What Do Infrastructure-as-Code Practitioners Discuss: An Empirical Study on Stack Overflow

Mahi Begoug*, Narjes Bessghaier*, Ali Ouni*, Eman Abdullah AlOmar†, Mohamed Wiem Mkaouer‡
*ETS Montreal, University of Quebec, Montreal, QC, Canada
†Stevens Institute of Technology, Hoboken, New Jersey, USA
‡Rochester Institute of Technology, Rochester, NY, USA
{mahi.begoug.1,narjes.bessghaier.1}@ens.etsmtl.ca, ali.ouni@etsmtl.ca, ealomar@stevens.edu, mwmvse@rit.edu

***Abstract*—Background. Infrastructure-as-Code (IaC) is an emerging practice to manage cloud infrastructure resources for software systems. Modern software development has evolved to embrace IaC as a best practice for consistently provisioning and managing infrastructure using various tools such as Terraform and Ansible. However, recent studies highlighted that developers still encounter various challenges with IaC tools.**

Aims. We aim in this paper to understand the different challenges that developers encounter with IaC and analyze the trend of seeking assistance on Q&A platforms in the context of IaC. To this end, we conduct a large-scale empirical study investigating developers' discussions in Stack Overflow.

Method. We first collect IaC-relevant tags on Stack Overflow, constituting a dataset that comprises 52,692 questions and 64,078 answers. Then, we group questions into specific topics using the Latent Dirichlet Allocation (LDA) method, which we optimize using a Genetic Algorithm (GA) for parameter's fine-tuning. Finally, to gain better insights, we analyze the identified topics based on different criteria such as *popularity* and *difficulty*.

Results. Our findings reveal an average yearly increase of 150% in terms of IaC-related questions and 135% in terms of users between 2011 and 2022. Furthermore, we observe that IaC questions revolve around seven main topics: *server configuration*, *policy configuration*, *networking*, *deployment pipelines*, *variable management*, *templating*, and *file management*. Notably, we found that *server configuration* and *file management* are the most popular topics, *i.e.,* the most discussed among IaC developers, while the *deployment pipelines* and *templating* topics are the most difficult.

Conclusions. Our results shed light on IaC challenges that are often encountered by developers on popular Q&A platforms. These findings reveal important implications for practitioners seeking better support for IaC tools in real-world settings and for researchers to better understand the IaC community needs and further investigate IaC in different aspects.

***Index Terms*—Infrastructure-as-Code (IaC), Stack Overflow, Topic Modeling, Developers Discussions**

I. Introduction

Infrastructure-as-Code (IaC) is a practice that continuously manages the infrastructure of software systems by provisioning the infrastructure resources [1], such as computing resources (*e.g.,* virtual machines, containers, servers, clusters), storage resources (e.g., databases, filesystems), and network resources (e.g., user profiles, IP addresses). Various IaC tools, including Ansible [2], Terraform [3], CloudFormation [4], and ARM Template [5], allow practitioners to manage infrastructure resources through code, like any software development code. Thus, IaC practitioners can involve coding practices, such as automated testing and version control, to push infrastructure changes and accelerate the project's delivery process [6].

Recently, several companies adopted the IaC technology into their development pipeline, such as Shopify [7] and Uber [8]. However, as IaC brings several advantages, it could also be challenging to deal with its code complexity related to managing different and dependent resources [9]. Hence, IaC code could be susceptible to misconfigurations and faults, which could have severe consequences. That is, modifying complex IaC files can be error-prone, time-consuming, and difficult [1], [10]. For instance, some IaC changes, known as *Configuration Drift* [1], can cause complex errors and lead to inconsistencies between the actual and desired configuration when executing IaC files. Furthermore, during an interview with 44 senior developers, Guerriero et al. [11] discovered that handling the diverse formats of Infrastructure-as-Code (IaC) tools can be challenging and often requires specialized users. Additionally, the researchers highlighted the importance of having an ontology that defines the categories and relationships of IaC tools or formats to help understand how they can fit together in a given context.

To mitigate these issues in IaC, several studies have explored various aspects [12], such as the co-evolution between IaC and other code files (*e.g.,* test, production, etc.) to analyze the growth in size and complexity of IaC files [13], the quality metrics to evaluate the IaC files [14], defects prediction in IaC files to automatically detect defects at the implementation level [9], as well as the detection of bad implementation and design practices, *i.e.,* smells in IaC artifacts [15]–[18]. Other studies have opted for surveys and/or interviews with developers to investigate IaC-specific challenges related to bugs and anti-patterns [11], [19]–[21]. While interviewing practitioners offers valuable insights in a given context, there is a need for further understanding of the challenges that IaC practitioners face in a broader context. Thus, we focus our study on real-world challenges that developers currently encounter with IaC.

To overcome IaC challenges in practice, developers typically seek assistance for their problems in Q&A web platforms [22]. One of the most popular platforms is Stack Overflow (SO), which is widely used for discussing software engineering technical issues and exchanging solutions through posts [23]. Even though IaC tools provide diverse options for provisioning

978-1-6654-5223-6/23/$31.00 ©2023 IEEE

capabilities, developers may encounter challenges that vary, to some extent, from those in traditional coding. For example, in the SO post shown in Quote 1, developers need help to deploy software systems across multiple infrastructures simultaneously (*i.e.,* hosts). This post has received the correct answer after 6 years, while being viewed over 144k times, and scored 59 times, which may indicate that IaC developers struggle to provide the correct solution to the issue [24].

Ansible: deploy on multiple hosts at the same time
💬 "*Is it possible to run ansible playbook, which looks like this (it is an example from this site):..in multithread mode?.. I want to run three "includes" at the same time (it is deploying to different hosts anyway), like in this diagram:...*"

Quote 1: A question related to simultaneously deploying across different hosts [24]

In this paper, our goal is to investigate IaC-related posts and analyze the different challenges encountered by IaC developers in practice. We perform an exploratory study on 116,770 posts related to IaC on SO. We chose SO as several studies have used it to investigate various software development practices, such as software security [25], refactoring [26], continuous integration [27], and blockchain [28]. In SO, posts can be questions or answers created by users. Each question can have a list of answers, but only one answer is accepted as the correct one to the question. A question also belongs to a set of tags. A tag categorizes the question in development areas. Users can also vote on the quality of posts. Each post has a list of attributes (*i.e.,* metadata) such as id, title, body, creation date, and PostTypeId [29].

We investigate the collected IaC posts to study their trends and analyze their growth. We aim to identify the different topics in which developers ask questions and study their *popularity* and *difficulty*. Our study offers IaC researchers, developers, and educators an opportunity to learn about the different challenges related to IaC development practices. The following research questions serve as our study's guide:

(Trends) ***RQ1: What is the evolution of IaC-related posts?*** In this RQ, we examine the volume of IaC-related questions (52,692), answers (64,078), and unique users (42,702) participating in both questions and answers. As a result, we found that questions on IaC have increasingly evolved from 2011 to 2022. Further, we observe a significant increase in the number of posts between 2018 and 2019, which seems to be the result of various IaC tool-related events. This finding shows that new changes or upgrades to IaC tools could bring several new challenges, particularly when they are poorly documented.

(Topics) ***RQ2: What are the main IaC-related topics asked by developers?*** We applied the LDA technique tuned with GA to classify IaC challenges into different topics. Our findings reveal that IaC practitioners tend to ask about 7 main topics, namely *Server Configuration*, *Policy Configuration*, *Networking*, *Deployment Pipelines*, *Variable Management*, *Templating*, and *File Management*. Specifically, *Server Configuration* is the most asked topic having 23% of the total number of IaC questions.

(Challenges) ***RQ3: Which IaC-related topics are most popular and difficult?*** In this RQ, we study the popularity of the identified topics in terms of views and votes, as well as their difficulty in terms of the time taken to receive a correct answer, and the percentage of unanswered questions. Our analysis shows that the most popular topics are *Sever Configuration* and *File Management* which contain recurrent challenges encountered by developers. On the other side, the most difficult topics are *Deployment Pipelines* and *Templating*, *i.e.,* questions pertaining to these two topics may either remain unanswered or require a significant amount of time to be answered correctly.

Replication Package. We provide a comprehensive replication package of our dataset and scripts available online for future replication and extension purposes [30].

Paper Organization. The rest of this paper is organized as follows. In Section II, we describe our empirical approach. In Section III, we discuss the results, while in Section IV, we highlight the implications of our study. In Section V, we discuss the potential threats to validity, and lastly, we provide the conclusion and future directions in Section VII.

II. Empirical Study Methodology

Our study aims to provide a better understanding of IaC problems that are frequently encountered by developers in practice. To answer our three research questions defined in the previous section, we extract IaC posts from SO and study their trends over the years. Specifically, we intend to identify the different topics where developers seek assistance and determine which topics are most popular and difficult.

Figure 1 illustrates the workflow of our study methodology which includes five main steps: (1) collect data, (2) extract IaC tags, (3) extract and clean IaC posts, (4) apply topic modeling (LDA). In the following, we describe each step.

A. Collect Stack Overflow Data

This step consists of downloading the available snapshot of the SO database (updated on December 2022) from the Stack Exchange Data Dump Archive [31] including SO data in XML files. In our study, we are interested in the posts' XML file [32] which contains developers' questions and answers along with their corresponding metadata. As this file exceeds 90 GB in size, we use the SoTorrent [33] workflow to migrate the "Posts.xml" file to Bigquery Table [34] in the Google Cloud Platform (GCP), which helps us manage this large dataset for our analysis. The SO dataset, denoted as S, contains posts from September 2008 to December 2022. We exclude the year 2008 in which Stack Overflow was launched as incomplete data in a year can lead to misleading analysis [26].

B. Extract IaC Tags

In this step, we extract all posts that are IaC-related. To do so, we retrieved every question which was tagged as *"infrastructure-as-code"* in our dataset S. Our analysis covers the period from 2011 to 2022 since the first post that is associated with the *"infrastructure-as-code"* tag was in 2011. We ended

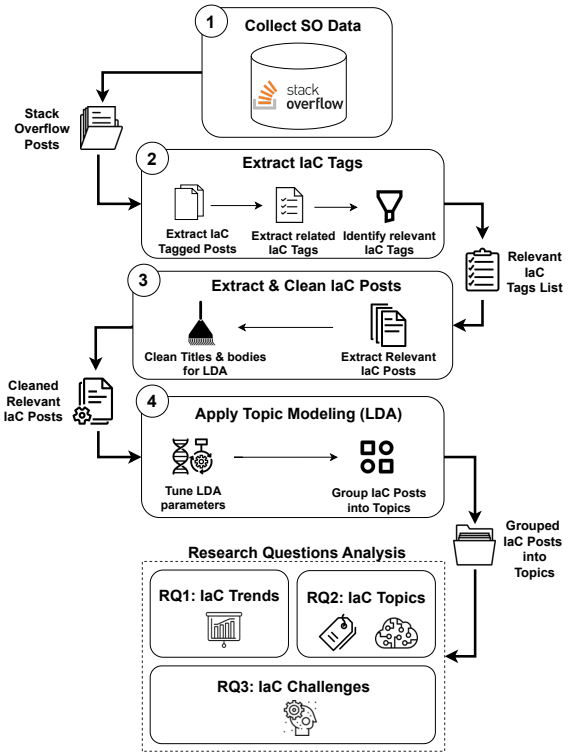

Fig. 1: Design of our empirical study methodology

up with a total of 354 IaC-related questions denoted as Q_{iac}. However, we believe that restricting our posts only on the tag *"infrastructure-as-code"* may not be inclusive, as some users may not use this specific tag when asking IaC questions. To address this issue, we employed techniques used in previous works to develop a set of tags that are IaC-related [35]–[37]. First, we retrieved the list of associated tags with *"infrastructure-as-code"* in Q_{iac}. Second, we used two filter metrics to determine significantly relevant tags that represent IaC, namely *relevance* and *significance* metrics. The relevance metric, $Relevance(t)$, quantifies the proportion of posts in Q_{iac} with a tag t in the posts we have in S. The significance metric, $Significance(t)$, is calculated as the frequency of tag t in Q_{iac}. These metrics are defined as follows:

$$Relevance(t) = \frac{\text{Number of questions tagged with t in } Q_{iac}}{\text{Number of questions tagged with t in } S} \quad (1)$$

$$Significance(t) = \frac{\text{Number of questions tagged with t in } Q_{iac}}{\text{size of } Q_{iac}} \quad (2)$$

To determine whether a tag is relevant or not, we use the corresponding $Relevance(t)$ and $Significance(t)$ values that should be higher than a threshold (σ, ω). We experimented with different combinations of $\sigma_i \in \{0.0005, 0.001, 0.0015, 0.002, 0.0025, 0.003\}$ and $\omega_j \in \{0.01, 0.02, 0.03, 0.04, 0.05, 0.06\}$. As shown in Table I, we found a set including 27 tags with a threshold ranging between 0.0005 and 0.01. The highest threshold value ranging between 0.003 and 0.06 pertain to a set containing seven tags.

Therefore, we decided to study the set with the largest number of tags to avoid losing data and restricting the analysis scope. We further examine the selected set of 27 tags to choose only relevant ones in the IaC context. In particular, we follow the following steps:

- Step 1: *Voting on the tags*. The first three authors carried out an independent review process to exclude tags that are not specific to IaC. They cast a vote of either *"Yes"* or *"No"* on tags. For instance, an author could cast a *"No"* vote on the *"google-kubernetes-engines"* tag, considering its relevance to Kubernetes [38] which is a container orchestrator tool, instead of IaC.
- Step 2: *Measuring the initial agreement*. After the voting step, the authors meet to discuss the evaluation of the tags. We employed the Krippendorff's α [39] coefficient to assess the inter-rater reliability among the authors. Our analysis yielded an initial agreement rate of 0.76, indicating a substantial agreement [40] among the authors with regard to the assigned tags. The authors agreed on keeping 14 tags, removing 4 tags, and disagreeing on 9 tags (*e.g., "terragrunt"*, *"deveops"*, *"azure-devops"*).
- Step 3: *Resolving disagreements*. In this step, the authors discussed to resolve the disagreements concerning the 9 disagreed upon tags found in step 2. In the case of a tag t not receiving three "Yes" or "No" votes from all three authors, they further analyze the potential contexts related to the tag by randomly selecting 20 Stack Overflow posts associated with the tag in question. If two authors come to an agreement regarding the removal or selection of the tag, the disagreement is considered as resolved. This process resulted in the removal of 8 tags judged irrelevant to the IaC context, such as *"amazon-eks"*, *"amazon-ecs"*, and *"infrastructure"*. We finally reached an agreement of 0.94, indicating near-perfect agreement [40]. As a result, we obtained the following T tag list including 18 tags: *infrastructure-as-code*, *ansible*, *pulumi*, *terraform-provider-gcp*, *terraform0.12+*, *google-deployment-manager*, *terraform-provider-aws*, *amazon-cloudformation*, *arm-template*, *aws-cdk*, *terraform-provider-azure*, *azure-resource-manager*, *terraform*, *azure-bicep*, *aws-cloudformation-custom-resource*, *terragrunt*, *hcl*, *hashicorp*.

C. Extract & Clean IaC Posts

From the T tag list, we form our IaC dataset by querying the S dataset, where the posts are tagged with tags that belong to T. As a result, we obtained 116,770 posts, of which 52,692 (0.45%) are questions and 64,078 (0.55%) are answers.

To cluster IaC posts using topic modeling, it is necessary to exclude irrelevant information that may be considered as noise. To achieve this, only questions are pre-processed, which has been shown to reduce bias in prior studies [26], [41]. The titles and bodies of the selected posts are treated as a corpus for topic modeling, which involves multiple steps such as removing code snippet tags and their contents (*i.e.,* ⟨code⟩?⟨/code⟩), HTML tags, and hyperlinks. In addition, word contractions are expanded, and numbers, non-alphabetical characters, punctuation marks, and English stop words are removed. Stop words are identified using a list of English stop

TABLE I. Sample of Tag sets from 36 experimentation' of different threshold values.

(σ_i, ω_j)	Tag set	No.
0.0005,0.01	arm-template ansible devops amazon-ecs azurebicep pulumi terraform-provider-gcp terraform0.12+ azure-devops google-deployment-manager terraform-provider-aws google-kubernetes-engine amazon-cloudformation azure infrastructure cloud amazon-web-services aws-cdk aws-cloudformation-custom-resource terraform-provider-azure terragrunt infrastructure-as-code hcl azure-resource-manager hashicorp terraform amazon-eks	27
0.0005,0.06	devops pulumi terraform-provider-aws amazon-cloudformation azure amazon-web-services aws-cdk infrastructure-as-code terraform	9
0.002,0.03	arm-template dDevOpsazure-bicep pulumi terraform-provider-gcp terraform-provider-aws amazon-cloudformation infrastructure aws-cdk terraform-provider-azure infrastructure-as-code azure-resource-manager terraform	13
0.003,0.06	devops pulumi terraform-provider-aws amazon-cloudformation aws-cdk infrastructure-as-code terraform	7

words from Mallet's list [42] and the default set from NTLK [43]. Furthermore, the Porter stemmer algorithm [44] is applied to transform the words in the corpus to their root form.

D. Apply Topic Modeling

To cluster the pre-processed posts into topics, we leverage the Latent Dirichlet Allocation (LDA) method. Our use of LDA for the topic modeling analysis follows prior research based on Stack Overflow posts (*e.g.,* [26], [45], [46]). LDA creates a probabilistic model that assigns K topics to each post, following I training iterations [47]. The dominant topic that best describes the corresponding post is determined by the highest distribution. In our study, we use the Mallet [48] implementation of LDA. In addition, LDA has two control parameters: α, which controls the topic distribution for each post, and β, which supervises the distribution of keywords within a topic. It has been demonstrated by Bigger et al. [49] that these parameters have a significant impact on the results of LDA. We have set the values of α and β to the recommended values of $K/50$ and 0.01, respectively.

Moreover, the number of topics K and iterations I can both impact the quality of the LDA model. Selecting inappropriate values may result in a long list of noisy topics or a few broad topics. It is thus crucial to identify the optimal configuration for these parameters [36]. LDA parameters tuning presents an optimization problem since we need to find the best set of parameters from a large configuration space that are suited for genetic algorithms (GA) [50], [51]. Hence, we used GA [52] for our LDA parameters tuning.

1) Adapt of Genetic Algorithm (GA): To tune LDA parameters using GA, we define the different components of GA:

- **Initialization:** We define the candidate solution as a combination of two attributes: K topics and I training iterations. To generate the initial population, we randomly assign two values within specific ranges: K varies from 2 to 50, and I ranges between 500 and 3,000.
- **Fitness function:** To assess the goodness of a candidate solution, we define a fitness function (*i.e.,* objective). We seek to maximize the dissimilarity between the topics resulting from the LDA model. To address this, we use the *Topic Coherence* metric, which has shown a correlation with human interpretability and understandability [53].
- **Variation operators:** To produce new candidate solutions, known as offsprings, from our initial randomly generated population of size 100, we initially apply the standard single-point crossover operator with a probability of 0.7 by selecting two parents. Each new solution then takes an attribute from its parents. We also perform a mutation with a probability of 0.1 by replacing the value of an attribute with another value.
- **Stopping criteria:** GA process halts when it reaches the maximum number of generations, which we fixed to 300.

The best model obtained with GA consisted of 7 topics (K) and 3,000 learning iterations (I).

III. Research Questions Analysis

A. (Trends) RQ1: What is the evolution of IaC-related posts?

Motivation: We aim to study the growth of the number of IaC posts from the year of the first submitted IaC-related post. This initial analysis helps to understand to which extent users encounter challenges related to IaC.

Approach: To address this research question, we count the number of IaC-related questions and all the answers to every question, including both accepted and non-accepted answers. Furthermore, we count the yearly unique users who either post questions or answers.

Finding 1: From 2011 to 2022, we observe a continuous increase in the number of IaC questions, including both questions with and without accepted answers, and the number of users in SO. We show in Figure 2 the yearly trend of 1) IaC questions with and without accepted answers, 2) IaC questions with accepted answers, and 3) IaC questions without accepted answers. We see that the initial questions related to IaC were submitted on SO in 2011, consisting of a total of five questions. We also observe that the total number of IaC questions follows a consistently increasing pace over the years with an average increase percentage of 150% questions every year, except for 2018 and 2022, when the number of questions with accepted answers decreased by 105 and 455, respectively, compared to the previous years (2017 and 2021). The increase in IaC-related questions on Stack Overflow suggests that more users are looking for guidance and best practices to implement IaC effectively. This trend also indicates that IaC has become an essential part of modern software engineering, and there is a growing need for knowledge and expertise in automated infrastructure code in parallel with hardware advances. Additionally, Figure 3 shows the count of distinct users involved in IaC discussions per year. We observe a continuous rise in the number of distinct users participating in IaC questions and answers, with a yearly average increase percentage of 135%.

Finding 2: We also observe that the number of questions without accepted answers surpassed the number of questions with accepted answers in the last five years (*i.e.,* from 2018 to 2022). The increase in the number of questions without accepted answers in the last five years of

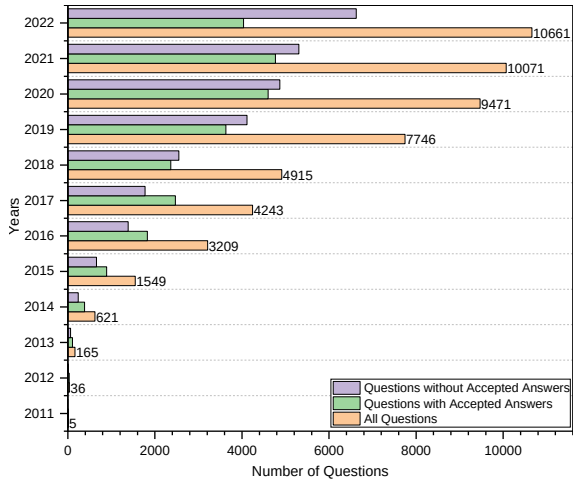

Fig. 2: Count of IaC questions and their related answers

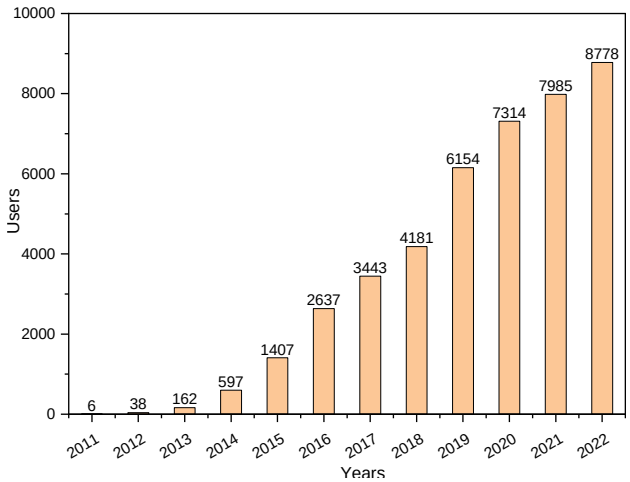

Fig. 3: Count of distinct users participating in IaC discussion

the history of IaC-related questions on SO could be due to several factors. As developers are increasingly adopting IaC as an infrastructure management practice for various application domains, the diversity, and complexity of IaC-related questions could increase, making it more challenging for other users to provide rightful answers. Our analysis could also showcase the limited expertise of the SO community in the field of IaC technology to provide satisfactory answers. Furthermore, as IaC technology is constantly evolving with new tools like Pulumi [54] and CDKTF [55], users might lack up-to-date knowledge about the emerging IaC practices and upgrades.

Finding 3: We observe a sudden increase in the number of questions and unique users between 2018 and 2019. From Figures 2 and 3, we observe a large increase in the number of questions and users in the years 2018 and 2019. Specifically, the number of questions increased by 2,831 (5.37%), while the number of unique users increased by 1,973 (4.62%). To better understand this sudden increase, several significant developments during this period garnered the attention of IaC users which were mentioned in the 2019 yearly review of Terraform [56] and the 2019 yearly event of the Ansible community [57], including the release of updated versions of the Terraform [58] and Ansible [59] tools. Terraform users use HCL's declarative configuration language [60]. However, before 2019, many users encountered issues with HCL, such as handling interpolations, initializing variables with double quotes, and rendering JSON. These problems led users to seek help on SO [61], [62]. In the same year, Ansible released two versions, namely, v2.8 and v2.9 [59], introducing significant improvements such as *Ansible collections* and *Ansible Galaxy* (3.x) [63]. Ansible collections allow users to package and distribute their file configurations across different platforms to promote reusability [64], which impacted the Ansible community [57]. These observations motivate us to further understand the challenges users encounter when dealing with IaC.

RQ1 summary: *The results of RQ1 reveal that a significant number of users use the Stack Overflow platform to find solutions to their IaC-related problems with over 42,702 unique users and 52,692 IaC-related questions.*

B. (Topics) RQ2: What are the main IaC-related topics asked by developers?

Motivation: Identifying the different topics that users discuss in their IaC-related posts provides insights into the different IaC areas where users require assistance. Understanding these topics can help in the identification of poor practices or shortcomings in the IaC tools and the development of new tools to assist IaC users in overcoming their issues. For example, Guerriero et al. [11] identified challenges encountered by IaC users in industrial projects, including the absence of familiar development tools such as debuggers and testing frameworks. This issue can hinder progress in IaC development and evolution. Hence, our goal in RQ2 is to identify the specific topics in which IaC users require assistance.

Approach: As described in Section II-D, we use the LDA algorithm to identify IaC topics. Afterward, we aim to assign labels to these topics based on prior research [35], [36]. Initially, the first three authors examined the top 20 keywords. Then, similar to Abdellatif et al. [36], the authors reviewed 30 randomly related posts for each topic. This phase aims to create a precise and meaningful label that accurately describes the keywords and the related posts. Finally, the authors vote to determine the most appropriate labels, repeating this process until they reach a satisfactory consensus on the topic names.

Finding 1: Our analysis reveals that IaC users on Stack Overflow discuss seven main topics: *Server Configuration*, *Policy Configuration*, *Networking*, *Deployment Pipelines*, *Variable Management*, *Templating*, and *File Management*. We provide in Table II the topics labels, the top 20 related keywords, and the percentage of questions per topic. In the following, we describe each topic and provide examples.

Server Configuration: the server configuration represents the main topic discussed by users with 23% of the questions (see Table II). The keywords "run", "error", and

TABLE II. LDA topics names, part of their stemmed keywords, and percentage of questions per topic

Topic	Related 20 top keywords	Count	Percentage
Server Configuration	[ansibl, run, playbook, host, task, command, instal, error, server, execut, script, fail, inventori, work, user, ssh, python, machin, packag, play]	12,077	23%
Policy Configuration	[aws, cloudform, creat, stack, lambda, templat, function, error, cdk, resourc, polici, api, bucket, code, deploy, role, paramet, iam, gateway, arn]	8,935	17%
Variable Management	[list, output, variabl, string, loop, json, result, condit, ansibl, valu, work, line, item, object, map, filter, task, key, return, data]	7,643	15%
Networking	[creat, instanc, group, terraform, aws, cluster, subnet, vpc, resourc, secur, tag, network, error, rule, address, node, privat, public, add, rout]	6,819	13%
Templating	[templat, azur, resourc, deploy, creat, error, servic, appli, arm, account, manag, set, paramet, secret, code, storag, access, key, group, work]	6,562	12%
Deployment Pipelines	[terraform, run, provid, version, state, configur, resourc, chang, updat, imag, code, appli, work, build, time, deploy, environ, plan, project, pipelin]	5,789	11%
File Management	[file, modul, variabl, role, var, key, directori, set, main, pass, path, defin, yml, user, work, copi, environ, folder, creat, call]	4,867	9%

"fail" associated with the server configuration-related posts indicate that users mainly seek help with fixing broken server configurations during an execution. For instance, the question in Quote 2, viewed 273K times, illustrates an example where a developer encountered an issue with starting a script to execute commands on a remote server that were related to specific parameters [65]. Additionally, users encounter challenges with installing and updating server dependencies and packages necessary for application and service execution [66] and establishing secure connections between separate servers [67]. Furthermore, we found an issue related to managing database servers that involve storing and retrieving data [68]. This topic aligns with Morris and Thompson [1], highlighting the operations team's ongoing struggles in obtaining solutions for server-related issues.

How to execute a shell script on a remote server using Ansible?
💬 "*I am planning to execute a shell script on a remote server using Ansible playbook...When I run the playbook, the transfer successfully occurs but the script is not executed.*"

Quote 2: A question about the server configuration topic [65]

Policy Configuration: the policy configuration represents the second most discussed topic with 17%. A policy configuration refers to activities made by developers to configure policies that control access to infrastructure resources, which can also be illustrated from associated keywords like "create", "stack", "bucket", "policy", "role", "paramet", and "gateway". This topic aligns with the documentation of Terraform that emphasizes the "*Policy-as-Code*" practice [69]. However, misconfigurations of policies can present a significant challenge for developers, resulting in vulnerabilities that could propagate to unintended resources. Rahman et al. [70] have identified 7 types of security smells such as *Default Admin* that can be present in policies configuration files. One popular related question, highlighted in Quote 3, attracted over 126k views and 147 scores. The developer encountered an issue when missing rules for resource deployment [71]. In Quote 4, the accepted answer provides conditions that can resolve the issue.

AWS Lambda: The provided execution role does not have permissions to call DescribeNetworkInterfaces on EC2
💬 "*..The provided execution role does not have permissions to call DescribeNetworkInterfaces on EC2..And then, I give the role "AdministratorAccess" Policy, I can save my source code correctly..This role can run Functions successfully before today..Is anyone know this error?*"

Quote 3: A question related to policy Configuration [71]

💬 "*This error is common if you try to deploy a Lambda in a VPC without giving it the required network interface related permissions ec2:DescribeNetworkInterfaces, ec2:CreateNetworkInterface, and ec2:DeleteNetworkInterface (see AWS Forum)..For example, this a policy that allows to deploy a Lambda into a VPC:..* "

Quote 4: Accepted answers to the question in Quote 3 [71]

Variable Management: the variables management topic comprises 15% of the posts in our dataset. It concerns issues that result from handling variables in IaC files, which can store various information about target platforms. Quote 5 illustrates a popular question (*i.e.,* 293K views and 156 scores) where the developer seeks help in confirming the correct declaration of a variable during the execution of configuration tasks [72]. Our manual analysis revealed that the issues in this area also concern data structures such as lists, maps, and objects, which developers use to manipulate a group of variables in IaC files (cf. Quote 6) [73]. The associated keywords to this topic, such as "list", "output", "variables", "string" and "loop" indicate the coding words that can be used in infrastructure code to deal with variables. Furthermore, this topic corresponds to the work conducted by Rahman et al. [74] which revealed that developers face difficulties related to variables while utilizing the Puppet configuration tool [75].

How to run a task when variable is undefined in Ansible?
💬 "*I am looking for a way to perform a task when Ansible variable is not registers or undefined... * "

Quote 5: A example of variable management topic [72]

How can I iterate through a map variable in terraform?
💬 "*Im trying to iterate through a variable type map...I want to create multiple resources with the different account names*"

Quote 6: A question related to iterating over variables [73]

Networking: the networking account for 13% of the questions in our dataset. Developers discuss how to establish and regulate network resources inside infrastructure platforms. In Table II, we see that the existing keywords for this topic like "create", "group", "subnet", "vpc", "address", "private", "public", and "rout" can highlight the activities related to the networking. The findings of Rahman et al. [74] indicate that 3.8% of SO questions in their study are around networking. For example, in Quote 7, a developer builds a specific network architecture to connect three subnetworks, but he struggled to allow traffic from each subnetwork [76]. Based on our analysis of related posts, developers handle network resources such as virtual cloud networks [77], domain name systems (DNS), gateways [78], and load-balancing rules [79] on various infrastructure providers such as AWS, Azure, and GCP. These

providers use different terminology for the same network resources. For instance, Azure refers to the virtual network as a "Vnet", while GCP and AWS call it a "Virtual Private Cloud (VPC)".

How to route between three subnets in an AWS VPC w/ Terraform?
💬 "*Can someone please provide the ideal or proper way to define these three subnets (public 'web', public 'dmz' w/ a bastion, and private 'app') so that instances on the 'web' subnet can access the 'app' subnet and that the bastion host in the DMZ can provision instances in the private 'app' subnet?*"

Quote 7: An example of question networking topic [76]

🏷 **Templating:** the templating related posts cover around 12% of the posts. Templating refers to using files with dynamic content rather than complex static files. The questions within this topic revolve around the challenges that arise when developers attempt to template files. This topic is in line with the findings of Rahman et al. [74] who have identified that developers face errors resulting from using templating tools like Embedded Ruby (ERB)-based templates [80]. For instance, in Quote 8, a developer solicits aid to avoid repeating the declaration of resources [81].

Can I loop over properties in ARM templates?
💬 "*I have an ARM template where I set up a load balancer and I want to add a number of port openings by adding rules and probes to the LB...What I would like to do is to have an array of the port numbers I want to create rules and probes for and loop over those instead of explicitly having to write each rule and probe as a property for the resource...*"

Quote 8: An example related to the templating topic [81]

Associated keywords to this topic like "account", "manage", "set", "parameter", "secret", "storage", "access" and "key" suggest activities related to managing key secrets. Even though Guerriero et al. [11] have shown that loading dynamically the secrets are one of the best practices for developing IaC practices, developers face challenges when attempting to template sensitive files that contain hard-coded secrets, such as passwords. For instance, a developer sought help in Quote 9 to include a certification that grants access to their resources [82]. The accepted solution proposed adding a specific declaration to the template to dynamically load the certification.

ARM Template for Importing Azure Key Vault Certificate in Function App
💬 "*..I have a function app which calls another API with a certificate. This certificate (.pfx) file is already present in the key vault. I am using below ARM template to import the certificate to SSL settings of the function app.....*"

Quote 9: An example of question related to treating secrets with templates [82]

🏷 **Deployment Pipelines:** the deployment pipeline covers approximately 11% of the questions in our dataset. In this topic, developers face problems running IaC files in deployment pipelines. These challenges result from the misconfiguration of special features and specific cases when using continuous delivery tools such as GitHub Action [83] and Jenkins [84]. For example, in Quote 10, an example of a question where the developer faces an issue setting up the appropriate Terraform command that allows the correct deployment of some resources [85]. Additionally, keywords such as "terraform", "run", "version", "change", "build", "deploy", "plan", "time", and "pipeline" may indicate that developers take a considerable time to continuously deploy and change their infrastructure code. The issues in this topic are related to the finding of Guerriero et al. [11] who have illustrated that developers take a considerable time to deploy their IaC files.

How can I remove all the extraneous output from redirected output in GitHub Actions?
💬 "*I have a GitHub Actions workflow that uses Terraform for its deployment. When Terraform is done, I want to take the Terraform output and send it to the next job in the workflow so that pieces can be extracted an used. Specifically, my Terraform deploys an Azure Function and then outputs the function app name. This then gets used to tell the next job where to deploy the Function code.*"

Quote 10: A question related to deployment pipelines [85]

🏷 **File Management:** the file management represents 9% of the IaC posts. We observe clearly that the related keywords such as a "file", "module", "directory", "path", and "yml" suggest activities concerning files. In this topic, developers are interested in structuring IaC files, breaking them down into smaller, reusable files (see Quote 11) [86], and importing content from other files. These questions are often project-specific and can be challenging to resolve. Additionally, developers encounter issues when treating non-configuration files that could contain sensitive data used in IaC files (see Quote 12) [87]. This topic also is in line with Rahman et al. [74] finding who emphasized that developers encounter issues when performing file operations using the Puppet IaC tool.

How to split an ansible role's 'defaults/main.yml' file into multiple files?
💬 "*In some ansible roles (e.g. roles/my-role/) I've got quite some big default variables files (defaults/main.yml). I'd like to split the main.yml into several smaller files. Is it possible to do that? I've tried creating the files defaults/1.yml and defaults/2.yml, but they aren't loaded by ansible.*"

Quote 11: Example of splitting IaC files [86]

Ansible: how to import GPG private key from file?
💬 "*Suppose the private key resides as a file on my local machine, how can I import the private key on a remote machine using Ansible? I've searched the documentation, but Ansible does not seem to have a module for this task.*"

Quote 12: Example of importing a private key file [87]

RQ2 summary: *Our analysis reveals that users mainly inquire about seven topics, including Server Configuration, Policy Configuration, Networking, Deployment Pipelines, Variable Management, Templating, and File Management. Notably, discussions related to Server Configuration account for approximately 23% of all IaC discussions, indicating its significant role in the development of IaC projects.*

C. *(Challenges) RQ3: Which IaC-related topics are most popular and difficult?*

Motivation: We aim to investigate topics that received significant attention from developers (*i.e.,* popular topics) and those that pose challenges in providing answers (*i.e.,* difficult topics).
Approach: To assess the popularity and difficulty levels of IaC topics, we follow a similar approach as prior studies [26], [36], [41]. For popularity, we measure the mean view count and score count of questions pertaining to each topic. The view count indicates the level of interest within the IaC community, while the score reflects the usefulness of the posts. Higher average values signify the most popular topic. Further, to determine topic difficulty, we count the percentage of questions without an answer (*i.e.,* unanswered questions), questions without accepted answers, and the median time to provide the accepted answer. A higher percentage of unanswered questions indicates that the topic is difficult to answer, while the percentage of questions without accepted answers indicates the extent to which developers master the topic.
Finding 1: Our analysis reveals that the *Sever Configuration* and *File Management* are the most popular topics. According to Table III, we observe that *Server Configuration* and *File Management* topics are the most popular among the six topics in terms of views and score average, while *Templating* topic is the least popular. Although the number of questions related to *Server Configuration* topics exceeds those of *File Management*, the latter has the highest average score count and view count by a margin of 97.43% and 0.24%, respectively. For example, in Quote 13, a question related to *File Management* topic is highly viewed (484k) and well-scored (212) [88]. This finding may indicate that users are more engaged to view posts that share similar issues as they encounter.

Ansible: How to delete files and folders inside a directory?
💬 "*...I want to remove all the files and folders inside the web directory and retain the web directory. How can I do that?*"

Quote 13: A question about managing files of web directory using Ansible [88]

Finding 2: Our analysis suggests that the topic of *Deployment Pipelines* is the most difficult in terms of the number of questions that remain unanswered or do not have accepted answers. Table III indicates that among the six topics, discussions related to *Deployment Pipelines* have the highest rate of unanswered questions (19.31%) and questions lacking accepted answers (59.44%). In Quote 14, an example of *Deployment Pipelines* related question was viewed 46k times and scored 55 in our dataset. The question primarily addressed an issue related to updating versions of IaC tools, which makes IaC scripts inconsistent across different versions [89].

"Invalid legacy provider address" error on Terraform
💬 "*I'm trying to deploy a bitbucket pipeline using terraform v0.14.3 to create resources in google cloud. after running terraform command, the pipeline fails...We updated our local version of terraform to v.0.13.0 and then ran...We still get the same error when initiating the bitbucket pipeline. Does anyone know how to get past this error?*"

Quote 14: A question highlighting the problem when using the different Terraform versions [89]

Finding 3: the *Templating* topic has the longest median time for questions to receive accepted answers, indicating its level of difficulty. In terms of time, we find that *Templating* questions took over 8 hours in median, which seems to be challenging for the IaC community to quickly provide accepted answers. An example of a question with this topic that took up to five years to receive an accepted answer is highlighted in Quote 15 [90]. The respondent in the accepted answer gave a simple code snippet and suggests referring to the official documentation. This may indicate that sometimes junior developers do not pay attention and follow the technical documentation.

Apply tags to Azure resource group within resource template file
💬 "*I'm using Powershell to create an Azure resource group from a template file and parameter file...Within the template I'm setting some tags on the resources within the group...I'd like to apply the same tag values to the resource group itself, but I don't see a way to do that within the schema of the template file. Am I missing something?*"

Quote 15: A question related to template files [90]

In contrast, questions related to *Variable Management* had the lowest average percentage of unanswered questions (8.87%) and median hours to receive an accepted answer (1.52 hours), indicating that this topic is relatively easier for users to answer. This topic has posts about how to deal with the syntax and data structure that contains variables in IaC scripts.

RQ3 summary: *On one hand, we found that IaC users frequently discuss questions related to Server Configuration and File Management. On the other hand, questions related to the Deployment Pipelines and Templating topics are challenging in terms of the number of unanswered questions and the time taken to receive an accepted answer.*

IV. Implications and Takeaways

Our study highlighted the growth of the IaC community on SO, the different topics that were discussed, and identified the most popular and difficult topics. In this section, we discuss how the IaC community such as practitioners, researchers, and educators can benefit from our findings.

A. Practitioners

We provide practitioners, such as developers or tool builders, with the following takeaways when practicing IaC.
Enhance the documentation support. We found in RQ1 that the number of IaC questions on SO is increasing. This means that IaC practitioners should make it easier for new users to learn how to use their tools. A good way to do this is by providing clear documentation and helpful tutorials. This will help new IaC users get started and become more confident with the tools they are using.
Attention to the impact of new features. We observed a rise in the number of IaC-related questions and users on SO between 2018 and 2019, which might be due to the improvements in

TABLE III. Popularity and difficulty of IaC topics

Topic	Popularity Metrics		Difficulty Metrics				
	Average Counts		Questions without accepted answers		Questions without answers		Median hours to accepted answers
	Views	Score	Count	Percentage	Count	Percentage	
Server Configuration	**3,689.65**	**2.16**	6,776	56.11%	1,764	14.61%	2.32
Policy Configuration	1,862.48	1.98	4,751	53.17%	1,333	14.92%	4.14
Variable Management	2,917.07	1.68	3,155	41.28%	678	8.87%	1.52
Networking	1,492.53	1.29	3,628	53.20%	1,050	15.40%	3.51
Templating	1,298.22	1.14	3,372	51.39%	816	12.44%	**8.56**
Deployment Pipelines	1,832.59	1.77	3,441	**59.44%**	1,118	**19.31%**	5.34
File Management	**3,787.08**	**2.40**	2,466	50.67%	596	12.25%	1.96

IaC tools such as Terraform, which made notable changes during this period, including updates to the HCL language configuration and the addition of template expressions. While these enhancements are promising, it is important for IaC tool vendors to carefully consider the potential impact of new features and changes on their clients, which could result in unexpected issues.

Improve testing and debugging tools. Based on RQ2 findings, the keywords "error" and "fail" arise in different topics (server configuration, templating), which may indicate that developers need more efficient testing frameworks and debugging tools to resolve the bugs they face. Therefore, tool builders can improve and introduce relevant features in their existing tools. Hence, developers can apply best practices of testing such as Test Driven Development (TDD) when provisioning infrastructure.

Insufficient skill. Deployment pipelines and templating are the most difficult topics as found in RQ3. Meanwhile, IaC users can struggle to find quick solutions related to these topics. Therefore, the Stack Overflow community might need to motivate IaC experts and developers to further contribute to these IaC-related challenges.

B. Researchers

Our findings can assist the research community in further understanding and digging into the IaC engineering practices. While all the topics we identified are important, we believe that researchers should pay more attention to the most challenging ones, as we highlight in the following points.

Investigate the issues on SO between 2018 and 2019. Based on our findings (RQ1), researchers can look at the history of SO discussions, especially between 2018 and 2019, to determine the different reasons and motivations behind the increment of IaC questions and users. Understanding these reasons can shed light on the area in which users are adopting IaC practices and seeking assistance.

Research on the defect of IaC scripts. We suggest researchers focus on developing new approaches to detect potential bugs in IaC files by empirically analyzing users' questions and answers. By looking at the kinds of problems that users are encountering and the solutions that are being offered on platforms like SO, researchers could identify patterns and common mistakes that lead to bugs in IaC files. This could then improve the development of new tools and techniques to help IaC developers avoid these kinds of errors and improve their code quality.

C. Educators

Towards a better introduction of IaC. As a way to make better course materials, educators can look into the different IaC topics from our study. For a complete introduction to the IaC area, they can include topics with more questions and views, like server configuration and file management. They can also provide practical examples by using Ansible, which is a prominent IaC tool in the area of server configuration.

More focus on Difficult Iac Topics. Educators should pay special attention in their teaching material to deployment pipelines and challenges related to IaC templates. Since templates use dynamic expressions and blocks, learners may need specific and practical explanations of template expressions.

V. Threats to validity

Internal threats to validity could be related to the threats that can produce errors in our study. One such threat concerns extracting the IaC posts using various tags, which can miss some posts or result in irrelevant posts. To mitigate this threat, we used relevance and significance metrics that are used in previous studies [41], [91], [92]. We then manually inspected the largest tag set to remove irrelevant tags. We believe that this process produced significantly relevant IaC tags. Another potential factor could be related to selecting the optimal number of topics and learning iterations. To address this factor, we tune the parameters of the topic modeling by using the genetic algorithm (GA), which was used in [93], [94]. Additionally, LDA is a probabilistic technique that can fail to cluster correctly some posts [47]. Therefore, we ran the LDA model three times and found no substantial difference in the results. The average number of 20 keywords, for each topic, that appeared in the three iterations was 139.66. In particular, in the second iteration, the term *"ansible"* replaced *"role"* keyword in the file management topic, but this change reverted to the initial results from the first iteration illustrated in Table II.

Construct threats to validity concern the threats that impact the consistency between experience and theory in our study. A possible threat is related to assigning labels to the topics, as it requires a comprehension of the related posts and matching them with topic keywords. To mitigate this threat, we followed previous studies [35]–[37], by individually examining the 20 most frequent keywords for each topic and reviewing randomly chosen 30 posts by the first and the second author. Then, through several meetings, the authors discussed the topics' labels and how representative are their corresponding keywords.

Another threat could be related to the metrics used to calculate the popularity and difficulty of IaC topics. We followed the previous studies that used the average count of views and scores to study the popularity of topics [35], [36]. For the difficulty of the topic, we utilize the percentage of unanswered questions, the percentage of questions without accepted answers, and the median time to provide an accepted answer [26].

External threats to validity can impact the generalizability of our findings which could limit the external validity of our study since our work focuses on analyzing the questions and answers related to IaC on SO. While we identified a considerable number of IaC-related questions and answers from diverse users on Stack Overflow, it is worth noting that users could use other alternative Q&A platforms to seek solutions. Therefore, we recommend replicating our study on other Q&A platforms.

VI. Related work

Infrastructure-as-Code (IaC). Researchers have conducted several studies to support IaC developers and analyze their challenges. Guerriero et al. [11] investigated the adoption, support, and challenges of IaC from practitioners' perspectives in the industry. By conducting 44 interviews with senior developers, the authors highlighted the need to test and maintain IaC code, as well as the need for consistent IaC development tools, such as testing frameworks and debuggers. Opdebeeck et al. [21] empirically study semantic versioning in Ansible roles (*i.e.,* reusable configuration blocks that can be shared as third-party libraries) to explore the types of changes that trigger certain types of version bumps. The study utilized a Random Forest classifier to predict applicable version bumps for Ansible role releases. They confirmed their findings with 6 developers and observed that the majority of Ansible developers adhere to the rules specified by the semantic versioning format. Ansible's developers follow the semantic versioning format to characterize package releases. Rahman et al. [19] conducted a mixed-method approach to identify development antipatterns by analyzing 2,138 open-source IaC scripts and conducting a survey with 51 practitioners. Their findings show 5 five development antipatterns namely, "boss is not around", "many cooks spoil", "minors are spoiler", "silos", and "unfocused contribution", demonstrating the importance of "as code" activities in IaC. Later, Rahman et al. [20] developed a defect taxonomy for IaC scripts to help practitioners improve the quality of IaC. The authors surveyed 66 practitioners to assess their agreement level with this identified defect taxonomy, finding that practitioners mostly agree with idempotency. Rahman et al. [74] also analyzed discussions on SO to identify challenges related to Configuration as Code (CaC), which mainly involves managing software systems independently of their infrastructures, while IaC focuses more on the infrastructure. They manually analyzed 2,758 questions created between 2010 and 2016 related solely to the puppet tool [75] and found that developers discussed 16 categories of questions around CaC.

Although there are recent studies that explored the difficulties faced by IaC developers and offered support, to the best of our knowledge, no prior studies have manually extracted all the challenges developers are facing on Q&A platforms in the context of IaC. Therefore, we aim to conduct a large-scale analysis to identify the issues and challenges in SO. This study can complement the existing efforts that are done with practitioners [11], [19]–[21].

Stack Overflow Studies. Despite several studies that have explored the perspectives of software developers by analyzing SO posts, our focus is only on studies that have used LDA to study different development areas such as deep learning development [95], docker development [41] and big data [35]. Rosen and Shihab [91] investigated discussions among mobile developers, which use different mobile platforms, and categorized questions into why, what, and how questions, as well as identified 40 topics classified into 5 categories.

While the existing studies attempt mainly to cluster SO posts, they struggle to find the optimal parameters, such as the number of topics and learning iterations. They usually experiment with defined intervals from prior works [35], [37]. A few studies have explored genetic algorithms to fine-tune LDA parameters (LDA-GA) [93], [94]. In the latter studies, they mostly applied the silhouette coefficient, as a fitness function, which is used to measure the similarity between clusters in classical clustering models [96]. This measure is, to some extent, harder to show a correlation with human interpretation [93]. We believe that an appropriate fitness function is needed to improve the results of LDA-GA. In our study, we used a different fitness function based on topics coherence that is in line with human judgments [53].

VII. Conclusions and future work

We conducted the first large-scale empirical study on Stack Overflow (SO) to explore the growth, topics, and challenges in the Infrastructure-as-Code (IaC) practice. Our findings reveal that IaC-related posts on SO have increased significantly from 2011 to 2022, indicating a growing interest in IaC. Using Latent Dirichlet Allocation (LDA) to categorize posts, we identified 7 primary IaC topics, including *Server Configuration*, *Policy Configuration*, *Networking*, *Deployment Pipelines*, *Variable Management*, *Templating*, and *File Management*. We discovered that IaC developers are primarily concerned with configuring servers and managing IaC files, while *deployment pipelines* and *templating* are the most difficult topics. Our study provides valuable takeaways for IaC researchers, practitioners, and educators to better support IaC in practice.

In future work, we plan to conduct interviews with IaC developers to gain insights into their practical challenges and compare them with our results. Additionally, we aim to conduct further empirical studies to identify a checklist of IaC-related challenges and explore conversations related to IaC on other Q&A platforms to supplement our findings.

Acknowledgment

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) RGPIN-2018-05960.

REFERENCES

[1] K. Morris and B. Thompson, *Infrastructure as Code*. O'Reilly Media, 2nd ed., 2020.

[2] R. Hat, "Ansible is simple it automation." https://www.ansible.com/. Accessed on: April 26, 2023.

[3] HashiCorp, "Terraform by hashicorp." https://www.terraform.io/. Accessed on: April 12, 2023.

[4] A. W. Services, "Provision infrastructure as code - aws cloudformation - aws." https://aws.amazon.com/cloudformation/. Accessed on: April 12, 2023.

[5] Microsoft, "Arm template documentation." https://learn.microsoft.com/en-us/azure/azure-resource-manager/templates/. Accessed on: April 12, 2023.

[6] J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional, 2010.

[7] "Using Terraform to Manage Infrastructure." https://shopify.engineering/manage-infrastructure-with-terraform. Accessed on: April 23, 2023.

[8] "Crane: Uber's Next-Gen Infrastructure Stack." https://www.uber.com/en-IN/blog/crane-ubers-next-gen-infrastructure-stack/. Accessed on: April 23, 2023.

[9] S. Dalla Palma, D. Di Nucci, F. Palomba, and D. Tamburri, "Within-project defect prediction of infrastructure-as-code using product and process metrics," *IEEE Transactions on Software Engineering*, vol. PP, pp. 1–1, 01 2021.

[10] N. Bessghaier, M. Sayagh, A. Ouni, and M. W. Mkaouer, "What constitutes the deployment and run-time configuration system? an empirical study on openstack projects," *ACM Transactions on Software Engineering and Methodology*, 2023.

[11] M. Guerriero, M. Garriga, D. A. Tamburri, and F. Palomba, "Adoption, support, and challenges of infrastructure-as-code: Insights from industry," in *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 580–589, 2019.

[12] A. Rahman, R. Mahdavi-Hezaveh, and L. Williams, "A systematic mapping study of infrastructure as code research," *Information and Software Technology*, vol. 108, pp. 65–77, 2019.

[13] Y. Jiang and B. Adams, "Co-evolution of infrastructure and source code - an empirical study," in *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pp. 45–55, 2015.

[14] S. Dalla Palma, D. Nucci, F. Palomba, and D. Tamburri, "Towards a catalogue of software quality metrics for infrastructure code," *Journal of Systems and Software*, vol. 170, p. 110726, 07 2020.

[15] R. Opdebeeck, A. Zerouali, and C. De Roover, "Smelly variables in ansible infrastructure code: detection, prevalence, and lifetime," in *19th International Conference on Mining Software Repositories*, pp. 61–72, 2022.

[16] A. Rahman, C. Parnin, and L. Williams, "The seven sins: Security smells in infrastructure as code scripts," in *IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pp. 164–175, 2019.

[17] T. Sharma, M. Fragkoulis, and D. Spinellis, "Does your configuration code smell?," in *13th International Conference on Mining Software Repositories*, pp. 189–200, 2016.

[18] N. Borovits, I. Kumara, D. Di Nucci, P. Krishnan, S. D. Palma, F. Palomba, D. A. Tamburri, and W.-J. v. d. Heuvel, "Findici: Using machine learning to detect linguistic inconsistencies between code and natural language descriptions in infrastructure-as-code," *Empirical Softw. Engg.*, vol. 27, dec 2022.

[19] A. Rahman, E. Farhana, and L. Williams, "The 'as code' activities: Development anti-patterns for infrastructure as code," *Empirical Softw. Engg.*, vol. 25, p. 3430–3467, sep 2020.

[20] A. Rahman, E. Farhana, C. Parnin, and L. Williams, "Gang of eight: A defect taxonomy for infrastructure as code scripts," in *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pp. 752–764, 2020.

[21] R. Opdebeeck, A. Zerouali, C. Velázquez-Rodríguez, and C. D. Roover, "Does infrastructure as code adhere to semantic versioning? an analysis of ansible role evolution," in *2020 IEEE 20th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pp. 238–248, 2020.

[22] A. Barua, S. W. Thomas, and A. E. Hassan, "What are developers talking about? an analysis of topics and trends in stack overflow," *Empirical Softw. Engg.*, vol. 19, p. 619–654, jun 2014.

[23] W. Zhu, H. Zhang, A. E. Hassan, and M. W. Godfrey, "An empirical study of question discussions on stack overflow," *ACM Transactions on Software Engineering and Methodology*, vol. 27, nov 2022.

[24] Sebaszw, "Ansible: deploy on multiple hosts in the same time." https://stackoverflow.com/q/21158689. Accessed on: April 29, 2023.

[25] H. Zhang, S. Wang, H. Li, T.-H. Chen, and A. E. Hassan, "A study of c/c++ code weaknesses on stack overflow," *IEEE Transactions on Software Engineering*, vol. 48, no. 7, pp. 2359–2375, 2022.

[26] A. Peruma, S. Simmons, E. A. AlOmar, C. D. Newman, M. W. Mkaouer, and A. Ouni, "How do i refactor this? An empirical study on refactoring trends and topics in Stack Overflow," *Empirical Software Engineering*, vol. 27, p. 11, Oct. 2021.

[27] A. Ouni, I. Saidani, E. Alomar, and M. W. Mkaouer, "An empirical study on continuous integration trends, topics and challenges in stack overflow," in *IEEE International Conference on Evaluation and Assessment in Software Engineering (EASE)*, pp. 1–10, 2023.

[28] Z. Wan, X. Xia, and A. E. Hassan, "What do programmers discuss about blockchain? a case study on the use of balanced lda and the reference architecture of a domain to capture online discussions about blockchain platforms across stack exchange communities," *IEEE Transactions on Software Engineering*, vol. 47, no. 7, pp. 1331–1349, 2021.

[29] "Answer to "Database schema documentation for the public data dump and SEDE"." https://meta.stackexchange.com/a/326361. Accessed on: April 25, 2023.

[30] "Replication package for the paper: "What Do Infrastructure-as-Code Practitioners Discuss: An empirical Study on Stack Overflow"." https://figshare.com/s/a0a1d1331142a11bb120. Accessed on: May 02, 2023.

[31] "Stack Exchange Data Dump : Stack Exchange, Inc. : Free Download, Borrow, and Streaming." https://archive.org/details/stackexchange. Accessed on: April 23, 2023.

[32] "stackexchange directory." https://archive.org/download/stackexchange. Accessed on: April 23, 2023.

[33] S. Baltes, L. Dumani, C. Treude, and S. Diehl, "Sotorrent: reconstructing and analyzing the evolution of stack overflow posts," in *Proceedings of the 15th International Conference on Mining Software Repositories* (A. Zaidman, Y. Kamei, and E. Hill, eds.), pp. 319–330, ACM, 2018.

[34] "Introduction to tables | BigQuery." https://cloud.google.com/bigquery/docs/tables-intro. Accessed on: April 23, 2023.

[35] M. Bagherzadeh and R. Khatchadourian, "Going big: A large-scale study on what big data developers ask," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2019, (New York, NY, USA), p. 432–442, Association for Computing Machinery, 2019.

[36] A. Abdellatif, D. Costa, K. Badran, R. Abdalkareem, and E. Shihab, "Challenges in chatbot development: A study of stack overflow posts," in *Proceedings of the 17th International Conference on Mining Software Repositories*, MSR '20, (New York, NY, USA), p. 174–185, Association for Computing Machinery, 2020.

[37] S. Ahmed and M. Bagherzadeh, "What do concurrency developers ask about? a large-scale study using stack overflow," in *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '18, (New York, NY, USA), Association for Computing Machinery, 2018.

[38] "Production-Grade Container Orchestration." https://kubernetes.io/. Accessed on: April 29, 2023.

[39] k. krippendorff, "Computing krippendorff's alpha-reliability," 2011.

[40] J. Hughes, "krippendorffsalpha: An r package for measuring agreement using krippendorff's alpha coefficient," 2021.

[41] M. U. Haque, L. H. Iwaya, and M. A. Babar, "Challenges in docker development: A large-scale study using stack overflow," in *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, ESEM '20, (New York, NY, USA), Association for Computing Machinery, 2020.

[42] "Mallet english stop words." https://github.com/mengjunxie/ae-lda/blob/master/misc/mallet-stopwords-en.txt. Accessed on: March 26, 2023.

[43] E. Loper and S. Bird, "NLTK: the natural language toolkit," *CoRR*, vol. cs.CL/0205028, 2002.

[44] M. F. Porter, *An Algorithm for Suffix Stripping*, p. 313–316. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997.

[45] A. Ouni, I. Saidani, E. Alomar, and M. W. Mkaouer, "An empirical study on continuous integration trends, topics and challenges in stack overflow," in *27th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, pp. 141–151, 2023.

[46] M. Openja, B. Adams, and F. Khomh, "Analysis of modern release engineering topics : – a large-scale study using stackoverflow –," in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 104–114, 2020.

[47] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, p. 993–1022, mar 2003.

[48] A. K. McCallum, "Mallet: A machine learning for language toolkit." http://mallet.cs.umass.edu. Accessed on: March 26, 2023.

[49] L. R. Biggers, C. Bocovich, R. Capshaw, B. P. Eddy, L. H. Etzkorn, and N. A. Kraft, "Configuring latent Dirichlet allocation based feature location," *Empirical Software Engineering*, vol. 19, pp. 465–500, June 2014.

[50] Y. Jiang, G. Tong, H. Yin, and N. Xiong, "A pedestrian detection method based on genetic algorithm for optimize xgboost training parameters," *IEEE Access*, vol. PP, pp. 1–1, 08 2019.

[51] N. Gorgolis, I. Hatzilygeroudis, Z. Istenes, and L. G. Gyenne, "Hyperparameter optimization of lstm network models through genetic algorithm," in *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pp. 1–4, 2019.

[52] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[53] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, p. 399–408, 2015.

[54] "Pulumi - Infrastructure as Code in any Programming Language." https://www.pulumi.com/. Accessed on: April 23, 2023.

[55] "CDK for Terraform." https://github.com/hashicorp/terraform-cdk. Accessed on: April 23, 2023.

[56] T. Team, "Hashicorp terraform 0.12 preview." https://www.hashicorp.com/blog/terraform-0-1-2-preview. Accessed on: March 26, 2023.

[57] J. Geerling, "Notes from the ansiblefest atlanta 2019 ansible contributor summit." https://www.jeffgeerling.com/blog/2019/notes-ansiblefest-atlanta-2019-ansible-contributor-summit. Accessed on: March 26, 2023.

[58] T. Team, "Terraform releases." https://releases.hashicorp.com/terraform/. Accessed on: March 26, 2023.

[59] "Ansible releases." https://github.com/ansible/ansible/releases. Accessed on: March 26, 2023.

[60] "Terraform configuration syntax." https://developer.hashicorp.com/terraform/language/syntax/configuration. Accessed on: March 26, 2023.

[61] "How to prepare rendered json for aws-cli in terraform?." https://stackoverflow.com/questions/52081329/how-to-prepare-rendered-json-for-aws-cli-in-terraform. Accessed on: March 26, 2023.

[62] "Terraform > unescaped interpolations." https://stackoverflow.com/questions/48805418/terraform-unescaped-interpolations. Accessed on: March 26, 2023.

[63] "Ansible galaxy." https://galaxy.ansible.com/?extIdCarryOver=true&sc_cid=701f2000001OH7YAAW. Accessed on: March 26, 2023.

[64] "Developing collections." https://docs.ansible.com/ansible/latest/dev_guide/developing_collections.html. Accessed on: March 26, 2023.

[65] "How to execute a shell script on a remote server using Ansible?." https://stackoverflow.com/q/21160776. Accessed on: April 17, 2023.

[66] "No package matching atomic-openshift-utils." https://stackoverflow.com/q/56961282. Accessed on: April 17, 2023.

[67] "How to connect to WIndows node using openSSH and Ansible?." https://stackoverflow.com/q/45696024. Accessed on: April 17, 2023.

[68] "Ansible how to correctly specify socket value when resetting Mysql password on Raspian." https://stackoverflow.com/q/66959406. Accessed on: April 17, 2023.

[69] "Sentinel by HashiCorp." https://docs.hashicorp.com/sentinel/concepts/policy-as-code. Accessed on: March 26, 2023.

[70] A. Rahman, C. Parnin, and L. Williams, "The seven sins: Security smells in infrastructure as code scripts," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pp. 164–175, 2019.

[71] "AWS Lambda:The provided execution role does not have permissions to call DescribeNetworkInterfaces on EC2." https://stackoverflow.com/q/41177965. Accessed on: April 17, 2023.

[72] "How to run a task when variable is undefined in ansible?." https://stackoverflow.com/q/30119973. Accessed on: April 17, 2023.

[73] "How can I iterate through a map variable in terraform." https://stackoverflow.com/q/57503110. Accessed on: April 17, 2023.

[74] A. Rahman, A. Partho, P. Morrison, and L. Williams, "What questions do programmers ask about configuration as code?," in *2018 IEEE/ACM 4th International Workshop on Rapid Continuous Software Engineering (RCoSE)*, pp. 16–22, 2018.

[75] Perforce, "Puppet infrastructure & it automation at scale." https://www.puppet.com/. Accessed on: April 12, 2023.

[76] "How to route between two subnets in an AWS VPC w/ Terraform?." https://stackoverflow.com/q/35822830. Accessed on: April 17, 2023.

[77] "How to create an Amazon VPC using AWS CloudFormation?." https://stackoverflow.com/q/10103154. Accessed on: April 17, 2023.

[78] "How to Attach Elastic IP to NatGateway via Cloud Formation." https://stackoverflow.com/q/66131377. Accessed on: April 17, 2023.

[79] "The target group does not have an associated load balancer." https://stackoverflow.com/q/53971873. Accessed on: April 17, 2023.

[80] "Language: Embedded ruby erb template syntax." https://www.puppet.com/. Accessed on: April 29, 2023.

[81] "Can I loop over properties in ARM templates?." https://stackoverflow.com/q/41882841. Accessed on: April 17, 2023.

[82] "ARM Template for Importing Azure Key Vault Certificate in Function App." https://stackoverflow.com/q/61045238. Accessed on: April 17, 2023.

[83] GitHub, "Github actions documentation." https://docs.github.com/en/actions. Accessed on: March 26, 2023.

[84] "Jenkins." https://www.jenkins.io/. Accessed on: March 26, 2023.

[85] "How can I remove all the extraneous output from redirected output in GitHub Actions?." https://stackoverflow.com/q/66496105. Accessed on: April 17, 2023.

[86] "How to split an ansible role's 'defaults/main.yml' file into multiple files?." https://stackoverflow.com/q/51818264. Accessed on: April 17, 2023.

[87] "Ansible: how to import GPG private key from file?." https://stackoverflow.com/q/57450590. Accessed on: April 17, 2023.

[88] "Ansible: How to delete files and folders inside a directory?." https://stackoverflow.com/q/38200732. Accessed on: April 17, 2023.

[89] ""Invalid legacy provider address" error on Terraform." https://stackoverflow.com/q/65396812. Accessed on: April 18, 2023.

[90] "Apply tags to Azure resource group within resource template file." https://stackoverflow.com/q/27994111. Accessed on: April 18, 2023.

[91] C. Rosen and E. Shihab, "What are mobile developers asking about? a large scale study using stack overflow," *Empirical Softw. Engg.*, vol. 21, p. 1192–1223, jun 2016.

[92] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, "The impact of automated parameter optimization on defect prediction models," *CoRR*, vol. abs/1801.10270, 2018.

[93] A. Panichella, B. Dit, R. Oliveto, M. Di Penta, D. Poshynanyk, and A. De Lucia, "How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms," in *2013 35th International Conference on Software Engineering (ICSE)*, pp. 522–531, 2013.

[94] X.-L. Yang, D. Lo, X. Xia, Z. Wan, and J.-L. Sun, "What security questions do developers ask? a large-scale study of stack overflow posts," *Journal of Computer Science and Technology*, vol. 31, pp. 910–924, 09 2016.

[95] J. Han, E. Shihab, Z. Wan, S. Deng, and X. Xia, "What do programmers discuss about deep learning frameworks," *Empirical Software Engineering*, vol. 25, 07 2020.

[96] J. Kogan, *Introduction to Clustering Large and High-Dimensional Data*. USA: Cambridge University Press, 2007.