: http://www.slideshare.net/geekle/devops-5348895/4-Dev{_}or{_}Opsbr{_}Software{_}is

[271] M. Ducy, "Chef Analytics and Slack = Awesome - Chef Blog," 2015. [Online]. Available: https://blog.chef.io/2015/07/24/chef-analytics-slack-awesome/

[272] B. Hemphill, "What is difference between docker, puppet, chef and vagrant? - Quora," 2015. [Online]. Available: https://www.quora.com/What-is-difference-between-docker-puppet-chef-and-vagrant

[273] H. Langeveld, "What is the difference between Docker and Vagrant? When should you use each one? - Quora," 2014. [Online]. Available: https://www.quora.com/What-is-the-difference-between-Docker-and-Vagrant-When-should-you-use-each-one

[274] M. Korsak, "Configuration Management: 4 Tools Reviewed - Linode Cube - Medium," 2016. [Online]. Available: https://medium.com/linode-cube/configuration-management-4-tools-reviewed-7233eb44e278{#}.fzaq4z47j

[275] Software Insider, "Xen vs KVM vs VirtualBox - Comparison of Open Source Virtualization Software." [Online]. Available: http://virtualization.softwareinsider.com/saved{_}compare/Xen-vs-KVM-vs-VirtualBox-Comparison-of-Open-Source-Virtualization-Software

[276] E. Siebert, "Choosing vSphere vs. Hyper-V vs. XenServer," 2010. [Online]. Available: http://searchitchannel.techtarget.com/feature/Choosing-vSphere-vs-Hyper-V-vs-XenServer

[277] G. Khairallah, "Amazon AWS Review By George Khairallah, Chief Technology Officer | IT Central Station," 2016. [Online]. Available: https://www.itcentralstation.com/product{_}reviews/amazon-aws-review-32224-by-george-khairallah

[278] T. Pott, "If hypervisor is commodity, why is VMware still on top?" 2015. [Online]. Available: http://www.theregister.co.uk/2015/04/23/vms{_}compared/?page=1

[279] M. Finnegan, "Microsoft Azure vs Amazon AWS public cloud comparison | IT Vendors | Computerworld UK," 2016. [Online]. Available: http://www.computerworlduk.com/it-vendors/microsoft-azure-vs-amazon-aws-public-cloud-comparison-which-cloud-is-best-for-enterprise-3624848

[280] W. Van Winkle, "Azure vs. AWS: Cloud Platform Comparison - Tom's IT Pro," 2015. [Online]. Available: http://www.tomsitpro.com/articles/azure-vs-aws-cloud-comparison,2-870.html

[281] S. Vaughan-Nichols, "Google Cloud Platform Review Rating | PCMag.com," 2015. [Online]. Available: http://www.pcmag.com/article2/0,2817,2496296,00.asp

[282] TechTarget, "Explain software testing benefits to the executives." [Online]. Available: http://searchsoftwarequality.techtarget.com/video/Explain-software-testing-benefits-to-the-executives

[283] S. McLaughlin, "Join the darkside: Selenium testing with Nightwatch.js," 2014. [Online]. Available: http://pt.slideshare.net/sethmcl/join-the-darkside-nightwatchjs

[284] VMware, "VMware Understanding Full Virtualization, Paravirtualization, and Hardware Assist," 2007.

[285] VMware, "Virtualization Overview," 2006.

[286] L. Poettering, T. Leemhuis, and K. Sievers, *Control Centre: The systemd Linux init system*. The H, 2012. [Online]. Available: http://h-online.com/-1565543

[287] G. J. Myers, *The art of software testing*. Wiley, 1979.

[288] C. Kaner, J. L. Falk, and H. Q. Nguyen, *Testing computer software*. New York, et al: Wiley, 1999.

[289] D. Gelperin and B. Hetzel, "The Growth of Software Testing," *CACM*, vol. 31, no. 6, 1988.

[290] G. T. Laycock, "The Theory and Practice of Specification Based Software Testing," 1993. [Online]. Available: http://www.mcs.le.ac.uk/people/gtl1/thesis.ps.gz

[291] L. Gurses, *Agile 101: What is Continuous Integration?*, 2007. [Online]. Available: http://www.jacoozi.com/blog/?p=18

[292] M. Fewster and D. Graham, *Software test automation : effective use of test execution tools*. Addison-Wesley, 1999.

[293] E. Dustin, T. Garrett, and B. Gauf, *Implementing automated software testing : how to save time and lower costs while raising quality*. Addison-Wesley, 2009.

[294] D. J. Mosley and B. A. Posey, *Just enough software test automation*. Yourdon Press, 2002. [Online]. Available: http://www.amazon.com/Just-Enough-Software-Test-Automation/dp/0130084689/ref=sr{_}1{_}5?s=books{&}ie=UTF8{&}qid=1337627825{&}sr=1-5

[295] D. Huizinga and A. Kolawa, *Automated defect prevention : best practices in software management*. Wiley-Interscience, 2007. [Online]. Available: http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470042125.html

[296] E. G. Kent Beck, *JUnit Cookbook*. junit.sourceforge.net. [Online]. Available: http://junit.sourceforge.net/doc/cookbook/cookbook.htm

[297] C. Metz, "The Chef, the Puppet, and the Sexy IT Admin," *Wired*, 2011. [Online]. Available: http://www.wired.com/2011/10/chef{_}and{_}puppet/

[298] I. Kumar, *Understanding Chef and Writing Cookbooks*. TO THE NEW, 2014. [Online]. Available: http://www.tothenew.com/blog/understanding-chef-and-writing-cookbooks/

[299] A. Sharp-Paul, *Puppet vs. Chef - The Battle Wages On*, 2013. [Online]. Available: http://www.scriptrock.com/blog/puppet-vs-chef-battle-wages

[300] C. Lueninghoener, "Getting Started with Configuration Management," *;login:*, vol. 36, no. 2, 2011. [Online]. Available: https://www.usenix.org/system/files/login/articles/105457-Lueninghoener.pdf

[301] Ansible, *Ansible FAQ*, 2016. [Online]. Available: http://docs.ansible.com/faq.html

[302] R. Hat, *The Benefits of Agentless Architecture*, 2016. [Online]. Available: http://cdn2.hubspot.net/hub/330046/file-479013288-pdf/pdf{_}content/The{_}Benefits{_}of{_}Agentless{_}Architecture.pdf?t=1390852839000

[303] Ansible, "Inventory | Ansible," 2014. [Online]. Available: http://docs.ansible.com/intro{_}inventory.html

[304] Ansible, "Playbooks | Ansible," 2014. [Online]. Available: http://docs.ansible.com/playbooks.html

[305] Elastic, "Elastic - Home," 2016.

[306] K. Subramanian, "7 critical things you should look for in an APM solution," 2014. [Online]. Available: http://karunsubramanian.com/apm/7-critical-things-you-should-look-for-in-an-apm-solution/

[307] V. Rajadhyaksha, "APM - My experiences of Application Performance Monitoring using New Relic vs dynaTrace," 2015. [Online]. Available: https://technicalmumbojumbo.wordpress.com/2015/10/22/apm-newrelic-dynatrace-comparison/

[308] Pingdom, "Web Performance Management - Pingdom," 2016. [Online]. Available: https://www.pingdom.com/product