Software and Hardware of Computer, Network, Telecommunication, Control, and Measurement Systems

Компьютерные сети, вычислительные, телекоммуникационные, управляющие и измерительные систем

Research article
DOI: https://doi.org/10.18721/JCSTCS.18312
UDC 004.896

AUTOMATION OF PREPARATION AND DEPLOYMENT OF INFORMATION INFRASTRUCTURE OF CLOUD SERVICES USING THE ANSIBLE TOOL

A.A. Kiseleva[1] ✉, *I.V. Nikiforov*[1], *M.V. Shishko*[2], *S.M. Ustinov*[1]

[1] Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russian Federation;

[2] Gazpromneft Information Technology Operator LLC, St. Petersburg, Russian Federation

✉ aleksandrakiseleva2001@gmail.com

Abstract. The preparation and deployment of information technology infrastructure when implementing corporate collaboration platforms in industrial companies takes up to 40% of total time spent on a project. Automation of these processes helps reduce indicated time costs. This article proposes an approach to automate the preparation and deployment of information infrastructure, as well as to automate the installation of cloud services using the Ansible application. The R7-office platform is chosen as the implemented solution, which has basic necessary functionality for working with documents of various formats, enables collaborative document editing and meets state requirements for import substitution of components. The approach is based on an algorithm for preparing information technology infrastructure and installing the target platform using a software tool written in the Python programming language. It enables automatic parsing of formalized requirements, generating access rights requests for users and creating playbooks for configuring the infrastructure using Ansible. According to the experimental results, the implementation of the proposed approach reduces the labor intensity of the infrastructure preparation process by 45%.

Keywords: R7-office, collaborative editing, automation, IT infrastructure, Ansible

Citation: Kiseleva A.A., Nikiforov I.V., Shishko M.V., Ustinov S.M. Automation of preparation and deployment of information infrastructure of cloud services using the Ansible tool. Computing, Telecommunications and Control, 2025, Vol. 18, No. 3, Pp. 131–143. DOI: 10.18721/JCSTCS.18312

Научная статья
DOI: https://doi.org/10.18721/JCSTCS.18312
УДК 004.896

АВТОМАТИЗАЦИЯ ПОДГОТОВКИ И РАЗВЕРТЫВАНИЯ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ ОБЛАЧНЫХ СЕРВИСОВ С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТА ANSIBLE

А.А. Киселева[1] ✉, *И.В. Никифоров*[1], *М.В. Шишко*[2], *С.М. Устинов*[1]

[1] Санкт-Петербургский политехнический университет Петра Великого,
Санкт-Петербург, Российская Федерация;

[2] ООО «Газпромнефть Информационно-Технологический оператор»,
Санкт-Петербург, Российская Федерация

✉ aleksandrakiseleva2001@gmail.com

Аннотация. Подготовка и развертывание информационно-технологической инфраструктуры при внедрении корпоративных платформ совместной работы в индустриальных компаниях может занимать до 40% общего времени, затрачиваемого на проект. Автоматизация этих процессов помогает снизить указанные временные трудозатраты. В данной работе предложен подход, позволяющий автоматизировать подготовку и развертывание информационной инфраструктуры, а также установку облачных сервисов при помощи приложения Ansible. В качестве внедряемого решения выбрана платформа «Р7-офис», которая обладает основным необходимым функционалом работы с документами различного формата, позволяет совместно редактировать документы, а также удовлетворяет требованиям государства по импортозамещению компонент. В основе подхода лежит алгоритм по подготовке информационно-технологической инфраструктуры и установке целевой платформы с помощью программного средства, написанного на языке программирования Python. Оно позволяет производить автоматический разбор формализованных требований, составление заявок на предоставление прав доступа пользователям и формирование плейбуков для настройки инфраструктуры с помощью Ansible. По результатам проведенных экспериментов внедрение предложенного подхода позволяет сократить трудоемкость процесса подготовки инфраструктуры на 45%.

Ключевые слова: Р7-офис, совместное редактирование, автоматизация, ИТ-инфраструктура, Ansible

Для цитирования: Kiseleva A.A., Nikiforov I.V., Shishko M.V., Ustinov S.M. Automation of preparation and deployment of information infrastructure of cloud services using the Ansible tool // Computing, Telecommunications and Control. 2025. Т. 18, № 3. С. 131–143. DOI: 10.18721/JCSTCS.18312

Introduction

Industrial companies widely use office applications for creating documents, spreadsheet, presentations, as well as for communicating and holding video conferences [1, 2]. For project teams, it is important to be able not only to edit files locally, but also to work with them collaboratively [3], since it saves up to 20% of time due to the absence of the need to transfer documents and synchronize them [4].

There are several products that provide collaborative editing services [5]. Some of them support the deployment on the company's internal servers, while others are available only as cloud services. There are both foreign solutions and domestic analogues [6].

It is also important to consider how much time, money and effort will be spent to implement a particular office suite, as this affects the timing and cost of the final product: the more labor costs are required to implement a solution, the longer the implementation and the higher its cost [7]. Therefore, the issue of choosing the method for deploying the platform within the infrastructure is also relevant [8].

Thus, an urgent task for companies is the implementation of a Russian corporate collaboration platform for the effective work of employees in projects with the maintenance of common documentation. At the same time, the implementation process itself can take up to 40% of the project time and require additional resources from the project administrator to generate and track various requests and engineers to directly configure the infrastructure. For this reason, it is necessary to develop a way to automate the preparation of the infrastructure and the deployment of software on it [9–13].

Comparative analysis of office packages

There are many solutions for editing a document by a group of people [3]. These solutions have similar functionality and are aimed at solving similar problems. To make a choice, it is necessary to conduct a comparative analysis of office packages [14, 5]. Let us take six of the most popular solutions: Yandex 360 [15, 16], Microsoft 365 [17, 18], Notion [19], Slite[1] (https://slite.com), Notejoy [20], R7-Office [21, 22].

Comparative analysis of office packages (Table 1) was conducted according to the most important criteria from the point of view of implementation [6]. These criteria are listed below.

1. The "Collaboration" criterion indicates the possibility or impossibility of collaborative document editing. This functionality is key in this article.
2. The criterion of "Completeness of functionality" is a cumulative criterion that reflects how extensive the functionality offered by the office suite is. The extent of the functionality affects how many tasks user is able to perform.
3. The "Cost" criterion indicates the cost of the solution's licenses. This criterion is highly important for business, as it affects the total cost of the solution.
4. The criterion of "Import dependence" indicates how possible it is to use the package in the Russian Federation, as well as whether the manufacturer is domestic. This is one of the key criteria for companies within the framework of the import substitution policy.
5. The "Using plugins/integrations" criterion reflects whether additional plugins and integrations are provided by the software. Such functionality is important for the further development of the solution, taking into account the specifics of the company.
6. The criterion of "Scalability" indicates how much the system can expand. This is important when the number of employees in the company is growing.
7. The "Compatibility with Russian operating systems" criterion reflects whether the package can work with various operating systems, including domestic ones. This criterion is also important in terms of the policy of software import substitution in companies.
8. The criterion of "Deployment on your own servers" indicates the possibility of deploying the software on the company's internal corporate network. This criterion is also the key one, as it directly affects the security of using the solution.
9. The "Maximum number of users" criterion reflects how many users an office suite can accommodate, which affects the scalability of the solution and the growth of the number of users during the development of the company.

Each of the listed criteria has a score, as well as a weight. For criteria 1, 4, 5, 6, 7, 8, the score is 0 or 1, depending on the presence or absence of the specified functionality. For criteria 2, 3, 9, there is a range of values where the maximum score is assigned when the criterion is exhaustively matched, and the minimum is assigned in the case of significant restrictions on the criterion in comparison with other office packages. The weight indicates the importance of the criterion within the framework of the studied task of implementing an office suite.

[1] Slite | AI-powered knowledge base, Available: https://slite.com (Accessed 02.09.2025)

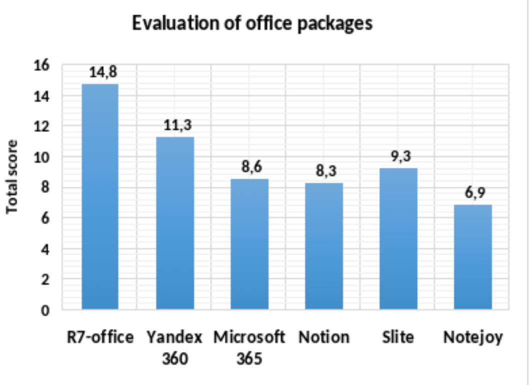


Fig. 1. Evaluation of office packages

As a result of the analysis, each office suite is assigned a total score, which is formed according to the following formula:

$$s = k_1 \times p_1 + k_2 \times p_2 + \ldots + k_n \times p_n,$$

where k_n is the score according to the criterion n, p_n is the weight by criterion n.

Table 1

Comparative analysis of office packages

Criteria	Range	Weight	Yandex 360	Microsoft 365	Notion	Slite	Notejoy	R7-office
Collaboration	0/1	1	1	1	1	1	1	1
Full functionality	1.5	1	5	5	3	3	2	5
Cost	1.5	0.5	3	1	2	4	4	5
Import dependence	0/1	1	1	0	0	0	0	1
Using plugins/integrations	0/1	0.5	0	0	1	1	1	1
Ability to scale	0/1	0.7	1	1	1	1	1	1
Compatibility with Russian operating systems	0/1	1	0	0	0	0	0	1
Deployment of own servers	0/1	1	0	0	0	0	0	1
Maximum number of users	1.3	0.7	3	2	3	3	1	3
Total			11.3	8.6	8.3	9.3	6.9	14.8

To decide on choosing an office suite, a bar chart has been compiled reflecting the results of the comparative analysis (Fig. 1).

Based on the comparative analysis of office packages, it can be concluded that the R7-office shows the best results according to most of the criteria presented.

Comparative analysis of deployment tools

Let us take four of the most popular tools: Puppet [23, 24], Chef [25], Ansible [26–28], SaltStack [29, 30].

Comparative analysis of deployment tools (Table 2) was conducted according to the criteria listed below.

Table 2

Comparative analysis of deployment tools

Criteria	Range	Weight	Puppet	Chef	Ansible	SaltStack
Level of development	1.3	1	1	2	3	3
Ease of use	1.3	0.7	2	2	3	3
Security	1.3	1	3	1	3	2
Main language	1.3	0.5	1	1	3	3
Documentation	1.3	1	3	2	3	1
Flexibility	1.3	0.5	3	3	2	3
Total			10.4	8.4	13.6	11.1

1. The “Level of development” criterion reflects how much effort a specialist of an average competence needs to master the tool. This criterion directly affects the labor input required for its implementation.

2. The criterion of “Ease of use” shows how intuitive the tool is for a specialist. It also affects the level of mastery and comfort of using the tool by a specialist.

3. The “Security” criterion assesses the tool's reliability and safety in deployment, whether it contains vulnerabilities that could allow the leakage of company data or the introduction of malware into the corporate network. This is one of the most important criteria, since the security of user data is a top priority for companies.

4. The criterion of “Main language” indicates the threshold for entry and the ease of mastering the tool, which is determined by the programming language it utilizes.

5. The “Documentation” criterion reflects how fully the system is equipped with the necessary documentation for using the tool. The completeness of the documentation also affects the speed of mastering the tool by a specialist.

6. The criterion of “Flexibility” reflects how flexible the tool is to use for different tasks and on different infrastructures. This criterion determines the breadth of use of the tool.

As with comparison of office packages, each criterion has a score and a weight. Similarly, the point score is a range of values. The weight indicates the importance of the criterion within the framework of the studied task of implementing an office suite.

Each instrument is assigned a total score, which is formed according to the following formula:

$$s = k_1 \times p_1 + k_2 \times p_2 + \ldots + k_n \times p_n,$$

where k_n is the score according to the criterion n, p_n is the weight by criterion n.

To make a choice of an IT infrastructure preparation tool, a bar chart has been compiled reflecting the results of the comparative analysis (Fig. 2).

The tools considered are useful and applicable in various tasks. In this article, Ansible tool is chosen since it shows the best results according to most of the criteria presented.

Proposed solution concept

This paper proposes an approach for automated deployment of information infrastructure in data centers, as well as automated installation of the “R7-office” office package.

The infrastructure preparation and the software deployment using the developed software tool, implementing the proposed approach, takes place according to the stages shown in Fig. 3.

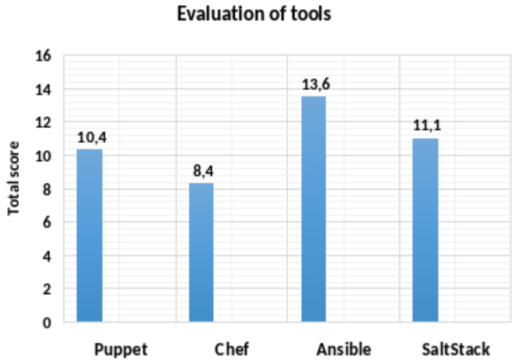


Fig. 2. Evaluation of tools

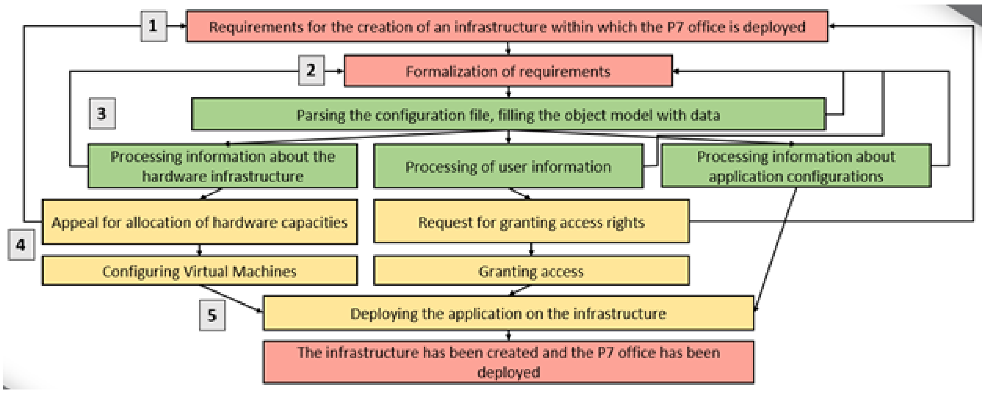

1
Requirements for the creation of an infrastructure within which the P7 office is deployed
2
Formalization of requirements
3
Parsing the configuration file, filling the object model with data
Processing information about the hardware infrastructure
Processing of user information
Processing information about application configurations
4
Appeal for allocation of hardware capacities
Request for granting access rights
Configuring Virtual Machines
Granting access
5
Deploying the application on the infrastructure
The infrastructure has been created and the P7 office has been deployed

Fig. 3. Scheme of IT infrastructure preparation and software deployment

The stages are as follows:

1. Collecting requirements for the software, the infrastructure for its deployment, as well as a list of users with their roles for access granting.

2. Formalizing requirements into the established format using the instructions. The output is a generated CSV-file containing structured, formalized infrastructure requirements and a list of system users with their roles for access granting. The requirements are formalized in a specific way to ensure the correct operation of the software.

3. Using Python scripts to parse the requirements file and generate configuration files for setup:

– YAML files for hardware configuration.

– A Bash-script for deployment application.

– Data for creating requests for the allocation of capacities and granting system access to users.

The described scripts are implemented within a software tool developed in Python, which aims to automatically parse requirements and generate the necessary output files for further stages.

4. Preparing the cluster and using Ansible [26–28] to configure the servers in accordance with the collected requirements, granting access to users. This includes configuring the OTRS system [31] to process requests for granting access to the office suite for employees and allocating capacity.

5. The final deployment of the application, which results in the infrastructure for the information system.

Implementation

1. Collecting and defining requirements for creating an infrastructure for R7-office deployment

At the initial stage, the system deployment requirements are identified. The minimum required set of requirements is divided into the following categories:

– Hardware infrastructure requirements:

– required infrastructure;

– technical requirements in a formalized form;

– a list of necessary access rights for employees.

2. Formalization of requirements

Using the instructions in CSV format (Fig. 4), the requirements are recorded in a single CSV file in a formalized form. To fill in the configuration file correctly, an instruction has been developed, also in CSV format. The file contains general information about the project with which the system is being implemented.

The structure of the CSV-file is as follows:

– General information about the project is specified at the top of the file: a code for quick search of the project in various information systems of the company, the project name, its purpose and objectives.

– This is followed by information about the system users for whom access needs to be requested, under the keyword "Users". The information is filled in without specific appearance and formatting requirements, except for the column-based organization of information.

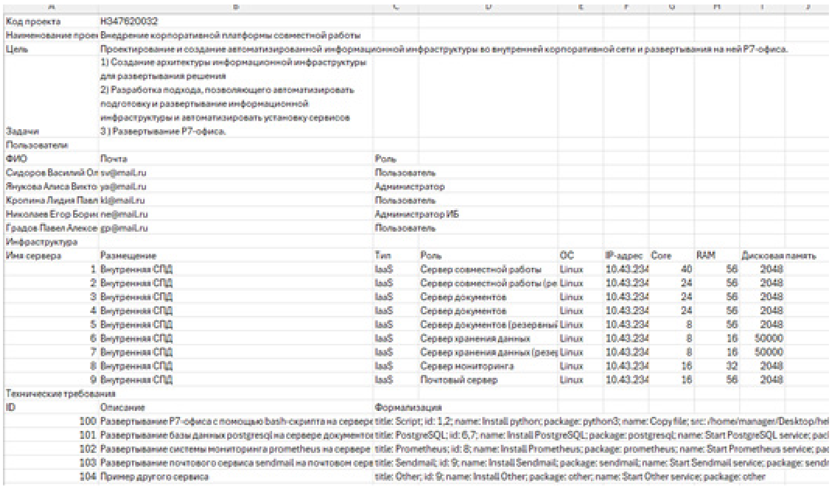
– Next, the hardware infrastructure requirements are listed under the keyword "Infrastructure". The information is also filled in without specific appearance and formatting requirements, except for the column-based organization of information.

– After that, the technical requirements for the infrastructure are described under the keyword "Technical requirements". The description is filled in freely, with formalization following specific rules to enable automatic conversion of requirements into configuration files.

To enable correct formalization of infrastructure requirements and their entry into a CSV-file, an instruction has been developed in CSV format. It describes two blocks.

The first block contains:

– a description of the types of YAML files [32] that the application supports, a description of the resulting functionality and a list of supported keywords;



A	B	C	D	E	F	G	H	I	J
Код проекта	Н347620032								
Наименование проек	Внедрение корпоративной платформы совместной работы								
Цель	Проектирование и создание автоматизированной информационной инфраструктуры во внутренней корпоративной сети и развертывания на ней Р7-офиса.								
Задачи	1) Создание архитектуры информационной инфраструктуры для развертывания решения 2) Разработка подхода, позволяющего автоматизировать подготовку и развертывание информационной инфраструктуры и автоматизировать установку сервисов 3) Развертывание Р7-офиса.								
Пользователи									
ФИО	Почта	Роль							
Сидоров Василий Ол	sv@mail.ru	Пользователь							
Янукова Алиса Викто	ya@mail.ru	Администратор							
Кропина Лидия Павл	kl@mail.ru	Пользователь							
Николаев Егор Борис	ne@mail.ru	Администратор ИБ							
Градов Павел Алексе	gp@mail.ru	Пользователь							
Инфраструктура									
Имя сервера	Размещение	Тип	Роль	ОС	IP-адрес	Core	RAM	Дисковая память	
1	Внутренняя СПД	IaaS	Сервер совместной работы	Linux	10.43.234	40	56	2048	
2	Внутренняя СПД	IaaS	Сервер совместной работы (ре	Linux	10.43.234	24	56	2048	
3	Внутренняя СПД	IaaS	Сервер документов	Linux	10.43.234	24	56	2048	
4	Внутренняя СПД	IaaS	Сервер документов	Linux	10.43.234	24	56	2048	
5	Внутренняя СПД	IaaS	Сервер документов (резервный	Linux	10.43.234	8	56	2048	
6	Внутренняя СПД	IaaS	Сервер хранения данных	Linux	10.43.234	8	16	50000	
7	Внутренняя СПД	IaaS	Сервер хранения данных (резе	Linux	10.43.234	8	16	50000	
8	Внутренняя СПД	IaaS	Сервер мониторинга	Linux	10.43.234	16	32	2048	
9	Внутренняя СПД	IaaS	Почтовый сервер	Linux	10.43.234	16	56	2048	
Технические требования									
ID	Описание	Формализация							
100	Развертывание Р7-офиса с помощью bash-скрипта на сервере	title: Script; id: 1,2; name: Install python; package: python3; name: Copy file; src: /home/manager/Desktop/hel							
101	Развертывание базы данных postgresql на сервере документов	title: PostgreSQL; id: 6,7; name: Install PostgreSQL; package: postgresql; name: Start PostgreSQL service; pacl							
102	Развертывание системы мониторинга prometheus на сервере	title: Prometheus; id: 8; name: Install Prometheus; package: prometheus; name: Start Prometheus service; pac							
103	Развертывание почтового сервиса sendmail на почтовом серв	title: Sendmail; id: 9; name: Install Sendmail; package: sendmail; name: Start Sendmail service; package: sendr							
104	Пример другого сервиса	title: Other; id: 9; name: Install Other; package: other; name: Start Other service; package: other							

Fig. 4. File containing the requirements

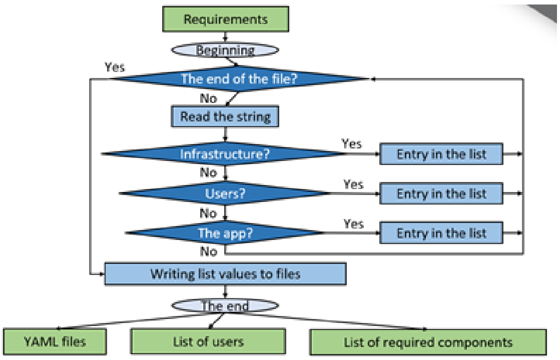


Fig. 5. Requirement analysis algorithm

– an algorithm of actions for cases when the file types implemented in the software are insufficient. In this case, the specialist can integrate support for a new type of service into the existing code.

The second block contains:

– requirements for the formalization of the functionality description;

– requirements for the formation of the source data file.

The instruction file is for informational purposes only for the employee who defines the requirements and is not used by the software. However, incorrect formalization of requirements makes proper use of the software solution impossible.

3. Parsing the configuration file, filling the object model with data

Automatically, using a Python software tool, the file is parsed into three components according to the requirement categories. Next, each part of the configuration file is processed.

The algorithm is shown in Fig. 5.

Description:

– CSV-file text is read line by line until the end of the file is reached;

– requirement category is identified using keywords in the configuration file;

– upon completion of the file analysis, three lists are formed with data corresponding to each category;

– data from each list is processed according to its own algorithm to generate output files;

– received output files are used to create OTRS requests for resource allocation and user access provisioning, as well as for configuring the infrastructure.

The processing of the CSV-file with requirements begins with a general analysis that separates it into three components. This is accomplished using a single library – "csv".

The file is read line by line. When one of the keywords is encountered: "Users", "Infrastructure" or "Technical requirements", – the relevant information is added to a dedicated list: user_list[], infra_list[] or tech_list[] using standard list manipulation methods.

Next, the data related to the list of users and the infrastructure are saved to separate "Users.csv" and "Infra.csv" files for further processing. Data related to technical requirements is passed to the next block of code for further processing and generation of Ansible playbooks.

4. Processing information about hardware infrastructure and users

Algorithm:

– data from the list is converted into text for an e-mail;

– e-mails are automatically sent to the OTRS system;

– the contractor processes and fulfill the requests;

– the request initiator tracks the status of registered requests, using a Python script, within the developed software.

The libraries “smtplib”, “csv”, “email” and “os.path” are used for implementation. The “Users.csv” file is parsed by the software, and an e-mail is generated requesting system access for users. The “Infra.csv” file is not converted in any way, but is sent to the OTRS system as an attachment. Thus, two e-mails with the necessary requests are sent to the system’s e-mail address for request registration.

In response to the created request, an e-mail with the number of the registered requests is sent to the sender. The developed software tool then enables request status tracking. To implement this feature, the libraries “datetime” and “pyotrs” are used. The user enters the tracking numbers into a text file and, after testing the program, receives a file with the status of the specified requests.

An OTRS system is configured to support this software solution. It is an open system for processing requests, which can be customized according to specific requirements. The system is prepared as follows:

– OTRS installation on a virtual machine with initial configuration;

– e-mail integration for sending and receiving system messages, including protocol configuration;

– creation of system agents and customers, as well as a request queue;

– configuration of request generation from e-mails;

– setup of automatic user notifications about the creation of a request.

5. Processing information about application configurations

Algorithm:

– data from the corresponding list is parsed by keywords and converted into several YAML files, serving as playbooks for Ansible.

– using the generated files and Ansible, virtual machines are configured and the office suite is deployed.

From the previously obtained tech_list[], the playbooks for virtual machines are generated.

The initial function decomposes the list into its components. The requirement ID is passed to a function that generates shared files for configuring Ansible: hosts and *_servers, where the * symbol indicates the name of the deployed service. In it, according to the received ID and using the infra_ list[], the IP address of the server is allocated and specified in the generated files. The requirements are also separated by the header keywords corresponding to the services being installed and transferred to functions that process each service separately. If a header contains a word unrelated to the keywords, a general YAML file is created for later specialist customization.

After a general analysis of the requirements, the YAML file is prepared for a specific service. Each supported service has a dedicated function. The structure of each such function is similar but differs in the specifics of configuring a particular service. The requirements are processed and converted into the YAML format of the playbook. For supported services like the PostgreSQL database, a .sql file is also generated to automatically add user information to the database.

6. Configuring virtual machines using Ansible

The configuration of the infrastructure using Ansible is as follows:

– initial configuration of virtual machines is performed: node manager and worker nodes, preparing them for Ansible operation;

– playbooks obtained by parsing the configuration file are transferred to a virtual machine that plays the role of node manager, and are launched sequentially, configuring the worker nodes and deploying the R7-office system.

To set up a virtual machine and deploy a R7-office, the following playbooks are created and used:

– a playbook for configuring the PostgreSQL database, which stores information about users and their roles for access control;

– a playbook for setting up the Prometheus monitoring server;

– a playbook for configuring the Sendmail mail server, enabling office-mail system integration;

– a playbook for launching a bash script, that handles the R7-office deployment program.

Future developments of the software solution may include additional playbooks for configuring supplementary services.

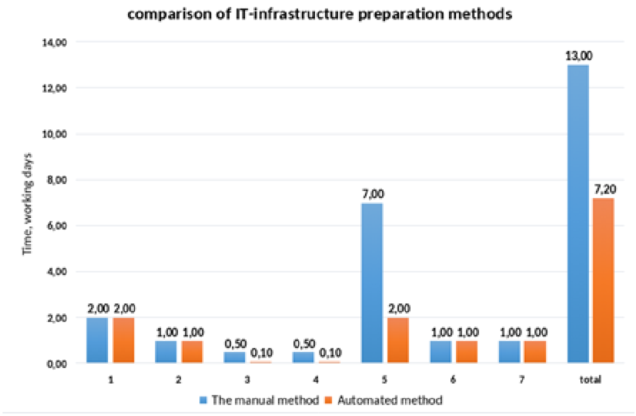


Fig. 6. Comparison of IT infrastructure preparation approaches

Experimental studies

Based on operational experience, an approximate time estimate has been made for the manual approach to parsing requirements for infrastructure preparation. This approach requires specialists to manually create playbooks for Ansible and submit requests for resource allocation and user access rights provisioning [4]. On average, for this method, with the number of servers up to 5, the number of users up to 100 and the number of deployed services up to 5, an average specialist needs up to two weeks to complete these tasks.

The calculation of the time taken by the proposed automated approach is carried out under constrained laboratory conditions using two nodes. According to the results, the full cycle of automated preparation of the IT infrastructure takes approximately 7 days. The automation of requirements analysis and configuration file generation reduced specialist workload by a factor of 5. The calculation of the values is shown in Table 3. Yellow indicates the phases that are automated using the proposed approach.

Table 3

Calculation of the time required to prepare the IT infrastructure in working days

Phase	A manual approach, (days)	An automated approach, (days)
Collecting requirements	2	2
Writing to a file	1	1
File analysis, formation of documents with parts of the requirements	0.5	0.1
Sending requests for access rights	0.5	0.1
Creating playbooks for setting up the infrastructure	7	2
Configuring Ansible on virtual machines	1	1
Launching playbooks	1	1
Total	13	7.2

A visual comparison of the time spent is shown on the graph (Fig. 6). It shows the number of points for each stage and in total.

The resulting temporary advantage is calculated using the following formula:

$$t = \left(t_1 - t_2\right) \div t_1 \times 100,$$

where t_1 is the initial time for the implementation of the manual approach, t_2 is the time obtained due to automation.

Thus, the average software deployment time during the implementation of the proposed approach is reduced by 45%.

Conclusion

The research conducted in the work and the comparative analysis presented showed that the most suitable package for solving the problem of reducing the company's resources by implementing a Russian corporate collaboration platform is the R7-office package, and the most successful tool for automating deployment according to a set of criteria is the Ansible tool.

The approach proposed in this work allows for automation of the preparation and configuration of the IT infrastructure, as well as the installation of the "R7-office" office package. The approach has the following distinctive features: automatic generation of configuration files for infrastructure configuration, files for requesting capacities and for providing access to users, as well as remote configuration of worker nodes and software deployment. The approach is implemented in a software tool using Ansible and Python technologies.

Experimental studies conducted in a local environment consisting of two nodes allow us to conclude that the time spent on implementation using the proposed solution can be reduced by 45%.

REFERENCES

1. **Gorkavchenko A.A., Zalukaeva E.A., Popenya I.V., Terekhov Ya.A.** Vozmozhnosti onlajn ofisov [Online Office Capabilities]. *Formirovanie i razvitie novoj paradigmy nauki v usloviyah postindustrial'nogo obshchestva: Sbornik statej po itogam Mezhdunarodnoj nauchno-prakticheskoj konferencii* [*Formation and development of a new paradigm of science in the conditions of post-industrial society: Collection of articles based on the results of the International scientific and practical conference*], 2023, Pp. 91–94.

2. **Mori G., Buzzi M.C., Buzzi M., Leporini B., Penichet V.M.R.** Collaborative editing for all: the google docs example. *Universal Access in Human-Computer Interaction. Applications and Services* (*UAHCI 2011*), 2011, Vol. 6768, Pp. 165–174. DOI: 10.1007/978-3-642-21657-2_18

3. **Kirvas V.A.** Sovremennye oblachnye informacionnye tekhnologii sovmestnoj raboty [Modern cloud information technologies for collaboration]. *Ekspertnye ocenki elementov uchebnogo processa: programma i materialy XIX mezhvuzovskoj nauchno-prakticheskoj konferencii* [*Expert assessments of elements of the educational process: program and materials of the XIX interuniversity scientific and practical conference*], 2017, Vol. 25, Pp. 35–43.

4. **Gorodnyaya L.V.** An approach to the measure of the labour required for programming. *Scientific Services & Internet*. 2020. Vol. 22, Pp. 192–209. DOI: 10.20948/abrau-2020-3

5. **Nikonenko A.N.** Methods and criteria for selection of suppliers. *World science: problems and innovations*, 2016, Vol. 2, Pp. 60–62.

6. **Sakoyan A.R., Puchkov V.I.** Challenges of the IT industry in the context of import substitution in 2024. *Lomonosovskie chteniya. Aktual'nye voprosy fundamental'nyh i prikladnyh issledovanij* [*Expert assessments of elements of the educational process: program and materials of the XIX interuniversity scientific and practical conference*], 2024, Pp. 16–20.

7. **Brooks F.P., Jr.** *The Mythical Man-Month* (*Essays on Software Engineering*). Reading: Addison-Wesley Publishing, 1975.

8. **Basynya E.A., Lukina M.S.** Avtomatizirovannaya ustanovka i konfigurirovanie servernyh reshenij [Automated installation and configuration of server solutions]. *Sovremennye materialy, tekhnika i tekhnologii* [*Modern materials, equipment and technologies*], 2016, Vol. 2, No. 5, Pp. 21–26.

9. **Ivlev V.A., Nikiforov I.V.** Aktual'nost' avtomatizirovannoj nastrojki infrastruktury IT-proekta [The relevance of automated configuration of IT project infrastructure]. *Sovremennye tekhnologii v teorii i praktike programmirovaniya* [*Modern technologies in the theory and practice of programming*], 2022, Pp. 79–80.

10. **Ivlev V.A., Mironenkov G.V., Nikiforov I.V., Kovalev A.D.** Generaciya informacionno-tekhnologicheskoj infrastruktury proekta na osnove neformalizovannyh trebovanij [Generation of the project's information technology infrastructure based on informal requirements]. *Sovremennye tekhnologii v teorii i praktike programmirovaniya* [*Modern technologies in the theory and practice of programming*], 2023, Pp. 242–244.

11. **Ivlev V.A., Nikiforov I.V., Yusupova O.A.** Automation method for configuring IT infrastructure for IT projects. *Proceedings of International Conference on Digital Transformation: Informatics, Economics, and Education* (*DTIEE2023*), 2023, Vol. 12637, Art. no. 126370D. DOI: 10.1117/12.2680779

12. **Phan T., Li W.-S.** Automated Configuration of System Infrastructure for SOA-Based Enterprise Computing. *2008 IEEE International Conference on e-Business Engineering*, 2008, Pp. 205–212. DOI: 10.1109/ICEBE.2008.97

13. **Putra W.R.A., Nurwa A.R.A., Priambodo D.F., Hasbi M.** Infrastructure as code for security automation and network infrastructure monitoring. *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, 2022, Vol. 22, No. 1, Pp. 201–214. DOI: 10.30812/matrik.v22i1.2471

14. **Panasenko S.V., Panasenko V.A.** Criteria for vendor-enterprises in the market of corporate search. *Newsletter of North-Caucasus Federal University*, 2015, Vol. 47, No. 2, Pp. 144–150.

15. **Kuznecova A.P.** Rossijskie oblachnye servisy dlya biznesa ot kompanij VK i "Yandeks" [Russian cloud services for business from VK and Yandex.]. *Social'no-ekonomicheskaya transformaciya v sovremennom mire: regional'nye proyavleniya* [*Socio-economic transformation in the modern world: regional manifestations*], 2022, Pp. 72–77.

16. **Kudryashov A., Nikitina E.** Comparative analysis of Yandex Office Packages.Documents, OpenOffice, WPS Office. *Nauchnoe soobshchestvo studentov XXI stoletiya, tekhnicheskie nauki* [*Scientific community of students of the 21st century, technical sciences.*], 2023, Pp. 52–59.

17. **Wilson K.** Microsoft office 365. In: *Using Office 365*, 2014, Pp. 1–14. DOI: 10.1007/978-1-4302-6686-0_1

18. **Habraken J.** *Microsoft Office Inside Out* (*Office 2021 and Microsoft 365*). Redmond (Washington): Microsoft Press, 2021.

19. **Mutovina N.V., Sahanov R.R.** Upravlenie IT proektami v Notion [IT Project Management in Notion]. *Proektnyj menedzhment v Kazahstane: sostoyanie, problemy i perspektivy* [*Project management in Kazakhstan: status, problems and prospects*]. 2021.

20. **Burgonov O.V., Konstantinova O.I., Platonov K.A.** Tools of digital transformation of research and educational organizations. *Journal of Legal and Economic Studies*, 2022, Vol. 4, Pp. 263–270. DOI: 10.26163/GIEF.2022.68.50.041

21. **Sharatova N.V.** Perekhod na R7-ofis v ramkah importozameshcheniya. Sravnitel'naya harakteristika s Microsoft Office [Transition to P7-office as part of import substitution. Comparative characteristics with Microsoft Office]. *Morskie tekhnologii: problemy i resheniya – 2023* [*Marine Technologies: Problems and Solutions – 2023*], 2023, Pp. 458–462.

22. **Zaslavskij A.A.** R7 ofis kak otechestvennaya al'ternativa paketu Microsoft Office [P7 office as a domestic alternative to the Microsoft Office package]. *Interaktivnoe obrazovanie* [*Interactive education*], 2022, Vol. 2, Pp. 25–27.

23. **Loope J.** *Managing Infrastructure with Puppet: Configuration Management at Scale*. Sebastopol (California): O'Reilly Media, 2011.

24. **Rhett J.** *Learning Puppet 4: A Guide to Configuration Management and Automation*. Sebastopol (California): O'Reilly Media, 2016.

25. **Yaremchuk S.** Upravlenie konfiguraciej s Chef [Configuration management with Chef]. *Sistemnyj administrator* [*System administrato*r], 2010, Vol. 7–8, Pp. 28–33.

26. **Medvedev O.S.** Nastrojka Linux-serverov s pomoshch'yu centralizovannogo upravleniya Ansible [Setting up Linux servers with centralized Ansible management]. *54-ya nauchnaya konferenciya aspirantov,*

magistrantov i studentov BGUIR [*54th scientific conference of postgraduate, master's and undergraduate students of BSUIR*], 2018, P. 335.

27. **Hochstein L.** *Ansible: Up and Running: Automating Configuration Management and Deployment the Easy Way*. Sebastopol (California): O'Reilly Media, 2017.

28. **Vorontsov Y.A., Mikhailova E.K.** Combined architecture of the configuration management system for computing infrastructure. *International Journal of Open Information Technologies*, 2021, Vol. 9, No. 11, Pp. 17–21.

29. **Anthony S.** *SaltStack for Configuration Management*. Bethlehem (Pennsylvania): Lehigh University, 2018.

30. **Hosmer B.** Getting started with Salt Stack – the other configuration management system built with Python. *Linux journal*, 2013. Available: https://www.linuxjournal.com/content/getting-started-salt-stack-other-configuration-management-system-built-python (Accessed 01.09.2025)

31. **Shabaev T.A.** *Avtomatizaciya processa obsluzhivaniya klientov s pomoshch'yu sistemy OTRS Free* [*Automate your customer service process with OTRS Free*], Bachelor's Thesis, Tomsk Polytechnic University, 2016.

32. **Eriksson M., Hallberg V.** *Comparison between JSON and YAML for data serialization*, Bachelor's Thesis, Royal Institute of Technology, 2011.

INFORMATION ABOUT AUTHORS / СВЕДЕНИЯ ОБ АВТОРАХ

Aleksandra A. Kiseleva
Киселева Александра Александровна
E-mail: aleksandrakiseleva2001@gmail.com
ORCID: https://orcid.org/0009-0004-8263-3289

Igor V. Nikiforov
Никифоров Игорь Валерьевич
E-mail: igor.nikiforovv@gmail.com
ORCID: https://orcid.org/0000-0003-0198-1886

Maxim V. Shishko
Шишко Максим Вадимович
E-mail: shishko.mv.post@gmail.com
ORCID: https://orcid.org/0009-0005-3406-7005

Sergey M. Ustinov
Устинов Сергей Михайлович
E-mail: usm50@yandex.ru
ORCID: https://orcid.org/0000-0003-4088-4798

Submitted: 03.12.2024; Approved: 26.06.2025; Accepted: 17.07.2025.
Поступила: 03.12.2024; Одобрена: 26.06.2025; Принята: 17.07.2025.