УДК 004.07
DOI https://doi.org/10.32838/2663-5941/2022.1/18

Киричек Г.Г.
Національний університет «Запорізька політехніка»

Щетінін М.О.
Національний університет «Запорізька політехніка»

УПРАВЛІННЯ КОНФІГУРАЦІЄЮ СЕРВЕРІВ НА ОСНОВІ ANSIBLE

У роботі запропоновано модель системи управління конфігурацією серверів і метод її програмної реалізації. У ході реалізації системи вирішено такі завдання: досліджені та порівняні наявні системи управління конфігурацією серверів; проведено аналіз і вибір технологій для реалізації системи управління; змодельована система управління; розгорнутий і налаштований майстер-сервер; на майстер-сервері конфігуровано систему управління та проведена її інтеграція з графічним інтерфейсом; створені сценарії для розгортання серверів із різними операційними системами й конфігураціями; проведено тестування реалізованої системи. Система включає графічний інтерфейс клієнта, сервер управління Ansible та клієнтські сервери. Для управління системою використовується графічний інтерфейс із відкритим вихідним кодом AWX, який взаємодіє із сервером управління й користувачем. Як хостинг-платформу для розгортання проєкту обрано сучасну хмарну хостинг-платформу Google Cloud. Сервер управління Ansible містить список клієнтських серверів і їх груп, сценарії, написані з використання мов Python та YAML, убудовані та сторонні модулі, написані мовою Python і плагіни. Графічний інтерфейс AWX як базу даних використовує PostgreSQL. Для виконання сценаріїв сервер управління конфігурацією підключається до клієнтських серверів за допомогою протоколу SSH. Метою роботи є проведення досліджень, вибір методів конфігурування серверів на основі різних операційних систем і реалізація системи управління конфігурацією серверів з метою вдосконалення процесу їх налаштування й конфігурації. Об'єктом дослідження є процес реалізації системи управління конфігурацією серверів. Предметом дослідження є моделі, методи, алгоритми та програмні засоби реалізації цієї системи. Ручне розгортання серверів займає вдвічі більше часу при розгортанні з попередньо встановленими вебсервером, сервером баз даних і службою PHP-FPM і навіть у чотири рази більше часу при розгортанні серверів із базовою конфігурацією. Одержаний результат можна використовувати для розгортання серверів із різними операційними системами та конфігураціями.

Ключові слова: *система управління конфігурацією серверів, Ansible, CentOS, Ubuntu, YAML, Python, SSH.*

Постановка проблеми. Розвиток вебтехнологій протягом останніх двох десятиліть і світовий карантин 2020–2021 рр. призвели до значної популяризації вебхостингу. Тому велика кількість приватних осіб та особливо підприємців зацікавилася можливістю займатися комерційною діяльністю в мережі Інтернет [1]. Отже, подібний сплеск популярності через значні затрати часу робить розгортання нових серверів вручну дуже неефективним. Виходячи із цього, більшість великих, середніх і навіть малих хостинг-компаній переходять або вже перейшли на розгортання серверів за допомогою систем управління конфігурацією [2; 3]. Ці системи автоматизують дії з налаштування та спрощують складні операції, які постійно повторюються й необхідно виконувати при черговому розгортанні серверу [4]. Це дає змогу значно зекономити час, коли необхідно налаштовувати велику кількість серверів або вносити зміни до їх конфігурацій.

Аналіз останніх досліджень і публікацій. У роботі розглянуто 4 найбільш популярні системи управління конфігурацією серверів (далі – СУКС): Ansible, Puppet, Chef і Salt [5]. У таблиці 1 наведено порівняння найважливіших характеристик розглянутих СУКС. Наведені дані показують, що Puppet [6] і Chef [2] є доволі складними СУКС, тому що обидві системи мають клієнт-серверну архітектуру та вимагають наявності баз даних.

Окрім цього, як Puppet, так і Chef, на відміну від Ansible [7] і Salt [3], підтримують роботу виключно в середовищі Linux, що робить їх менш універсальними системами. Системи Ansible та Salt доволі схожі, але Salt не має офіційного графічного інтерфейсу користувача та не виконує операції покроково, а це призведе до більш складного налаштування в разі виникнення проблем [8]. Вимоги, які висуваються до реалізованої системи, мають відповідати певним критеріям і мати майстер-сервер із конфі-

Таблиця 1

Порівняння систем управління конфігурацією серверів

	Ansible	**Puppet**	**Chef**	**Salt**
Мова програмування	Python	C++	Ruby	Python
Мова конфігурування	YAML	Puppet DSL	Ruby	YAML
База даних	-	PuppetDB	PostgreSQL	-
Метод обміну даними	SSH	Mcollective	RabbitMQ	ZeroMQ
Підтримувані ОС	FreeBSD, Linux, macOS, Solaris	Linux	Linux	FreeBSD, Linux
Наявність агента	-	+	+	+ -
Архітектура	Серверна	Клієнт-серверна	Клієнт-серверна	Серверна
Метод розгортання	Push	Pull	Pull	Push
Покрокове виконання	+	-	+	-
Наявність офіційного графічного інтерфейсу	+	+	+	-

гурованою СУКС; наявність графічного інтерфейсу користувача; можливість розгортання серверів із використанням різних операційних систем і конфігурацій, обраних користувачем із певного переліку.

Постановка завдання. Спираючись на мету роботи й вимоги до системи, визначимо завдання, які необхідно виконати в ході роботи: дослідити та порівняти наявні СУКС; провести аналіз і вибір технологій для реалізації системи управління; змоделювати систему; розгорнути й налаштувати майстер-сервер; конфігурувати СУКС на майстер-сервері та провести інтеграцію з графічним інтерфейсом; створити сценарії для розгортання серверів із різними операційними системами й конфігураціями; провести тестування реалізованої системи та проаналізувати результати.

Система складається з трьох основних частин: графічний інтерфейс користувача; сервер управління Ansible та клієнтські сервери [5]. Для управління системою використовується графічний інтерфейс з відкритим вихідним кодом AWX, який взаємодіє із сервером управління й користувачем. Сервер управління Ansible містить список клієнтських серверів і їх груп, сценарії, написані з використання мов Python [9] і YAML [10], вбудовані та сторонні модулі, написані мовою Python, а також різноманітні плагіни. Для виконання сценаріїв сервер управління підключається до клієнтських серверів за допомогою протоколу SSH [11].

Виклад основного матеріалу. Метою роботи є проведення досліджень, вибір методів конфігурування серверів на основі різних операційних систем і реалізація системи управління конфігурацією серверів з метою вдосконалення процесу їх налаштування й конфігурації. Об'єктом дослідження є процес реалізації системи управління

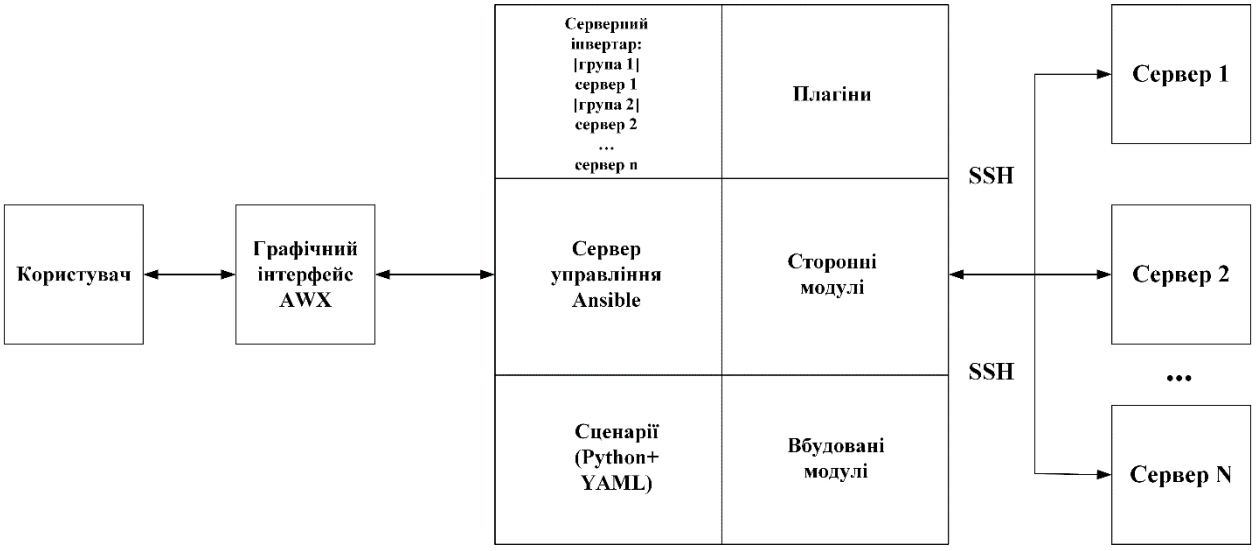

Рис. 1. Загальна модель системи

конфігурацією серверів. Предметом дослідження є моделі, методи, алгоритми та програмні засоби реалізації цієї системи.

Загальна модель системи (рис. 1) включає такі модулі й етапи функціонування: HTTP-запити від користувача, надіслані за допомогою веббраузера, отримує графічний інтерфейс AWX; він обробляє їх і пересилає для виконання серверу управління Ansible; сервер запускає необхідні сценарії на вказаній групі серверів за допомогою протоколу SSH і повертає результат графічному інтерфейсу AWX, який після обробки результату повертає користувачу HTTP-відповіді.

Використання ОС CentOS Stream 8 [12] як основи для сервера Ansible та графічного інтерфейсу AWX вимагає розміщення певним чином файлів сценаріїв і файлів, які використовуються в цих сценаріях. Як хостинг-платформу для розгортання проєкту обрано сучасну хмарну хостинг-платформу Google Cloud [13]. Кореневим каталогом для проєктів при використанні AWX служить каталог /var/lib/awx/projects, у якому, як приклад, використовуємо каталог проєкту university. У кореневому каталозі проєкту маємо точки входу, які служать для завантаження груп сценаріїв у певному порядку. Каталоги груп сценаріїв розміщуються в каталозі roles кореневого каталогу проєкту. Групи сценаріїв включають каталог tasks із файлом main.yml усередині, який завжди є точкою входу в певну групу сценаріїв.

Додаткові файли сценаріїв у будь-якій кількості також розміщені в цьому каталозі й написані мовою Python [9] із використанням формату YAML [10]. Окрім цього, каталоги груп сценаріїв містять у каталогах files і templates, відповідно, файли, які не потребують модифікації, та шаблони, які перед використанням потребують певної модифікації. При цьому Ansible, як і деякі інші системи управління конфігурацією, для створення сценаріїв вимагає використовувати процедурний стиль програмування [14]. Перевагами цього стилю є повторне використання коду в різних місцях програми без необхідності його копіювання; використання концепції пам'яті як сховища, уміст якого може оновлюватися операторами програми, і послідовне виконання операторів для перетворення початкового стану пам'яті [15].

Далі частково наведемо файли, створені та розміщені відносно кореневого каталогу проєкту:

– Base.yml та LAMP.yml – точки входу для завантаження груп сценаріїв, призначених для розгортання серверів із базовою конфігурацією та попередньо встановлених вебсервером; сервером баз даних і службою PHP-FPM, відповідно;

– у roles/apache/files/: mod_deflate.conf і mod_expires.conf – файли конфігурації модуля deflate й expires для Apache; далі в roles/apache/tasks розташовані сценарій і точка входу встановлення й налаштування Apache;

– у roles/apache/templates/: apache2.conf – головний файл конфігурації Apache для ОС Ubuntu; hostname.conf – файл конфігурації головного імені сервера для вебсервера Apache; httpd.conf – головний файл конфігурації Apache для ОС CentOS; index.html – файл, що містить головне ім'я сервера й розташовується в кореневому каталозі Apache; ssl_centos.conf – файл конфігурації модуля ssl для Apache в ОС CentOS і ssl_ubuntu.conf – файл конфігурації модуля ssl для Apache в ОС Ubuntu [16];

– у roles/base/tasks/: centos.yml – сценарій базового налаштування серверів, які використовують ОС CentOS; main.yml – точка входу в групу сценаріїв базового налаштування серверів; ubuntu.yml – сценарій базового налаштування серверів, які використовують ОС Ubuntu;

– у roles/cleanup/: cleanup.yml – сценарій для очистки серверів від даних налаштування; main.tml – точка входу в групу цих сценаріїв;

– у roles/core/: files/local_bin.sh – файл конфігурації системної змінної; files/ysctl.conf – файл конфігурації змінних ядра Linux; tasks/core.yml – сценарій підготовки сервера до встановлення вебсервера, сервера баз даних і служби PHP-FPM; tasks/main.yml – точка входу в групу цих сценаріїв;

– у roles/mail/: files/master.cf – файл для перевизначення директив поштового сервера Postfix; tasks/mail.yml – сценарій установлення й налаштування Postfix; tasks/main.yml – точка входу в групу цих сценаріїв; templates/main_centos.cf – головний файл конфігурації Postfix для ОС CentOS; templates/main_ubuntu.cf – файл конфігурації Postfix для ОС Ubuntu;

– у roles/mariadb/: files/limits.conf – файл конфігурації обмежень для сервера баз даних MariaDB; /files/MariaDB.repo – файл конфігурації репозиторію для MariaDB; tasks/main.yml – точка входу у групу цих сценаріїв; tasks/mariadb.yml – сценарій установлення й налаштування MariaDB;

– у roles/php/: files/php.ini – головний файл конфігурації PHP; tasks/php.yml – сценарій установлення й налаштування PHP-FPM; tasks/main.yml – точка входу в групу цих сценаріїв;

– у roles/preflight/tasks/: preflight.yml – сценарій перевірки сервера перед запуском інших сценаріїв; main.yml – точка входу в групу сценаріїв, яка відповідає за перевірку сервера перед запуском інших сценаріїв;

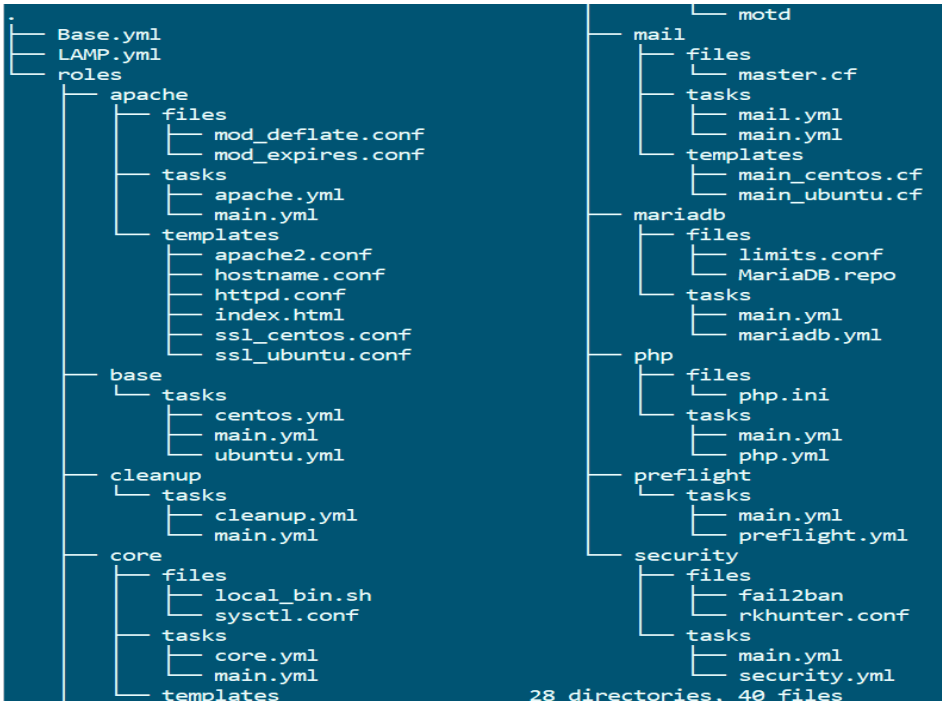

Рис. 2. Файлова структура проекту

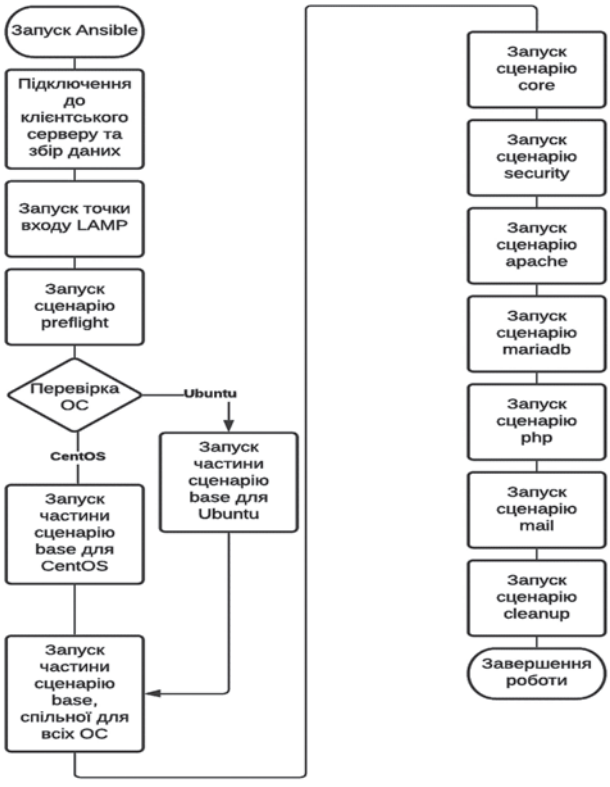

Рис. 3. Алгоритм роботи сценарію LAMP

– у roles/security/: files/fail2ban – файл конфігурації журналів служби Fail2Ban; files/rkhunter.conf – файл конфігурації застосунку rkhunter; tasks/security.yml – сценарій із налаштування системи безпеки; tasks/main.yml – точка входу в групу сценаріїв із налаштування системи безпеки.

Графічне зображення структури проєкту наведено на рисунку 2.

Алгоритм роботи LAMP із попередньо встановленими вебсервером; сервером баз даних і службою PHP-FPM наведено на рисунку 3.

Для запуску сценаріїв необхідно підключитися до серверу Ansible за протоколом SSH, перейти в каталог «/var/lib/awx/projects/university» за допомогою команди «cd/var/lib/awx/projects/university» й запустити обрану точку входу, використовуючи команду «ansible-playbook точка.yml», де «точка» є назвою точки входу. Серверний інвентар при використанні інтерфейсу Ansible завантажується з файлу «/etc/ansible/hosts».

Результати тестування швидкості розгортання серверів різної конфігурації за допомогою системи й порівняння зі швидкістю ручного розгортання наведені на рисунку 4.

З наведених результатів бачимо, що ручне розгортання серверів займає вдвічі більше часу при розгортанні з попередньо встановленими вебсервером, сервером баз даних і службою PHP-FPM, навіть у чотири рази більше часу при розгортанні серверів із базовою конфігурацією.

Далі наведемо порівняння швидкості розгортання серверів із різною конфігурацією за допомогою AWX та командного рядку Ansible (рис. 5).

Рис. 4. Результати тестування швидкості розгортання серверів

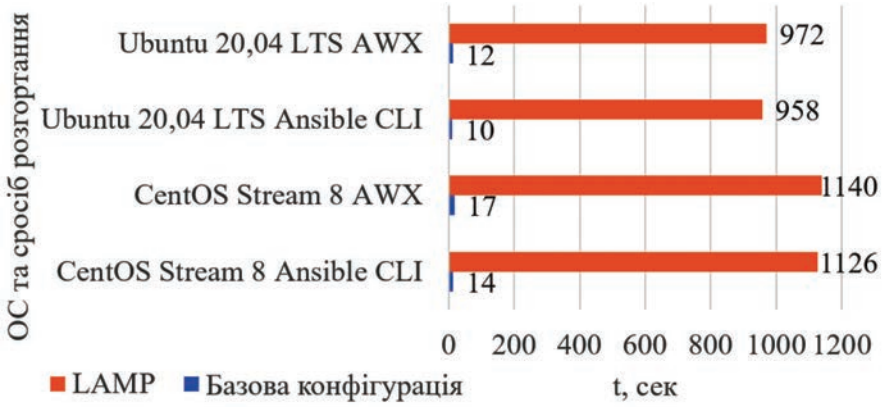

Рис. 5. Порівняння швидкодії AWX і командного рядка Ansible

З наведених результатів тестування можна побачити, що використання графічного інтерфейсу AWX дещо знижує швидкодію системи, але втрати швидкодії не є значними.

Висновки. У ході виконання роботи вирішено такі завдання: досліджено та порівняно наявні системи управління конфігурацією серверів; проведено аналіз технологій для реалізації задач управління конфігурацією серверів; змодельовано систему; розгорнуто та налаштовано майстер-сервер, на якому сконфігуровано систему управління конфігурацією та проведено інтеграцію з графічним інтерфейсом AWX; створено сценарії для розгортання серверів із різними операційними системами та конфігураціями; проведено тестування системи. З наведених результатів бачимо, що ручне розгортання серверів займає вдвічі більше часу при розгортанні з попередньо встановленими вебсервером, сервером баз даних і службою PHP-FPM, навіть у чотири рази більше часу при розгортанні серверів із базовою конфігурацією. Завдяки використанню Ansible реалізована система може використовуватися на будь-якому сервері з Unix-подібною ОС і здатна розгортати сервери на основі операційних систем CentOS Stream 8 і Ubuntu 20.04 LTS із конфігурацією, обраною користувачем.

Список літератури:

1. Рудьковський О.Р., Киричек Г.Г. Програмний комплекс з підтримки розподіленої взаємодії мережевих пристроїв та додатків. *Вчені записки ТНУ імені В.І. Вернадського. Серія «Технічні науки».* 2021. Вип. 32 (71). № 2. С. 229–234.

2. Marschall M. Chef Infrastructure Automation Cookbook. Second Edition. Birmingham : Packt Publishing, 2015. 258 p.

3. Sebenik C. Hatch, Salt T. Essentials: Getting Started with Automation at Scale. Sebastopol : O'Reilly Media Inc., 2015. 163 p.

4. Киричек Г.Г., Гаркуша В.Ю. Віртуалізація хостів на основі Proxmox VE в умовах надлишкового використання ресурсів. *Вчені записки ТНУ імені В.І. Вернадського. Серія «Технічні науки».* 2021. Вип. 32 (71). № 1. С. 78–84.

5. Киричек Г.Г., Щетінін М.О. Конфігурація серверів з використанням Ansible. *Modern scientific research: achievements, innovations and development prospects :* International scientific conference. Riga, 2021. P. 15–17.
6. Rhett J. Learning Puppet 4: A Guide to Configuration Management and Automation. Sebastopol : O'Reilly Media Inc., 2016. 550 p.
7. Geerling J. Ansible for DevOps. Victoria : Lenpub, 2015. 296 p.
8. Puppet vs. Chef vs. Ansible vs. SaltStack. URL: https:// jetpatch.com/blog/agent-management/puppet-vs-chef-vs-ansible-vs-saltstack.
9. Kirichek G., Kurai V. Implementation quadtree method for comparison of images. *14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering, TCSET 2018-Proceedings.* Lviv-Slavske, 2018. P. 129–132.
10. What is YAML? URL: https://blog.stackpath.com/yaml.
11. How does SSH work? URL: https://www.hostinger.com/ tutorials/ssh-tutorial-how-does-ssh-work.
12. About CentOS. URL: https://www.centos.org/about.
13. Google Cloud Platform. URL: http://wiseit.com.ua/cloud-services/google-cloud-platform.
14. Основні поняття програмування. URL: http://lib.mdpu.org.ua/e-book/sharov/lection/lec11.htm.
15. White G. Sivitanides M. Cognitive Differences Between Procedural Programming and Object Oriented Programming. San Marcos : Texas State University at San Marcos, 2005. 350 p.
16. About the Ubuntu project. URL: https://ubuntu.com/about.
17. Keating J. Mastering Ansible. Packt Publishing Ltd, 2015. 214 p.

Kirichek G.G., Shchetinin M.O. ANSIBLE-BASED SERVERS CONFIGURATION MANAGEMENT

The paper proposes a model of server configuration management system and a method of its software implementation. During the implementation of the system the following tasks were solved: researched and compared existing server configuration management systems; conducted analysis and selection of technologies for the implementation of the management system; simulated control system; deployed and configured master server; the control system is configured on the master server and its integration with the graphical interface is carried out; created scenarios for deploying servers with different operating systems and configurations and tested the implemented system. The system includes a graphical client interface, Ansible management server, and client servers. The system is managed by an open-source graphical user interface AWX, which interacts with the management server and the user. The modern cloud hosting platform Google Cloud was chosen as the hosting platform for the project deployment. Ansible management server contains a list of client servers and their groups, scripts written using Python and YAML, embedded and third-party modules written in Python, and plugins. The AWX GUI uses PostgreSQL as the database. And to run scripts, the configuration management server connects to client servers using SSH. The purpose of the work is to conduct research, select methods for configuring servers based on different operating systems, and implement a server configuration management system to improve the process of setting up and configuring them. The research object is the process of implementing a server configuration management system. The research subject is models, methods, algorithms, and software tools for implementing this system. Deploying servers manually takes twice as long to deploy with a pre-installed web server, database server, and PHP-FPM, and even four times as long to deploy servers with a basic configuration. The result can be used to deploy servers with different operating systems and configurations.

Key words: *servers configuration management system, Ansible, CentOS, Ubuntu, YAML, Python, SSH.*