Uniwersytet
im. Adama Mickiewicza
w Poznaniu

Wydział Matematyki i Informatyki

Marcin Woźniak

Numer albumu: 434812

Opis systemu Gentoo Linux ze szczególnym uwzględnieniem sposobów instalacji / kompilacji pakietów

Description of Gentoo Linux with particular emphasis on package installation / compilation

Praca magisterska na kierunku **informatyka**

napisana pod opieką

prof. Michała Hanćkowiaka

Poznań, czerwiec 2022

Spis treści

Streszczenie

Niniejsza praca dyplomowa opisuje system Gentoo Linux, razem z jego historią oraz dodatkowymi informacjami, które pomagają zrozumieć czym jest ta dystrybucja. W pracy została również opisany sposób w jaki można instalować pakiety, które w dalszej części zostały opisane oraz porównane.

Podczas pisania owej pracy udało mi się zautomatyzować proces instalacji systemu Gentoo Linux za pośrednictwem Ansible oraz autorskiego scenariusza, który wykonuje zadania (rozdział 7). Ansible to narzędzie udostępniania oprogramowania, zarządzania konfiguracją i wdrażania aplikacji, umożliwiające tworzenie infrastruktury jako kodu. Projekt na użytek pracy został kilkanaście razy uruchomiony ze względu na wiarygodność testów(rozdział 6).

Abstract

This diploma paper describes the Gentoo Linux system, along with its history and additional information to help you understand what the distro is. The work also describes how to install packages, which are described and compared in the following sections.

While writing this paper, I was able to automate the Gentoo Linux installation process via Ansible and my own scenario that performs the tasks (chapter 7). Ansible is a software sharing, configuration management, and application deployment tool that enables you to build infrastructure as code. The project for the purpose of work was launched several times due to the reliability of the tests (chapter 6).

Wstęp

Różne dystrybucje systemu operacyjnego Linux działają w różnoraki sposób. Same dystrybucje możemy podzielić ze względu na pochodzę (na jakiej głównej dystrybucji się opiera), sposób kompilacji, koszta.

W owej pracy została wybrana dystrybucja, która skupia się na instalowaniu pakietu ze źródła. Dystrybucja ta to Gentoo Linux. Skierowana jest ona do bardziej doświadczonych użytkowników z powodu jej trudnej instalacji. Stąd narodził się pomysł o napisaniu projektu, który instaluje w sposób automatyczny Gentoo Linux za pośrednictwem Ansible. Projekt ten znajduje się w rozdziale 7.

Rozdział 1

Wprowadzenie do systemu Gentoo Linux

1.1 Opis systemu Gentoo Linux

Gentoo Linux jest dystrybucją Linuxa opierającą się na kompilowaniu pakietów ze źródła. Ta dystrybucja wspiera takie architektury jak: alpha, amd64, arm, arm64, hppa, ia64, mips, ppc, ppc64, s390, sparc, oraz x86.

Rysunek 1: Logo Gentoo Linux

Źródło: `https://www.gentoo.org/inside-gentoo/artwork/gentoo-logo.html`

Aby zarządzać pakietami używamy `Portage` (definicja znajduje się 1.5.1)[1].

Sam kod źródłowy jest kompilowany lokalnie zgodnie z preferencjami użytkownika, które można ustawić. Temat ten będzie poświęcony w dalszej sekcji 1.5.2. Oczywiście w systemie Gentoo Linux istnieje możliwość instalacji binarnych, tylko dla niektórych większych pakietów. Dobrym przykładem jest `www-client/firefox-bin`.

Drzewo repozytorium Gentoo Linux podzielone jest na kategorię a następnie na pakiet. Przykładowy zrzut ekranu znajduje się poniżej na rysunku 2.

summary refs log tree commit diff log

Mode	Name	Size		
-rw-r--r--	.gitignore	79	log	plain
d---------	acct-group	15735	log	plain
d---------	acct-user	14529	log	plain
d---------	app-accessibility	996	log	plain
d---------	app-admin	7052	log	plain
d---------	app-antivirus	174	log	plain
d---------	app-arch	3287	log	plain
d---------	app-backup	1806	log	plain
d---------	app-benchmarks	929	log	plain
d---------	app-cdr	1652	log	plain
d---------	app-containers	1431	log	plain
d---------	app-crypt	5142	log	plain
d---------	app-dicts	23455	log	plain
d---------	app-doc	1564	log	plain
d---------	app-editors	2789	log	plain
d---------	app-emacs	11610	log	plain
d---------	app-emulation	3168	log	plain
d---------	app-eselect	1675	log	plain
d---------	app-forensics	990	log	plain
d---------	app-i18n	4199	log	plain
d---------	app-laptop	697	log	plain
d---------	app-metrics	1438	log	plain
d---------	app-misc	10182	log	plain
d---------	app-mobilephone	412	log	plain
d---------	app-office	1708	log	plain
d---------	app-officeext	322	log	plain
d---------	app-pda	467	log	plain

Rysunek 2: Drzewo repozytorium Gentoo Linux

Źródło: `https://gitweb.gentoo.org/repo/gentoo.git/tree/`

1.2 Historia systemu Gentoo Linux

Gentoo Linux został stworzony przez Daniela Robbinsa jako dystrybucja Enocha Linux. Celem było stworzenie dystrybucji bez prekompilowanych plików binarnych, która byłaby dostosowana do sprzętu i zawierała tylko wymagane programy. Nazwa koniec końców uległa zmianie na Gentoo (gatunek Gentoo to najszybciej pływający pingwin). Gentoo Linux 1.0 został wydany 31 marca 2002 roku.

1.3 Stage w Gentoo Linux

Podczas instalacji systemu Gentoo Linux można wybrać `Stage`, czyli archiwum podstawowych plików używanych do instalacji Gentoo Linux [3]. Podział tych archiwów jest następujący:

- Stage 1 - zawiera, co jest konieczne do zbudowania zestawu narzędzi (różne kompilatory, linkery i biblioteki językowe niezbędne do kompilowania innego oprogramowania) dla systemu docelowego; kompilowanie tego docelowego łańcucha narzędzi z innego, istniejącego wcześniej systemu hosta jest znane jako ładowanie systemu docelowego.[3]

- Stage 2 - zawiera, narzędzia do samo-hostingu systemu docelowego, który jest następnie używany do kompilacji całego innego podstawowego oprogramowania użytkownika dla systemu docelowego.[3]

- Stage 3 - zawiera, minimalny zestaw skonfigurowanych narzędzi aby przyspieszyć instalację systemu. W podręczniku instalacyjnym do wersji procesora amd64 oraz do każdej innej architektury ten stage jest zalecany.[3]

- Stage 4 - To archiwum zawiera jądro i program ładujący, dzięki czemu zapewniają system startowy. Nie jest jednak podejmowana żadna automatyczna detekcja sieci, więc nie są one przeznaczone do zastąpienia zwykłej metody instalacji opartej na podręczniku.[3]

1.4 Zalety systemu Gentoo Linux

- Personalizacja od zera.

  Nie mamy żadnych instalatorów GUI[1]. Sam system instalujemy od zera. Do tego możemy użyć jakiegokolwiek bootowalnego pendrive z system Linux posiadający dostęp do internetu. Kolejne kroki instalacji opisane są w podręczniku użytkownika[8].

- Wydajność

  System Gentoo Linux wspiera praktycznie każdą architekturę[7], daje nam to bardzo duże możliwości jeżeli chodzi o prędkość działania systemu jak i wydajność. Dodatkowo każda z zainstalowanych aplikacji, serwisów będzie bardziej wydajna na Gentoo Linux niż na innym systemie operacyjnym.

- Instalacja pakietów ze źródła.

  Menadżerem plików jest `Portage`, który napisany jest w języku Python. Sam sposób pobierania pakietów ze źródła opiera się na drzewie repozytorium, które jest synchronizowane za pomocą `rsync` albo `git` . Bezpośrednia kompilacja wybranego programy jest bezpieczniejsza, ponieważ nie używamy już wcześniej skompilowanej pliku binarnego, w którym nie wiem co się może znajdować. Same pakiety w repozytorium

[1] GUI - Graphical user interface - Graficzny interfejs

podzielone są na kategorie, a następnie na pakiety (`kategoria/pakiet`). Aby pobrać dowolny pakiet musimy skorzystać z narzędzia wiersza poleceń jakim jest `Emerge`.

- Możliwością instalowania tylko to co uważamy za potrzebne.
  Przy instalacji pakietu `Emerge`, oferuje nam użycie dowolnej oferowanej flagi (USE flags) przez pakiet. Same flagi musimy zdefiniować w pliku[2]. Przykładowe wywołanie `Portage` przy użyciu terminalowego narzędzia `Emerge`.

  Kod źródłowy 1.1: Przykładowe wywołanie instacji pakietu `htop` z kategorii `sys-process`. Źródło: Opracowanie własne

```
yorune@Gentoo ~ $ emerge --ask sys-process/htop

These are the packages that would be merged, in order:

Calculating dependencies... done!
[ebuild N ] sys-process/htop-3.0.4-r1 USE="unicode -debug
-hwloc -lm-sensors -openvz -vserver"

Would you like to merge these packages? [Yes/No] Yes
>>> Verifying ebuild manifests
>>> Emerging (1 of 1) sys-process/htop-3.0.4-r1::gentoo
>>> Jobs: 0 of 1 complete, 1 running Load avg: 5.64, 5.89, 6.02
```

  W kodzie 1.1 dostrzegamy `USE="unicode -debug -hwloc -lm-sensors -openvz -vserver"`, widoczne tam są flagi, które podczas instalacji `Emerge` do kompiluje. Flaga oznaczana znakiem minus nie będzie brana pod uwagę.

[2]Lokalizacja pliku korzystającego z USE flags `/etc/portage/package.use`

- Dokumentacja

  Systemu Gentoo jest bardzo dobrze udokumentowany. Znajdziemy wszystkie niezbędne informacje dotyczące wybranego przez nas instalowanego pakietu, jego konfigurację (przykładowa konfiguracja pakietu `www-servers/apache`[9]), a także najnowsze wiadomości[10] ze świata Gentoo Linux.

- Trwałość

  Kolejną cechą przemawiającą za tym systemem jest jego trwałość. Same systemy operacyjne są wersjonowane (na przykład CentOS7, CentOS8[5] i tym podobne). Czyli chcąc zaktualizować system CentOS musimy go przeinstalować na nowo. W Gentoo Linux możemy posiadać niezaktualizowany system przez długi czas, gdzie w każdej chwili zaktualizować system do najnowszej wersji.

1.5 Portage

1.5.1 Definicja

`Portage` to oficjalny menadżer pakietów i system dystrybucji w Gentoo Linux. Jest sercem systemu. Zapewnia on zaawansowane rozwiązania zależności, elastyczne budowanie i instalację oprogramowania ze źródła, a także dodatkową jego funkcją jest tworzenie, zarządzanie i dystrybucja plików binarnych[2].

W plikach drzewa repozytoriów (rysunek 2) znajdują się pliki `.ebuild`, które służą jako przepisy dla `Portage`. Przykładowy plik `.ebuild` będzie omawiany w dalszej części (1.6)

1.5.2 Preferencje użytkownika

Użytkownik systemu może zmienić jego preferencje dotyczące na przykład ilości rdzeni używanych podczas kompilacji. Sam plik znajduje się w lokalizacji `/etc/portage/make.conf` (sekcja dotycząca zagadnienia 1.5.4) [4]

1.5.3 Flagi w Portage

Podczas samej instalacji Gentoo Linux użytkownicy dokonują wyborów do czego będą używali systemu. Czy system będzie wykorzystywany na stacji roboczej, serwerze, a może na komputerze przenośnym? Każdy pakiet posiada jakieś flagi dotyczące jego konfiguracji albo funkcjonalności. Przykładem może być pakiet www-client/firefox-bin do jego uruchomienia będziemy potrzebowali środowiska graficznego. Aby móc zainstalować ten pakiet w środowisku graficznym w jego flagach musimy wybrać `X` - oznaczającą włączeniem tej flagi. Kolejnym przykładem będzie kod znajdujący się w sekcji 1.1.[6]
Same flagi możemy konfigurować na poziome globalnym całego sytemu w pliku konfiguracyjnym 1.5.4 `/etc/portage/make.conf` lub konkretnego pakietu w pliku `/etc/portage/package.use`.[6]

1.5.4 Przykładowy plik konfiguracyjny

Plik `/etc/portage/make.conf` wygląda następująco:

Kod źródłowy 1.2: Przykładowy plik `/etc/portage/make.conf`. Źródło: Opracowanie własne

```
COMMON_FLAGS="-march=native -O2 -pipe"
CFLAGS="${COMMON_FLAGS}"
CXXFLAGS="${CFLAGS}"
FCFLAGS="${CFLAGS}"
```

```
FFLAGS="${CFLAGS}"

PORTAGE_BINHOST="http://packages.gentooexperimental.org/packages/"
GENTOO_MIRRORS="https://mirror.eu.oneandone.net/linux/distributions/
    gentoo/gentoo/"

MAKEOPTS="-j8 --load-average=6.4"
EMERGE_DEFAULT_OPTS="--jobs=8"

ACCEPT_LICENSE="*"

USE="-bluetooth -kde -gnome -systemd"

PORTDIR="/var/db/repos/gentoo"
DISTDIR="/var/cache/distfiles"
PKGDIR="/var/cache/binpkgs"
CCACHE_DIR="/var/cache/ccache"

LC_MESSAGES=C
GRUB_PLATFORMS="efi-64"
```

W powyższym pliku konfiguracyjnym możemy zdefiniować różne zmienne, którą wpływają na instalację oraz kompilację pakietów. Dla przykładu zmienna `EMERGE_DEFAULT_OPTS="--jobs=8"` oznacza, że każdy pakiet będzie kompilowany do ośmiu równoległych kompilacji oraz `ACCEPT_LICENSE="*"` oznacza akceptuj każdą licencję. Jeżeli chodzi o inne flagi informację o nich możemy znaleźć na stronie Gentoo [4].

1.5.5 Przykłady użycia

Aby zainstalować na przykład pakiet `app-editors/vim` z `Portage` używamy następującej komendy 1.3.

Kod źródłowy 1.3: Przykładowe wywołanie instacji pakietu `vim` z kategorii `app-editors`

```
rescue / # emerge --ask vim


These are the packages that would be merged, in order:


Calculating dependencies... done!
[ebuild N ] app-eselect/eselect-vi-1.2
[ebuild N ] dev-libs/libsodium-1.0.18_p20210617 USE="asm urandom -
    minimal -static-libs -verify-sig" ABI_X86="(64) -32 (-x32)"
    CPU_FLAGS_X86="-aes -sse4_1"
[ebuild N ] app-editors/vim-core-8.2.4328-r1 USE="acl nls -minimal"
[ebuild N ] app-editors/vim-8.2.4328-r1 USE="acl crypt nls -X -
    cscope -debug -gpm -lua -minimal -perl -python -racket -ruby (-
    selinux) -sound -tcl -terminal -vim-pager" LUA_SINGLE_TARGET="
    lua5-1 -luajit" PYTHON_SINGLE_TARGET="python3_9 -python3_8 -
    python3_10"
[ebuild N ] app-vim/gentoo-syntax-1 USE="-ignore-glep31"


Would you like to merge these packages? [Yes/No] Yes
```

W kodzie 1.3 dostrzegamy `USE="acl crypt nls -X -cscope -debug -gpm -lua -minimal -perl -python -racket -ruby (-selinux)-sound -tcl -terminal -vim-pager"`, widoczne tam są flagi, które podczas instalacji `Emerge` do kompiluje. Flaga oznaczana znakiem minus nie będzie brana pod uwagę. Dodatkowo widzimy także, że podczas instalacji pakietu `app-editors/vim` doinstalowanie

zostanie na przykład `dev-libs/libsodium`, który jest potrzebny do poprawnego działania pakietu.

1.6 Ebuild - plik instalacyjny w systemie Gentoo Linux

1.6.1 Definicja pliku `ebuild`

Plik `ebuild` to plik tekstowy, zwykle przechowywany w repozytorium, który identyfikuje konkretny pakiet oprogramowania i mówi menedżerowi pakietów Gentoo, jak go obsłużyć. Pliki `ebuild` używają stylu składni podobnego do bash i są ustandaryzowane oraz zgodnie z określoną wersją EAPI (aktualna wersja to 8)[11].

1.6.2 Jak pisać pliki ebuild?

Aby stworzyć pakiet dla przykładu `app-misc/scrub` w wesji `2.6.1`.

1. Utwórz w lokalnym przykładowym repozytorium folder `app-misc/scrub`

```
mkdir -p /var/db/repos/example_repository/app-misc/scrub
```

2. Zmień folder na `/var/db/repos/example_repository/app-misc/scrub`

3. Przejdź do wybranego edytora i utwórz plik `scrub-2.6.1.ebuild`

4. Podczas edycji skopiuj szablon

```
# Copyright 1999-2022 Gentoo Authors
# Distributed under the terms of the GNU General Public License
    v2
```

```
EAPI=8

DESCRIPTION="Some words here"
HOMEPAGE="https://github.com/chaos/scrub"
SRC_URI="https://github.com/chaos/${PN}/releases/download/${PV
    }/${P}.tar.gz"

LICENSE="GPL-2"
SLOT="0"
KEYWORDS="~amd64 ~x86"
IUSE=""

DEPEND=""
RDEPEND="${DEPEND}"
BDEPEND=""
```

5. Aby przetestować `scrub-2.6.1.ebuild` uruchom komendę

```
ebuild ./scrub-2.6.1.ebuild manifest clean unpack
```

6. Aby zainstalować `scrub-2.6.1.ebuild` uruchom komendę

```
ebuild ./scrub-2.6.1.ebuild clean install merge
```

Rozdział 2

Instalacja pakietów binarnych a kompilacja ze źródła

Cały rozdział powstał na podstawie artykułu [13].

2.1 Pakiety binarne

2.1.1 Definicja pakietów binarnych

Pakiety, które znajdują się w repozytoriach posiadające rozszerzenia archiwum. To łączy wszystko w jeden plik w celu łatwego dostępu i dystrybucji. Same pakiety binarne pobierane są z oficjalnych repozytoriów.

2.1.2 Przykładowa instalacja pakietu binarnego w Gentoo Linux

Jeżeli pakiet posiada sufiksu `-bin` na przykład `www-client/firefox-bin`. Aby zainstalować pakiet wykonujemy:

```
emerge www-client/firefox-bin
```

Natomiast jeżeli pakiet nie posiada sufiksu `-bin` musimy napierw zdefiniować zmiennej `PORTAGE_BINHOST` w pliku `/etc/portage/make.conf`. Po dodaniu zmiennej można użyć komendy, aby pobrać

2.2 Pakiety kompilowane ze źródła

2.2.1 Definicja pakietów kompilowanych

Pakiety, takie gdzie konfiguracja, kompilacja oraz instalacja odbywa się na lokalnym komputerze nazywamy pakietem kompilowanym. Sam kod źródłowy pochodzi z oficjalnego repozytorium danego pakietu.

2.2.2 Instalacja kompilowanego pakietu

Aby zainstalować pakiet, który wymaga kompilacji używamy

```
emerge <NAZWA-PAKIETU>
```

Rozdział 3

Sposoby instalacji pakietów

3.1 GNU Compiler Collection (GCC)

Sama instalacja pakietów w systemie Gentoo Linux w sposób domyślimy używa `GCC` (`GNU Compiler Collection`). Większość pakietów instaluje się z użyciem wcześniej wspomnianego kompilatora. Przykładem takiego pakietu są między innymi `www-client/firefox` oraz `app-office/libreoffice`. W dalszej części pracy zostaną porównane sposoby instalacji owych pakietów.

3.2 GCC razem z CCACHE

Jeżeli pakiet musimy ponownie kompilować albo instalujemy nową wersję pakietu możemy w systemie Gentoo Linux skonfigurować `CCACHE`. Samym powodem ponownej kompilacji może być błąd albo błędnie zainstalowany pakiet. Pakiet `CCACHE` działa w następujący sposób: podczas zwykłej kompilacji

pakietu, dodatkowo zapisuje pliki, które zostały przekompilowane. Pliki te są odczytywane oczywiście jeżeli kod źródłowy nie uległ zmianie. Wtedy ponowna kompilacja będzie wykonywać się o wiele krótszym czasie. Wszelkie informacje dotyczące konfiguracji `CCACHE` można znaleźć na Gentoo Wiki [12].

Rozdział 4

Instalacja pakietu www-client/firefox

4.1 Z użyciem GCC

Aby zainstalować pakiet `www-client/firefox` należy uruchomić komendę jako superużytkownik albo normalny użytkownik z prawami superużytkownika.

```
sudo emerge www-client/firefox
```

4.2 Z użyciem CCACHE

Aby zainstalować pakiet `www-client/firefox` z użyciem `CCACHE` należy najpierw zainstalować oraz skonfigurować pakiet `dev-util/ccache` (zgodnie z instrukcją [12]). Następnie uruchomić komendę jako superużytkownik albo

normalny użytkownik z prawami superużytkownika.

```
sudo emerge www-client/firefox
```

4.3 Z użyciem wersji binarnej

Instalacja przekompilowanej wersji pakietu `www-client/firefox` (binarnej) należy uruchomić komendę jako superużytkownik albo normalny użytkownik z prawami superużytkownika

```
sudo emerge www-client/firefox-bin
```

Rozdział 5

Instalacja pakietu app-office/libreoffice

5.1 Z użyciem GCC

Aby zainstalować pakiet `app-office/libreoffice` należy uruchomić komendę jako superużytkownik albo normalny użytkownik z prawami superużytkownika.

```
sudo emerge app-office/libreoffice
```

5.2 Z użyciem CCACHE

Aby zainstalować pakiet `app-office/libreoffice` z użyciem `CCACHE` należy najpierw zainstalować oraz skonfigurować pakiet `dev-util/ccache` (zgodnie z instrukcją [12]). Następnie uruchomić komendę jako superużytkownik albo

normalny użytkownik z prawami superużytkownika.

5.3 Z użyciem wersji binarnej

Instalacja przekompilowanej wersji pakietu `app-office/libreoffice` (binarnej) należy uruchomić komendę jako superużytkownik albo normalny użytkownik z prawami superużytkownika

```
sudo emerge app-office/libreoffice-bin
```

Rozdział 6

Porównanie sposobów kompilacji

Wielokrotne testy instalacji pakietów odbywały się na maszynie w serwerowni firmy Hetzner posiadającej parametry:

- CPU: Intel i7-4770 (8) @ 3.900GHz
- GPU: Intel HD Graphics
- RAM: 31973MiB
- OS: Gentoo Linux

	www-client/firefox-101.0.1	app-office/libreoffice-7.3.4.2
GCC	0h:41m:00s	1h:08m:55s
GCC + CCACHE	0h:43m:53s	1h:14m:55s

Tablica 6.1: Pierwszy raz kompilacji pakietów `www-client/firefox-101.0.1` oraz `app-office/libreoffice-7.3.4.2`

	www-client/firefox-101.0.1	app-office/libreoffice-7.3.4.2
GCC	0h:41m:00s	1h:08m:55s
GCC + CCACHE	0h:11m:48s	0h:04m:06s

Tablica 6.2: Kolejny raz kompilacji pakietów `www-client/firefox-101.0.1` oraz `app-office/libreoffice-7.3.4.2`

Podczas kompilowania pakietu `www-client/firefox-101.0.1` dostrzegalna jest poprawa szybkości podczas kolejnej kompilacji. Pierwszy raz trwał 41 minut ale kolejny z dodatkowym użyciem `CCACHE` trwał prawie 12 minut.

Praktycznie identyczna sytuacja jest podczas kompilowania pakietu `app-office/libreoffice-7.3.4.2`. Gdy normalna kompilacja trwa godzinę oraz 8 minut, razem z `CCACHE` trwa 4 minuty.

Rozdział 7

Projekt na użytek pracy - automatyzacja instalacji systemu Gentoo Linux za pomocą Ansible

Podczas pracy magisterskiej postanowiłem, że utworzę skrypt opierający się na `Ansible` automatyzujący instalację Gentoo Linux.

7.1 Definicja Ansible

Ansible to narzędzie udostępniania oprogramowania, zarządzania konfiguracją i wdrażania aplikacji, umożliwiające tworzenie infrastruktury jako

kodu[14]. Do komunikacji `Ansible` używa protokołu SSH[1].

7.2 Jak działa Ansible?

Po napisaniu scenariusza (ang. playbook) sam scenariusz, może opisać kilkanaście zadań. Dla przykładu chcemy zaimplementować instalację `Nginx` na systemie operacyjnym `Ubuntu`. Plik realizujący podane informacje znajduje się poniżej 7.1.

Kod źródłowy 7.1: Przykładowy plik `webservers.yml`

Źródło: "Ansible w praktyce" [15]

```
---
- name: Konfiguracja serwerów WWW
  hosts: webservers
  tasks:
    - name: Instalacja Nginx
      apt: name=nginx
    - name: Instalacja pliku konfiguracyjnego
      template: src=nginx.conf.j2
        dest=/etc/nginx/ngingx.conf
      notify: Restart Nginx
  handlers:
    - name: Powiadomienie Nginx
      service: name=nginx
        state=restarted
```

Plik `webservers.yml` 7.1:

1. Tworzy skrypt Pythona bazujący na pliku `.yml`, który instaluje pakiet Nginx

[1]Bezpieczny protokół powłoki (ang.Secure Shell Protocol)

2. Kopiuje skrypt do serwerów, które zdefiniowaliśmy jako `webservers`

3. Uruchamia skrypt na serwerach (jednocześnie)

4. Czeka, aż wszystkie zadania zostaną wykonane na serwerach

7.3 Rola dotycząca automatycznej instalacji Gentoo Linux

Sama rola została napisana przez autora pracy magisterskiej. Rolę można zaimplementować do każdego scenariusza, oraz komputera. Sam kod źródłowy znajduje się na repozytorium GitHub. Aby moć uruchomić tą rolę trzeba napisać scenariusz albo użyć gotowego, który także znajduje się na repozytorium GitHub

7.4 Przykładowe użycie scenariusza

Sam scenariusz został pobrany z repozytorium GitHub. Uruchamiamy za pomocą komendy. Sam scenariusz

```
yorune@MacBook ~/git/gentoo-installer main $ ansible-playbook -u
    root -i 148.251.79.183, main.yml -vv
```

Kod źródłowy 7.2: Przykładowy użycie scenariusza. Źródło: Opracowanie własne

```
PLAY [Simple playbook to install gentoo in Hetzner]

TASK [Gathering Facts]
ok: [148.251.79.183]
```

```
TASK [ansible−gentoo−installer : Load a vars]
ok: [148.251.79.183]

TASK [ansible−gentoo−installer : include_tasks]
included: roles/ansible−gentoo−installer/tasks/create_partitions.yml for 148.251.79.183

TASK [ansible−gentoo−installer : Get info about partitions]
ok: [148.251.79.183]

TASK [ansible−gentoo−installer : Remove all partitions]
changed: [148.251.79.183] => (item={'num': 1, 'begin': 0.0, 'end': 0.0, 'size': 0.0, 'fstype
    ': '', 'name': 'primary', 'flags': ['bios_grub'], 'unit': 'tib'})
changed: [148.251.79.183] => (item={'num': 2, 'begin': 0.0, 'end': 0.0, 'size': 0.0, 'fstype
    ': 'ext4', 'name': 'primary', 'flags': ['boot', 'esp'], 'unit': 'tib'})
changed: [148.251.79.183] => (item={'num': 3, 'begin': 0.0, 'end': 1.82, 'size': 1.82, '
    fstype': 'ext4', 'name': 'rootfs', 'flags': [], 'unit': 'tib'})

TASK [ansible−gentoo−installer : Create partitions grub_part]
changed: [148.251.79.183]

TASK [ansible−gentoo−installer : Create partitions boot_part]
changed: [148.251.79.183]

TASK [ansible−gentoo−installer : Create partitions root_part]
changed: [148.251.79.183]

TASK [ansible−gentoo−installer : Format partition grub]
changed: [148.251.79.183]

TASK [ansible−gentoo−installer : Format partition root]
changed: [148.251.79.183]

[...]
```

```
PLAY RECAP
148.251.79.183 : ok=47 changed=14 unreachable=0 failed=0 skipped=0 rescued=0
    ignored=0
```

Spis rysunków

Spis tablic

Lista kodów źródłowych

Bibliografia

[1] Strona poświęcona Gentoo Linux na Wikipedii
https://en.wikipedia.org/wiki/Gentoo_Linux,
dostęp maj 2022 r.

[2] Gentoo Wiki - Portage
https://wiki.gentoo.org/wiki/Portage,
dostęp maj 2022 r.

[3] Gentoo Wiki - Stage tarball
https://wiki.gentoo.org/wiki/Stage_tarball,
dostęp maj 2022 r.

[4] Gentoo Wiki - Stage tarball
https://wiki.gentoo.org/wiki//etc/portage/make.conf,
dostęp maj 2022 r.

[5] Uwagi do wydania CentOS,
https://wiki.centos.org/Manuals/ReleaseNotes,
dostęp maj 2022 r.

[6] Idea Flag w Gentoo Linux
https://wiki.gentoo.org/wiki/Handbook:AMD64/Working/USE,
dostęp maj 2022 r.

[7] Architektury Systemu Gentoo,
https://wiki.gentoo.org/wiki/Property:Architecture,
dostęp maj 2022 r.

[8] Dokumentacja instalacji systemu Gentoo Linux,
https://wiki.gentoo.org/wiki/Handbook:Main_Page,
dostęp maj 2022 r.

[9] Dokumentacja pakietu www-servers/apache,
https://wiki.gentoo.org/wiki/Apache,
dostęp maj 2022 r.

[10] Najnowsze wiadomości i artykuły systemu Gentoo Linux,
https://www.gentoo.org/news/,
dostęp maj 2022 r.

[11] Package Manager Specification,
https://projects.gentoo.org/pms/8/pms.pdf,
dostęp maj 2022 r.

[12] Gentoo Wiki dot. CCACHE,
https://wiki.gentoo.org/wiki/Ccache,
dostęp maj 2022 r.

[13] Artykuł: "Pakiety binarne a pakiety źródłowe: których należy użyć?"
https://www.makeuseof.com/tag/binary-source-packages-use/,
dostęp maj 2022 r.

[14] Strona poświęcona Ansible na Wikipedii
https://en.wikipedia.org/wiki/Ansible_(software),
dostęp maj 2022 r.

[15] Ansible w praktyce - Automatyzacja konfiguracji i proste instalowanie systemów
Autorzy: Lorin Hochstein, René Moser
ISBN Książki drukowanej: 978-83-283-4172-2