# DevSecOps: A Security Model for Infrastructure as Code Over the Cloud

Amr Ibrahim
*School of Information Technology and computer science, Nile University, Center of informatics science, Nile University*
Cairo, Egypt
Amr.Mohamed@nu.edu.eg

Ahmed H. Yousef
*School of Information Technology and computer science, Nile University, Center of informatics science, Nile University, Faculty of Engineering, Ain Shams University*
Cairo, Egypt
Ahassan@nu.edu.eg

Walaa Medhat
*School of Information Technology and computer science, Nile University, Center of informatics science, Nile University, Faculty of computer and artificial intelligence, Benha University*
Cairo, Egypt
Wmedhat@nu.edu.eg

2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) | 978-1-6654-6677-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/MIUCC55081.2022.9781709

**Abstract— DevSecOps includes security practice while applying DevOps. DevSecOps help secure the whole DevOps process. This paper aims to define a DevSecOps module to be used by the infrastructure team while applying infrastructure as code. The proposed module solves the problem of security by including security practice with the DevOps cycle to reach DevSecOps. The module was tested to measure time efficiency. A small survey was created to test other DevSecOps metrics and enhance future work.**

***Keywords—DevOps, DevSecOps, Cloud computing, Infrastructure as code***

## I. INTRODUCTION

In recent years, software companies during digital Transformation have focused more on customer satisfaction and accelerating product delivery to the customer. They used a lot of best practice methodologies, including agile software development, ITIL, ... . The most recent trending approach used now by most software companies is DevOps [1]. DevOps is a group of continuous delivery practices that aim to decrease the delivery time, increase the delivery efficiency, and reduce time among releases while maintaining software quality [2]. DevOps helps agility reach continuous customer delivery by increasing implication between the Development Team and the operation team. The main goal of DevOps has been defined as decreasing the time between the development and operation of software without negatively affecting quality [3].

Cloud computing is used within DevOps to reduce costs and resources, which help increase continuous delivery. Instead of buying on-premises servers with its accessories and hiring many resources to manage these infrastructures, companies move to the cloud. So, they buy only for resources they need and employ only a few people to manage the cloud. Infrastructure as code also plays an essential role in speeding the process. Infrastructure as code (IaC) helps set up infrastructure like servers, load balancers, and databases on the cloud using code instead of doing it manually every time. Therefore, if we have any new project, we can set up the whole Infrastructure by running the code instead of doing all the operations from the beginning every time. A widespread tool for Infrastructure as code is Terraform [4].

After DevOps, companies found many challenges related to security while applying DevOps techniques. Therefore, the market needs to introduce new keywords like DevSecOps. DevSecOps is a new paradigm that breaks the Security Team Silo into the DevOps Methodology and adds security practices to the software development life cycle (SDLC) [3]. Research now tries to define many aspects related to DevSecOps [5], but it is still a gap in DevSecOps research.

The proposed module in this paper defines a DevSecOps module for infrastructure as code. The challenge here is how to define a fully secured module and an automated one as well. The main contribution of this paper is collecting and arranging all single security steps found in previous works and creating well defined, fully automated securing module in IaC. The module used terraform and ansible and applied on AWS cloud.

Three research questions were formulated to specify the goal of the paper.

- Question 1: What is the benefit of Cloud for DevOps?
- Question 2: Why use Terraform and Ansible rather than other tools?
- Question 3: How can we implement a module to reach DevSecOps in IaC?

This paper is organized as follows. Section two shows the literature review. Section three shows tools and definitions. Section four tackles the methodology and implemented module. Section five introduces the result and discussion. Section six concludes the paper and presents future work.

## II. LITERATURE REVIEW

The literature highlighted essential papers related to DevSecOps. Some focused-on theories and principles. Others recommend tools and frameworks.

In [6], the author showed that DevSecOp is not a marketing word and showed many examples. The author men-

978-1-6654-6677-6/22/$31.00 ©2022 IEEE

tioned many tools used in the industry while adapting security with DevOps. The paper categorized the tools into scanning tools, security frameworks tools, Results consolidation tools, monitoring tools, and finally logging tools and showed examples like OWSAP ZAP in scanning tools. In [7], the author tackled the number of projects exposing secret information inside configuration files, deployment scripts, or the code itself in GitHub public repositories and mentioned its effect on application security. DevSecOps metrics were introduced in [8]. This study identified the metrics that teams can use to measure the effectiveness of DevSecOps methods implementation inside organizations. In [9], the author mentioned a pilot study for teaching DevSecOps. The author focused more on teaching study rather than DevSecOps techniques. In [10], the author focused on best automation practices in software processes and gave an industrial case study.

In [11], a multivocal literature review was created. The author focused on previous work done for DevSecOps. The paper determined how DevSecOps can be defined and the challenges of applying DevSecOp. The author mentioned the benefits if DevSecOps is applied successfully. Continuously integrating security was introduced in [12]. The author noted the security practice for continuous deployment used by large firms like Facebook and Google. In [13], the author investigated the issues when organizations seek to adhere to security in DevOps in a manner that does not limit the organization's automated pipelines' velocity. In [14], the author mentioned DevSecOps and related it to agency theory.

In [15], the author implemented a new project management tool based on DevSecOps.The author noted how we could implement a project management tool based on DevSecOps practice. In [16], the author mentioned the security impact and lessons learned from misusing DevSecOps. In [17][18][19], the authors focused on Devsecops and security in the cloud with different aspects. In [20], the author mentioned the security risks from Third-party dependencies on the software supply chain and proposed a framework for dynamic analysis of software and its third-party dependencies.

Finally, DevSecOps was introduced in many papers. However, a few ones mentioned DevSecOps in IaC. In [21], the author applied qualitative analysis on 1,726 IaC scripts to identify seven security smells. The proposed study helps practitioners avoid insecure coding practices while developing infrastructure as code. The author implemented a security linter tool to check these seven smells in IaC scripts.

## III. TOOLS AND DEFENITIONS

Chef, Puppet, Ansible, and SaltStack are all configuration management tools. They are designed to install and manage software on existing servers. CloudFormation and terraform are provisioning tools. They are designed to provision the servers. Table one shows why we suggest using ansible rather than other tools.

*Table 1. configuration management tools comparison*

| factor | Tools | | | |
|---|---|---|---|---|
| | **Ansible** | **Chef** | **puppet** | **Salt Stack** |
| language | YAML | Ruby | DSL & ERP | YAML |
| installation | Agent | Agent | Agent | Agentless |
| Approach | Declarative and imperative | Declarative | Declarative | Declarative and imperative |

Table 1 includes the most popular configuration management tools. The first factor we compare is the language. YAML is the best language to write configuration. YAML is a human-readable data-serialization language. They are commonly used for configuration files and in applications where data is stored or transmitted. The second factor is the installation type. Agent means that the tool requires installing an agent on the remote server to apply the configuration. Ansible is agentless and does not need to install an agent on a remote server. Only open an SSH connection between the master and remote server. The third factor is that the approach for the language is declarative or imperative. Ansible supports both types, using YAML and agentless.

*Table 2. provisioners tools comparison*

| factor | Tools | | |
|---|---|---|---|
| | ***Terraform*** | ***CloudFormation*** | ***Azure Resource manager*** |
| providers | All Providers | AWS | AZURE |
| Language | DSL | YAML | YAML |
| Appraoch | Declarative and imperative | imperative | imperative |

Table 2 compare commonly used infrastructure as code tools. The first factor is the cloud providers supported by the language. Terraform can set up the environment on all public cloud providers like AWS, AZURE, and GCP. CloudFormation support only AWS. AZURE resource manager support only AZURE provider. Terraform does not support YAML language, but it is simple in writing and supports both declarative and imperative. Terraform is the best language used to establish the environment over the cloud.

AWS has a service named S3 used for object storage. This service stores files, images, static websites, etc. Also, aws has a service named aws system manager. AWS system manager contains a service called parameter store. Parameter store used to store secrets like .pem files. Terraform contain a file name terraform state file. This file stores metadata about the installed infrastructure created by terraform code. If anyone has access to this state file, he can change the environment and know every detail of the domain. Ansible is a configuration management tool used to automate server operation tasks like installing Apache web servers on the servers [22]. Ansible contains a file named inventory. Inventory files contain IP addresses for all machines managed by ansible. If anyone shares this file, he

can know the IP addresses of servers that host the applications. Tfsec is a static analysis security scanner for Terraform code. Tfsec uses static analysis of terraform templates to spot potential security issues. Tfsec will be used in the defined module to reach security for IaC.

Knowing the old process of setup infrastructure as code help understanding the module. DevOps Team implements and runs Terraform code first. Terraform state file kept inside the server. After the infrastructure is installed, the Ip address of the target servers puts manually inside the Ansible inventory file. The generated ssh pem. Files kept locally to use by ansible. Ansible used a SSH key file to access the servers to install the required packages. Finally, ansible runs to configure the servers. The issue in this process is that the team keeps terraform state file, Ansible Inventory file, SSH pem. File locally. Moreover, many operations are done manually.

## IV. METHODOLOGY

Achieving DevSecOps, we need to automate the complete process and include security practices in infrastructure as code [26]. This module aims to define a clearly defined module while applying IaC. This module was implemented using Terraform, Ansible, and bash scripting and tested on the AWS cloud.

Figure 2 shows the steps we should do to apply the module.

- DevOps Engineer runs the terraform code to create or modify the cloud environment.
- Tfsec check security spots in the terraform code.
- Updated terraform state file uploaded to s3 bucket.
- SSH keys downloaded from parameter store to use by ansible.
- Terraform generate the IP address to use by ansible inventory.
- Ansible inventory dynamically created with terraform public IP's
- Ansible configures the servers.
- Finally, all secrets are deleted automatically from the server.

The module solves the problem of automating the whole process and securing all the secrets. Deep dive in the steps. Instead of keeping it local, the state file is uploaded to an encrypted s3 bucket. SSH keys are stored inside the AWS parameter store. After running the code, SSH keys are downloaded automatically inside the server. Then, terraform generate an output file containing an IP Address for all running machines. Ansible inventory is created automatically from the output file instead of created manually. After ansible configures the servers, all the secrets and inventory files are deleted automatically. The new in the module is the integration between terraform and ansible. Previous work does not provide fully secured steps from step one to the final. This module provides a secure step for the whole process.

Another alternative, instead of uploading the state file in the s3 bucket, we can use terraform cloud workspace. Terraform cloud manages infrastructure collections with workspaces instead of directories. A workspace contains everything Terraform needs to manage a given infrastructure group, and separate workspaces function like working directories. Workspaces have an option to upload the state file on the account remotely instead of keeping it local.

## IV. RESULTS AND DISCUSSIONS

Table 3 includes answers for a small module evaluation survey questions. The evaluators from different software firms. The survey consists of the following questions.

- Question one: does the module secure the infrastructure as code?
- Question two: does the module automate the whole process?
- Question three: What are the other comments?

All the answers agree with questions one and two. In other comments, Engineer one mentioned that A deeper analysis is needed for any generated files to measure the security. The evaluator did not recommend any analysis tool after asking for a recommendation. This point can be included in future work. The module contains the tfsec tool, which creates deeper analysis for the terraform code, not the whole module. Tfsec was included in the module after a recommendation from engineer two. The remaining engineers satisfy with the module without any new recommendations

The module is applied in a small environment over the AWS cloud. The purpose of this experiment is to measure time efficiency for the module. Figure 1 shows the architecture diagram of the setup environment. The environment includes two virtual machines with a load balancer to balance. Load balancer put in the public subnet and virtual machines on a private subnet.

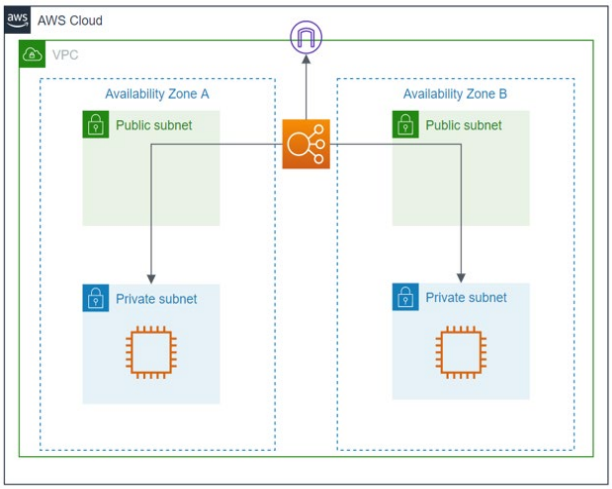


*Figure 1 AWS Architecture*

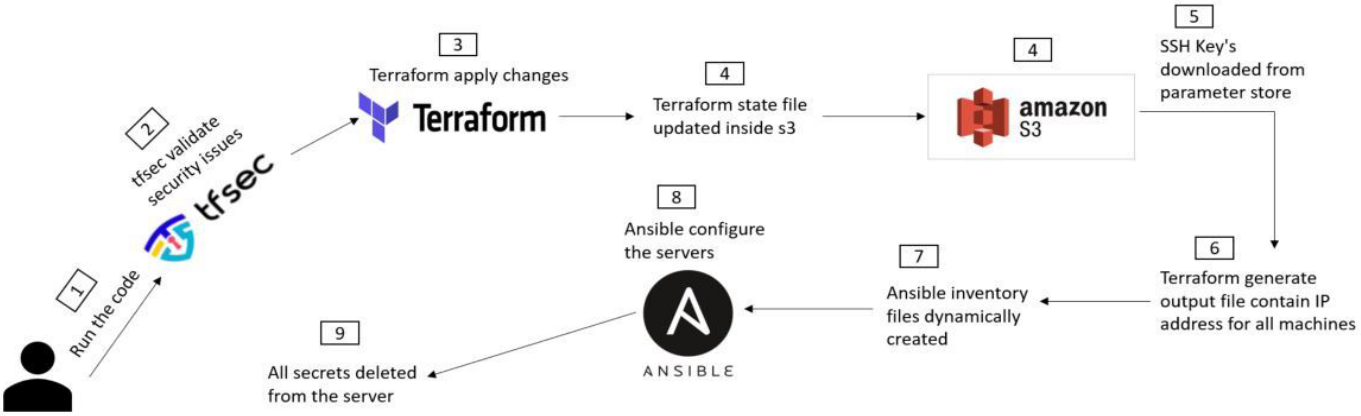


*Figure 2 DevSecOp module*

*Table 3. questions survey for the module*

| Name | Company | Country | position | Question 1 | Question 2 | Question 3 |
|---|---|---|---|---|---|---|
| Engineer one | amazon /imdb | England | Software engineer | yes | yes | A deeper analysis is needed for any generated files to measure the security |
| Engineer two | heycar | Germany | DevOps Engineer | yes | yes | You can also use validation tools in the process like tfsec to measure security in the module |
| Engineer three | Cegedim | Egypt | Tech. lead | yes | yes | It will save time and cost, every time in the initial configuration you get peace of mind |
| Engineer four | Vodafone | Egypt | cloud architect | yes | yes | It's a very good method to create infrastructure as code |
| Engineer five | nfsc | KSA | DevOps Engineer | yes | yes | No comments |

Table 4 shows the difference in time in setup the environment. The total number of servers needed to establish projects exceeds hundreds and thousands for large enterprise projects. Using the manual method will take days and months to set up and destroy the whole infrastructure. However, we can use hours only to set up everything with the automated module. Also, automation will replace much effort for cost, and many hired employees need to set up everything manually.

*Table 4. Time needs to setup environment*

| Method | Time |
|---|---|
| Manual | 30 minutes |
| Automated using module | 5 minutes |

## V. CONCLUTION AND FUTURE WORK

This paper solves the issue of securing infrastructure as code while setting up the infrastructure for software projects. The paper recommends the DevSecOps module for infrastructure as code. The module automates and secures the whole process. The module was evaluated and tested in two ways. First, A test environment was created, and the module was applied to measure time efficiency. Secondly, members from different software firms tested and evaluated the module. The results show that the module is six times faster than setting up the environment manually. In future work, we will focus more on profound analysis tools for generated secrets as per the engineer's suggestion in table 3. Another important point we should focus on is applying DevSecops on all DevOps tools and cycles like CI/CD and monitoring and logging to fit the gap in the research area in DevSecOps.

## REFERENCES


[1] Oyce, W. I. B. J. (2005). E Fficiency in P Ractice and P Ricing of a Ccounting S Ervices. Journal of Accounting and Finance Research, 13(4), 1–9..

[2] Hakli, A., Taibi, D., & Systa, K. (2019). Towards cloud native continuous delivery: An industrial experience report. Proceedings - 11th IEEE/ACM International Conference on Utility and Cloud Computing Companion, UCC Companion 2018, 314–320. https://doi.org/10.1109/UCC-Companion.2018.00074

[3] Erich, F. (2019). Devops is simply interaction between development and operations. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 11350 LNCS. Springer International Publishing. https://doi.org/10.1007/978-3-030-06019-0_7

[4] J. Prassanna, A. R. Pawar, and V. Neelanarayanan, "A review of existing cloud automation tools," Asian J. Pharm. Clin. Res., vol. 10, no. July, pp. 471–473, 2017, doi: 10.22159/ajpcr.2017.v10s1.20519.

[5] J. S. Lee, "The DevSecOps and Agency Theory," Proc. - 29th IEEE Int. Symp. Softw. Reliab. Eng. Work. ISSREW 2018, pp. 243–244, 2018, doi: 10.1109/ISSREW.2018.00013.

[6] Mohan, V., & Ben Othmane, L. (2016). SecDevOps: Is it a marketing buzzword? Mapping research on security in DevOps. Proceedings - 2016 11th International Conference on Availability, Reliability and Security,ARES2016,542–547. https://doi.org/10.1109/ARES.2016.92

[7] Yasar, Hasan. "Experiment: Sizing exposed credentials in github public repositories for ci/cd." 2018 IEEE Cybersecurity Development (SecDev). IEEE, 2018.

[8] Prates, Luís, et al. "Devsecops metrics." EuroSymposium on Systems Analysis and Design. Springer, Cham, 2019.

[9] Okolica, James S., Alan C. Lin, and Gilbert L. Peterson. "Gaming DevSecOps-A Serious Game Pilot Study." National Cyber Summit. Springer, Cham, 2020.

[10] Mohan, Vaishnavi, Lotfi ben Othmane, and Andre Kres. "BP: security concerns and best practices for automation of software deployment processes: An industrial case study." 2018 IEEE Cybersecurity Development (SecDev). IEEE, 2018.

[11] Myrbakken, Håvard, and Ricardo Colomo-Palacios. "DevSecOps: a multivocal literature review." International Conference on Software Process Improvement and Capability Determination. Springer, Cham, 2017.

[12] Williams, Laurie. "Continuously integrating security." Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment. 2018.

[13] Abrahams, Muhammad Zaid, and Josef J. Langerman. "Compliance at Velocity within a DevOps Environment." 2018 Thirteenth International Conference on Digital Information Management (ICDIM). IEEE, 2018.

[14] Lee, Jong Seok. "The devsecops and agency theory." 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW). IEEE, 2018.

[15] Zunnurhain, Kazi, and Saniora R. Duclervil. "A New Project Management Tool Based on DevSecOps." 2019 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2019.

[16] Morales, Jose Andre, et al. "Security impacts of sub-optimal DevSecOps implementations in a highly regulated environment." Proceedings of the 15th International Conference on Availability, Reliability and Security. 2020.

[17] Elsayed, Marwa A., and Mohammad Zulkernine. "Integrating Security in Cloud Application Development Cycle." 2018 International Conference on Software Security and Assurance (ICSSA). IEEE, 2018.

[18] Carturan, Sara, and Denise Goya. "Major Challenges of Systems-of-Systems with Cloud and DevOps–a financial experience report." 2019 IEEE/ACM 7th International Workshop on Software Engineering for Systems-of-Systems (SESoS) and 13th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (WDES). IEEE, 2019.

[19] Carturan, Sara BO Gennari, and Denise Hideko Goya. "A systems-of-systems security framework for requirements definition in cloud environment." Proceedings of the 13th European Conference on Software Architecture-Volume 2. 2019.

[20] Ohm, Marc, Arnold Sykosch, and Michael Meier. "Towards detection of software supply chain attacks by forensic artifacts." Proceedings of the 15th international conference on availability, reliability and security. 2020.

[21] Rahman, A., Parnin, C., & Williams, L. (2019, May). The seven sins: Security smells in infrastructure as code scripts. In 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE) (pp. 164-175). IEEE.

[22] P. Masek, M. Stusek, J. Krejci, K. Zeman, J. Pokorny, and M. Kudlacek, "Unleashing full potential of ansible framework: University labs administration," Conf. Open Innov. Assoc. Fruct, vol. 2018–May, no. May, pp. 144–150, 2018, doi: 10.23919/FRUCT.2018.8468270