[1] **Ramasankar Molleti**

End-To-End Cloud Infrastructure Automation

Journal of Electrical Systems

Abstract: - The dispersed cloud has transformed the ways of organization IT management through flexible provision resources over the internet, therefore changing IT operational models. Automation is considered a key enabler for cloud environments as it helps in the deployment, scalability, development, testing and production. Consequently, end-to-end automation in cloud infrastructure is researched in this paper where the focused tools are AWS CloudFormation, Puppet, Ansible, Chef, Kubernetes, Terraform, Azure Automation, SaltStack, VCM, CFEngine, and Foreman. It reviews their deployments in various clouds and talks about issues like security and integration difficulties. The goal of this research would be to provide pragmatic knowledge to working IT professionals in order to work on cloud automation technologies more efficiently.

Keywords: Cloud automation, scalability, AWS, Azure, GCP

1.0 Introduction

1.1 Overview

Effectively, cloud computing has completely changed the nature of the management and deployment of Information Technology in various organizations. It makes it possible to provision resources over the internet in an elastic fashion thus changing the fundamental models of IT substantially. Cloud infrastructure consists of Internet-supported services such as storage, servers, databases, networking, and many more from the CSPs such as AWS, MS Azure, and GCP [2].

1.2 Importance

Cloud automation enables the modern enterprise environment within cloud infrastructure because of the ability to achieve operational optimization in the organization. Some of the benefits that organizations can get through automated infrastructure include an effective means of implementing rapid resource configuration and deployment, and scaling of resources, conformity of the infrastructure from developers to a testing and production environment. This not only helps to increase the working output, but also helps leaders and members of a team dedicate their time to important projects rather than spend their time fixing routine pesky problems.

1.3 Purpose

Therefore, the aim of the paper is to analyze the phenomenon of end to end automation in the context of cloud infrastructure broadly. It will go further in explaining the different aspects in the process of automating the creation, management and continuously of cloud resources. Furthermore, the paper has the objective of identifying the advantages and possible drawbacks of integrating such automation to show the prospects and specific case studies of adoption. In conclusion, it aims at providing IT workers and managers with all the necessary information required to successfully apply automation in a cloud environment and thus obtain the desired results with cloud computing technologies.

2.0 Background and Context

2.1 Evolution of Cloud Computing

The evolution of cloud computing has taken roots which have recorded certain developments that have been accredited to the management of IT infrastructures. Personally, most of the IT resources were centralized in enterprise data centers where they called for capital investment, with cumbersome costs on hardware, personnel, and even power consumption. Virtualization arrived in the early years of the new millennium to provide the

[1] Independent Researcher, Illinois, United States of America

conditions required by cloud computing by offering virtual machines. This leveled up the usage of the resources and paved the way to the scalability and elasticity appearing in most modern cloud platforms [3]. While adopting cloud computing, CSPs started providing additional services such as IaaS, PaaS, and SaaS for the organizations to host their applications and services [13]. This shift cut down the capital expenditures and also ensured improved efficiency and increased the speed of development of new applications and updates.

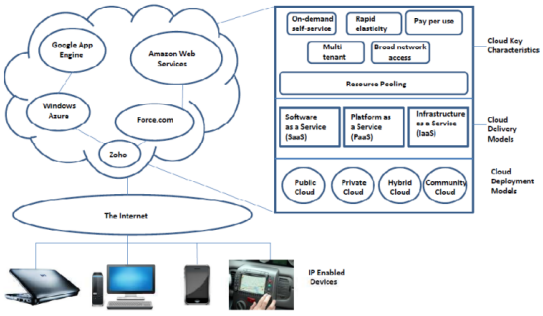

Figure 1: Clouds computing services as SaaS PaaS and IaaS

(Source: Padhy, *et al.* 2011)

2.2 Key Concepts

Provisioning: Automation enables the fast creation of the infrastructure resources like virtual machines, storage volumes, and network resources through code or manifest files.

Configuration Management: Application and system resources are set and controlled by products such as Ansible, Chef, and Puppet and so on to enforce organization SOPs [4].

Orchestration: Container orchestration frameworks like Kubernetes help in managing the distribution, scalability and controlling the apparatus of application and microservices; which are built through containers over several clusters of virtual /physical hosts.

Monitoring and Management: Automated monitoring solutions monitor the cloud resources' performance, availability and security, and generate notification and take necessary actions to ensure high availability and performance.

2.3 Challenges Addressed

Coordinating and synchronizing disparate automation tools and processes within the hybrid or multi-cloud environments may be challenging and may call for an expert's attention. In relation to the subject of automated infrastructure configuration and deployment, one of the most important aspects that need to be addressed are the security and compliance matters related to them. Companies should hire qualified specialists familiar with the automation tools, cloud solutions, and standards to provide proper automation for their infrastructures.

3.0 Review of the technologies and tools

3.1 Automation tools overview

The automation of cloud infrastructure is accompanied by the existence of numerous effective tools mainly tailored at optimizing the process of provisioning, configuration management, as well as overall orchestration. Users can define the resources to be placed in the AWS cloud by using the YAML or JSON templates known as AWS CloudFormation templates. It provides an automated process for creating and managing AWS infrastructure, as well as managing different AWS services like EC2, S3, RDS etc [5].

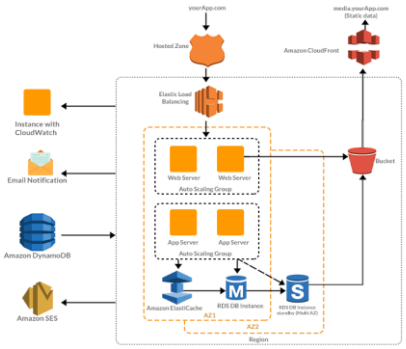

Figure 2: A End-to-End design in AWS architecture

(Source: https://k21academy.com)

It helps in automating the probes and also guarantees that the corresponding settings are similar across the environments. Puppet is one of the leaders in CM, and it has Puppet Enterprise as its commercial product. It employs a DSL to dictate and apply configurations over a variety of infrastructure parts. Puppet works across the public, private, and hybrid cloud domains and is very well scalable and portable with popular cloud services. Also, due to the non-agent nature of the tool, it uses YAML code to configure cloud infrastructure through what is called playbooks. It is also an advanced version of ansible where the management is improved by job scheduling, GUI inventory, and multiple playbooks. It synchronizes well with cloud solutions, for instance, AWS, Azure, or Google Cloud environments. Chef provides Chef Automate for the infrastructure automation process, and by using the DSL, it enforces configurations for continuous delivery creations.

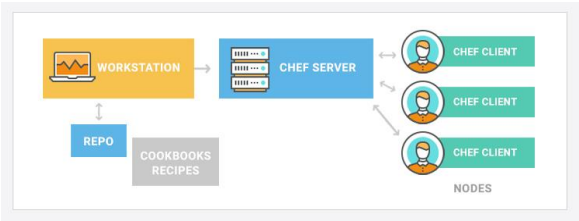

Figure 3: Chef client-server configuration

(Source: https://logz.io)

This one can be installed on local servers or run on cloud, with the options like workload compliance and GUI workflow pipelines [1]. The tool can be easily incorporated with AWS, Azure, VMware and Google Cloud platforms.

Kubernetes stands out for the container Orchestration and Automation for Deploying, Scaling, and Managing Containerized Applications. It hides the infrastructural details especially to do with the optimum use of resources and workload in a given cloud environment. Kubernetes is becoming popular and is supported by all the major cloud companies and supports microservices architectures. Terraform uses an “infrastructure as code” method for managing cloud resources on an increasing scale. It leverages simple, expressive text files for infrastructure declaration and synchronization, across AWS, Azure, Google Cloud, and others. Terraform is a powerful tool that has the capability of displaying resource dependencies as well as the execution plan of an operation for infrastructure updates [18]. This tool enables the creation of infrastructure on Google Cloud by applying configuration templates. It works well with parallel implementations and uses templates for cloud configurations, to increase the level of repetition and standardization.

Table for the tools and cloud implementation

Tool	Cloud Implementation
AWS CloudFormation	AWS
Puppet	Public, private, hybrid clouds
Ansible	AWS, Azure, Google Cloud
Chef	AWS, Azure, VMware, Google Cloud
Kubernetes	AWS, Azure, Google Cloud
Terraform	AWS, Azure, Google Cloud, etc.
Azure Automation	Azure, non-Azure environments
SaltStack	AWS, Azure, VMware, etc.
VCM	VMware-based clouds (vCenter, vCloud Director)
CFEngine	Amazon EC2
Foreman	AWS, Google Cloud, OpenStack, VMware

Azure Automation delivers runbook processing for processes in enterprise hybrid environments, operating system update management, and configuration. Designed for Windows and Linux heterogeneous environments, it must be orchestrated via PowerShell or Python runbooks to boost operability and compliance. SaltStack is used for managing configuration along with compliance on security across various clouds. It is agentless, but the agents are supported for scaled infrastructure management if needed. Regarding specific cloud, SaltStack supports AWS, Azure, VMware, etc. , with focusing on the security and scalability aspects (Johansson, J., 2017). Some advice for VCM are suits for VMware-based cloud environments; VCM allows centralized configuring for models, also performs compliance checks, and can be integrated with VMware products, vCenter, and vCloud Director. CFEngine is another effective mixture of automatic configuration management useful for big-scale IT environments. For configuration enforcement, it relies on self-governing entities and can interface with Amazon EC2 in the deployment on cloud. Foreman: There will be a solid management of the server from bare metal provisioning to the orchestration of the same by Foreman. It is also tightly embedded with Puppet for configuration management and has had plugin support for Chef, Ansible, and SaltStack. As it does provision cloud instances on AWS, Google Cloud, OpenStack, and VMware [6].

3.2 Comparison

This is on the basis of the simplicity that Ansible enjoys when it comes to implementation as compared to the tools like Puppet and Chef that need one to learn their specific DSL and are very powerful in handling configuration management. Ansible and SaltStack are used for the automation of the process of deployment and excision of overhead whereas Puppet and Chef enforce client nodes to have an agent installed on them. Puppet

and Chef are good at large structures with thousands of nodes; therefore, they are ideal for enterprises. Both Ansible and Terraform are more scalable, but on the same note, they take a different approach to scalability because of the structure. All tools work with all main cloud providers, though the level of integration and the available services differ. Kubernetes is most effective with container centric environments while Terraform is more effective with infrastructure provisioning across cloud providers.

3.3 Integration

All of the mentioned tools, CloudFormation, Puppet, Chef, Ansible, and the relatively new Terraform provides first class support for AWS services and thereby provides full coverage of AWS based infrastructure automation. Azure supports automation through Puppet, open-sourced Chef, Ansible, and Terraform; however, it relies on Azure Automation for automating processes. This concise engineering of solutions on GCP is achieved through automation tools such as Ansible, Terraform, and Google Cloud Deployment Manager to increase speed and efficiency of operational activities. Terraform stands out when it comes to the infrastructure in multiple clouds because it supports infrastructure as code of AWS, AZURE, Google Cloud, and other cloud suppliers. Puppet and Chef remain relevant where hybrid cloud is concerned since it denotes that the cloud is well set to support diverse infrastructures in a consistent manner [7]. Deciding what tools to use depends on certain demands regarding infrastructure, certain objectives of the enterprise, and the IT environment of the enterprise. These will assist in drawing out the evaluation criteria that will enable one to select tools that best meets your cloud infrastructure automation needs.

4.0 Review of the architecture of Cloud Computing_300

4.1 Review of the design Principles for implementing end-to-end cloud automation by cloud computing

Modularity and Reusability: It assists in establishing automation processes with components that can be used and rearranged in varying contexts and systems. This approach reduces the complicatedness of managing cloud infrastructures and improves flexibility.

Standardization and Consistency: It assists in normalizing the processes, as well as settings with the help of tools and platforms like IaC and CM platforms (Puppet, Chef, Ansible) [8]. The accurate configuration and the recurrent use of the available equipment reduce errors and raise dependability.

Scalability and Elasticity: It is useful in creating solutions enabling automation of resources in the cloud to adapt to fluctuating workload in the future. Find and use different kinds of orchestration like Kubernetes if working with containers or AWS Auto Scaling for application scaling.

Resilience and High Availability: It aids in designing systems having backup components and creating automatic failure recovery systems. It helps obtain and maintain constant access to critical applications and services through anticipation and self-healing actions.

Security and Compliance by Design: It is useful for the incorporation of security measures at various phases of the automation processes, from deployment to use. It sets up security and compliance into methods and engages services like AWS Config or Azure Security Center to decrease risks and ensure imitate compliances.

4.2 Review of the components for the provisioning, configuration management, monitoring, and compliance

Provisioning: It employs the Infrastructure as Code (IaC) tools like Terraform or AWS CloudFormation to provision cloud resources. Declarative templates are used to define infrastructures in an organization, giving standardized and manageable configuration.

Table for the CloudFormation and Terraform comparison

	CloudFormation	Terraform
PROS	Ul (debugging, overview) CFN Init	Open Source Modules Refresh

	Syntax	(reconciliation) Count
CONS	Verbose Repeated Code Support (not open source) Confused CLI	Debugging Splitting stack is hard Syntax

Configuration Management: Puppet, Chef and Ansible are a few of the examples that are used to automate the software and infrastructure components. They keep commanding certain states, apply configurations, and do policy-based work for cloud systems.

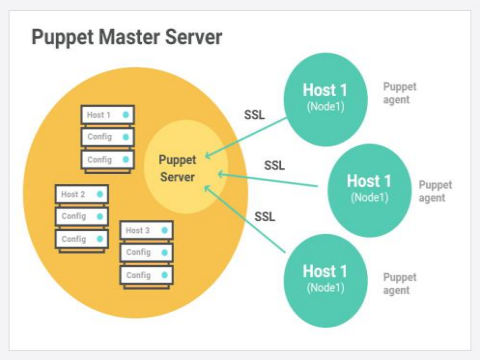

Figure 4: Puppet client-server configuration

(Source: https://logz.io)

Monitoring: This is like Prometheus, Grafana, AWS Cloud watch, and Azure monitor to mention but a few. Measures include collecting metrics automatically and alerting and troubleshooting to guarantee that the cloud environment health and performance are well managed.

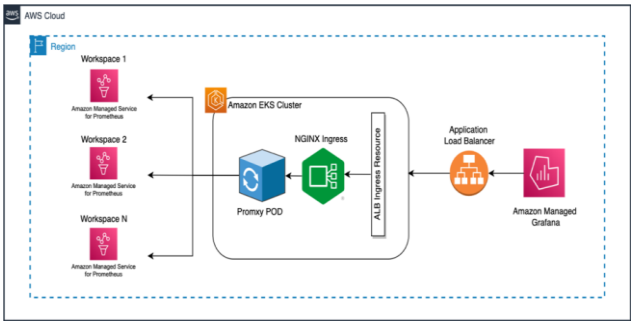

Figure 5: Amazon Managed Grafana

(Source: https://d2908q01vomqb2.cloudfront.net)

Compliance: For compliance checks and governance, the governance processes should be automated using Puppet Enterprise or Chef Automate. This also ensures a secure and compliant Cloud infrastructure by enforcing security policies & auditing misconfigured resources and remediation of the non-compliant ones.

5.0 Review of the implementation strategies

5.1 Workflow Automation

A critical factor in flows in cloud environments is the sound practices related to the workflow automation. To broker applications, companies use orchestration instruments like Kubernetes or Docker Swarm that, or instruments like AWS ECS. These have the added advantage of enabling accurate resource and workload management which is very essential for hybrid and multi-cloud environments. CI/CD tools such as Jenkins or

GitLab CI/CD pipeline means to integrate and automate the development and deployment processes hence guaranteeing the velocity of software delivery. Also, Event-Driven Architecture and Serverless computing like AWS Lambda, Azure Functions allow for responding to system events with actions which in return boosts scalability.

5.2 Best Practices

It is therefore significant to apply best practices for cloud automation in order to get the best results as far as the performance and management are concerned. Terraform and AWS CloudFormation etc. let the organizations follow infrastructure as a code to provision and manage the hardware resources with the help of code like structures [9]. Therefore, this approach helps since it avoids a scenario where there are differences in deployments between environments. Configuration management using Puppet, Chef, or Ansible ensures compliance, automates the process of setting up devices and probes for change, and updates them to policy. Regular checks with powerful tools like Prometheus or Grafana or the providers' native monitoring allow for early detection of incidents and improving system availability.

5.3 Security Considerations

The issue of security still plays a primary role of concern in cloud automation as it deals with new threats as well as protection of data. Using permanent guidelines for Identity and Access Management (IAM) with role-based access controls (RBAC) allows controlling the rights and minimizing the access to critical assets [16].

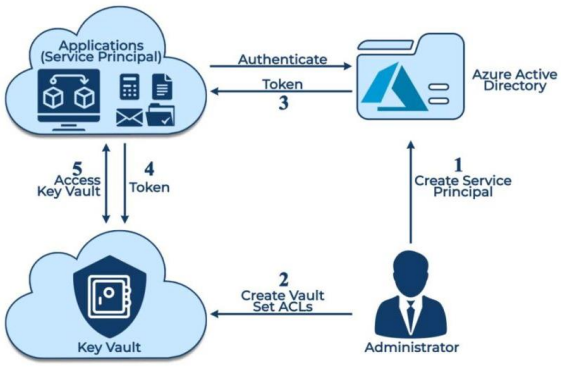

Figure 6: Authentication in Azure Key Vault

(Source: https://www.encryptionconsulting.com/)

TLS/SSL, AWS KMS, or Azure Key Vault ensure the data security both during the transfer and storage, while the automated settings help maintain consistency regardless of the environment [10]. Products such as AWS Config or Azure Security Center execute checks and audits; they are constantly looking for deviations from the configurations and may remediate non-compliant resources on their own.

6.0 Benefits and Challenges

6.1 Benefits of Automation

Automation of cloud infrastructures entails great benefits to organizations that want to achieve optimum operation and effectiveness [11]. The benefits that may be derived from measures such as automated provisioning means that configuration and management will be free from human interference hence setting the stage for deployment of systems to take much shorter time than what would have been the case with lots of manual interferences. It means that it is cheaper as resources are utilized based on how frequently they are needed and that the business has low operational expenses. Also, automation is effective and efficient in maintaining a consistent configuration resulting in the standard compliance of configuration management across the numerous cloud environments. It provides a way of fast growing, thus making it possible for an organization to adapt quickly to changes in business/customer needs while at the same time being highly dependable. Lastly, it relieves organizations of tedious functions allowing key human resources to be best utilized in properly transforming organizational strategies for improvements and growth.

6.2 Operational Challenges

Similar to other types of solutions, the adoption of cloud automation has several performance-related consequences at the operational level. The first one is that the establishment of automation frameworks and the connections between different tools in different clouds produces rather high entry barriers. Businesses need to employ qualified staff for the proactivity of automation of various processes and smooth running of robots. Moreover, automation of various processes need appropriate governance to address specific elements of security, compliance as well as resources consumption. Another is the possibility of amplification of risk in the case of automation, should controls not also be effectively incorporated. Different governments are making rules which are obstructing the growth of cloud computing [12]. Easily other factors that may cause problems to happen include misconfiguration or unauthorized access to data or services, this test proves the needs of constant examination and tuning on the automation practices that have been implemented.

7.0 Future Trends

7.1 Emerging Technologies of the cloud infrastructure automation

Cloud automation and its development are linked with AI and ML, novel technologies in the contemporary world. Big data and advanced AI will fundamentally change the terms of the preventive predictability and remedial action on cloud infrastructure. Smart automation tools will use even more AI algorithms for resource allocation optimization, failure anticipation and decision-making automation. Implementation of Serverless and Function as a Service (FaaS) will persist and increase due to versatility and effectiveness in letting organizations gain better control at present microservices and costs [14]. Furthermore, the emergence of edge computing puts the demand for automation tools to support multiple and geographically distributed structures in different segments of the business, apart from data centers. Cloud providers are helping in the deployment of security and privacy. It also helps in providing orchestration service, and helps in automation of the cloud service.

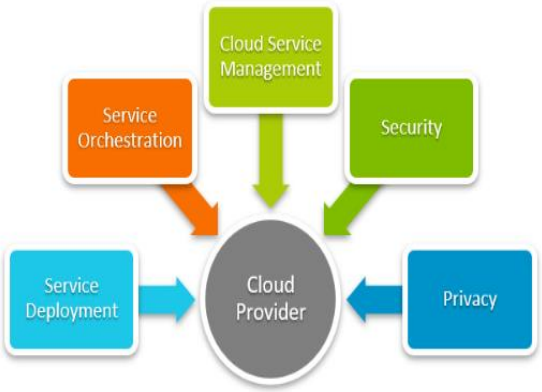

Figure 7: Provisions of the cloud service providers

(Source: https://s7280.pcdn.co/)

7.2 Impact of IT operations and infrastructure management.

Cloud automation indeed dramatically affects IT operations and infrastructure management in terms of quantity and quality. Automation replaces laborious tasks carried out by traditional IT to minimize the workload while at the same time enhancing the timings for new services and applications. Another strategic implication is that while the level of LCM implementation will keep growing, IT teams will shift their attention from low-value activities such as day-to-day operations to those that add value more directly [15]. In addition, automation creates positive change and adaptation to new technologies in ways that are efficient for the IT organization. However, the promotion of automation requires the raising of IT professionals in the understanding and management of automated processes, their efficiency, and compliance with the objectives and legislation of the organization. Thus cloud automation is very important in the growth and administration in an organization.

8.0 Conclusion

Thus it can be concluded that cloud solution automation of end-to-end key IT infrastructure is one of the biggest breakthroughs in IT solutions that seeks to bring more efficiency and scalability along with reduction of costs. Though these problems include but are not limited to complexity and security risk, they are solvable through conducting proper planning, strong governance, and organizational learning. Even in the future, continued integration of AI with such technologies as serverless computing and edge technologies will continue the advancement of cloud automation and transform the paradigms of IT operations and infrastructure management. There is no doubt that those organizations which have adopted cloud automation and become proficient in its use are likely to achieve a competitive advantage in terms of speed and innovation thus' propelling their digital evolution agenda.

Reference List

Journals

[1] Paul, F 2018, "The best tools for cloud infrastructure automation," New Relic, URL: https://newrelic.com/blog/best-practices/best-cloud-infrastructure-automation-tools.

[2] Infopulse 2018. Choose the Best: AWS vs. Azure vs. Google Cloud Comparison. URL: https://www.infopulse.com/blog/choose-the-best-aws-vs-azure-vs-google-cloud-comparison.

[3] Sowmya, S.K., Deepika, P. and Naren, J., 2014. Layers of cloud–IaaS, PaaS and SaaS: a survey. *International Journal of Computer Science and Information Technologies*, *5*(3), pp.4477-4480.

[4] Hagara, M., 2018. Evaluation of Infrastructure as a Code for Enterprise Automation.

[5] Wadia, Y., 2016. *AWS Administration–The Definitive Guide*. Packt Publishing Ltd.

[6] Joshi, N. and Shah, S., 2019. A comprehensive survey of services provided by prevalent cloud computing environments. In *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018, Volume 1* (pp. 413-424). Springer Singapore.

[7] Bigelow, SJ 2018, Grasp the key concepts of hybrid cloud orchestration. URL: https://www.techtarget.com/searchcloudcomputing/tip/Grasp-the-key-concepts-of-hybrid-cloud-orchestration.

[8] NOCERA, D.I.F., DI NOIA, T. and GALLITELLI, D., 2016. Innovative techniques for agile development: DevOps methodology to improve software production and delivery cycle.

[9] Prassanna, J., Pawar, A.R. and Neelanarayanan, V., 2017. A review of existing cloud automation tools. *Asian J Pharm Clin Res*, *10*, pp.471-473.

[10] Itkis, G., Kaiser, B.H., Coll, J.E., Smith, W.W. and Cunningham, R.K., 2016. Charting a Security Landscape in the Clouds: Data Protection and Collaboration in Cloud Storage.

[11] Xue, C.T.S. and Xin, F.T.W., 2016. Benefits and challenges of the adoption of cloud computing in business. *International Journal on Cloud Computing: Services and Architecture*, *6*(6), pp.01-15.

[12] Avram, MG 2014, "Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective," Procedia Technology, 12529–534.

[13] Padhy, R.P., Patra, M.R. and Satapathy, S.C., 2011. X-as-a-Service: Cloud Computing with Google App Engine, Amazon Web Services, Microsoft Azure and Force. com. *Com. Int. J. Comput. Sci. Telecommun*, *2*(9).

[14] Sewak, M. and Singh, S., 2018, April. Winning in the era of serverless computing and function as a service. In 2018 3rd International Conference for Convergence in Technology (I2CT) (pp. 1-5). IEEE.

[15] Ljubicic, M., Ivezic, N., Kulvatunyou, B., Nieman, S., Anicic, N. and Marjanovic, Z., 2017, June. Business process model life-cycle management in cloud manufacturing. In *International Manufacturing Science and Engineering Conference* (Vol. 50749, p. V003T04A039). American Society of Mechanical Engineers.

[16] Kumbhare, A., Simmhan, Y. and Prasanna, V., 2012, June. Cryptonite: A secure and performant data repository on public clouds. In *2012 IEEE Fifth International Conference on Cloud Computing* (pp. 510-517). IEEE.

[17] Johansson, J., 2017. A comparison of resource utilization and deployment time for open-source software deployment tools.

[18] Mara, N., 2018. Generic Deployment Tools for Telecom Apps In Cloud: terraforming.