



**Politechnika
Śląska**

Dokumentacja realizowanego projektu

2020/2021

Zarządzanie systemami informatycznymi

*Demonstrator wybranego systemu informatycznego związanego
z DevOps na przykładzie Puppet*

Kierunek: Informatyka

Członkowie zespołu:

Paulina Loska

Marta Kirsch

Gliwice, 2020/2021

Spis treści

1	Wprowadzenie	2
1.1	Role w projekcie	2
1.2	Cel projektu	2
2	Założenia projektowe	3
2.1	Założenia techniczne i nietechniczne	3
2.2	Stos technologiczny	3
2.3	Oczekiwane rezultaty projektu	3
3	Realizacja projektu	4
3.1	Problematyka	4
3.2	Wstęp teoretyczny	4
3.3	Tutorial	5
4	Wnioski	14

1 Wprowadzenie

1.1 Role w projekcie

Na wykonanie projektu składa się przeanalizowanie założeń wstępnych przed samą realizacją projektów, w tym wstęp teoretyczny, następnie implementacja, wykonanie testów po implementacji i sporządzenie dokumentacji. W późniejszej fazie zawieramy również prezentację stworzonego projektu oraz dyskusja.

Zadania rozdzieliliśmy następująco:

Założenia - Paulina Loska

Implementacja - Marta Kirsch

Testy po implementacji - Marta Kirsch

Dokumentacja - Paulina Loska

1.2 Cel projektu

Celem naszego projektu jest przedstawienie jednego, wybranego przez nas spośród trzech, systemu informatycznego związanego z DevOps. Jest to zarówno opis jego działania, założenia samego systemu informatycznego, jak na przykład - dla kogo jest stworzony, bądź jaki ma cel działania i funkcjonowania, jak również przedstawienie jego działania. Naszym wybranym systemem jest Puppet. Projektowi przyświeca następujący cel: aby osoba, która wejściowo nie wie nic z zakresu DevOps, po przejściu przez nasz projekt znała problematykę tematu i potrafiła samodzielnie wykonać model za pomocą programu Puppet.

2 Założenia projektowe

2.1 Założenia techniczne i nietechniczne

Założenia techniczne to stanowisko komputerowe z programami/funkcjami wymienionymi w poniższym punkcie. Założenia nietechniczne to kilka godzin całkowitego spokoju, aby móc w pełni skupić się nad projektem.

2.2 Stos technologiczny

- komputer z systemem Windows z zainstalowanym oprogramowaniem wirtualizacyjnym (np. Oracle VM VirtualBox). Dwie wirtualne maszyny CentOS 7 z dostępem do internetu (maszyny powinny mieć przypisane adresy ip należące do sieci hosta)
lub
- dwa komputery z systemem CentOS 7 z dostępem do internetu.

2.3 Oczekiwane rezultaty projektu

Przedstawienie działania systemu informatycznego Puppet. Znajomość problematyki, nomenklatury oraz umiejętność samodzielnego wykonania modelu, np. z użyciem maszyn wirtualnych.

3 Realizacja projektu

3.1 Problematyka

Pan Jarek pracuje w firmie informatycznej. Jest odpowiedzialny za zajmowanie się serwerami Linux, firma posiada ich sto. Pan Jarek dostał polecenie, aby na każdym z serwerów pojawił się konkretny plik tekstowy z zadaną zawartością. Złapał się za głowę, zaczynając kolejno podchodzić do serwerów i wprowadzając założone zmiany. Miał nadzieję, że zdąży ukończyć zadanie do końca tygodnia. W pracy przydatne byłoby narzędzie, dzięki któremu wprowadzi zmiany jedynie na jednym serwerze, a na pozostałych zmiany te zastosowane zostaną automatycznie.

3.2 Wstęp teoretyczny

Puppet to bardzo popularne oprogramowanie, które ułatwia zarządzanie serwerami. Automatyzuje też ich konfigurację. Mamy sporo możliwości, jak tworzenie, modyfikację plików, zmiany ich praw, poprzez instalację wymaganego oprogramowania oraz monitorowania wybranych procesów, a w razie problemów ich automatyczne uruchomienie - te wszystkie procesy możemy zdefiniować na centralnym serwerze, a nowe maszyny automatycznie wprowadzą poprawki.

- puppet server (serwer) - komputer/serwer, na którym wprowadzać będziemy zmiany (kolejno dostarczane do puppet klientów)
- puppet client (klient) - komputer/serwer, który zgłosi się do puppet serwera, deklarując chęć otrzymywania od niego aktualizacji
- manifest - skrypt napisany w języku Puppet DSL, który zawiera informacje o potrzebnych zmianach na komputerach puppet klientów. Dzięki językowi Puppet DSL manifesty tworzy się bardzo intuicyjnie, składnia przypomina tworzenie zdań w języku angielskim.
- certyfikat - posiada dołączony **hostname** potencjalnego klienta, który wysyła go z prośbą o podpisanie. Podpisanie przez puppet serwer certyfikatu oznacza uznanie danego innego serwera za swojego klienta.
- puppet facter - usługa, która zbiera informacje o aktualnym stanie puppet klienta, a następnie wysyła je do serwera. Serwer następnie porównuje otrzymane **facts** ze swoimi manifestami. Jeśli należy wprowadzić jakieś zmiany na puppet kliencie, serwer kompiluje zmiany do **katalogów** i wysyła je do swoich klientów.

- puppet agent - usługa, która odbiera wysłane od puppet serwera **katalogi** i wykonuje je lokalnie. Po wprowadzeniu zmian, generowany jest raport i wysyłany do puppet serwera.

Na początku, aby puppet serwer uznał dany inny serwer za swojego klienta, musi zostać nawiązane połączenie SSL między nimi. Potencjalny klient wysyła prośbę o podpisanie certyfikatu do puppet serwera, który następnie odsyła mu podpisany certyfikat. Od tego momentu może następować wymiana danych pomiędzy puppet serwerem a klientem. Aktualizacje stanu z serwera można ustawić tak, aby automatycznie pobierały się same (np. co dany interwał czasu).

3.3 Tutorial

Na serwerze ustawiamy nazwę hosta (hostname) na puppet (na chwilę realizowania projektu jest to wymagana przez oprogramowanie domyślna nazwa - można ją zmienić w plikach konfiguracyjnych).

```
[root@centos7 ~]# hostnamectl set-hostname puppet
[root@centos7 ~]# hostname
puppet
[root@centos7 ~]# _
```

Sprawdzamy i zapisujemy przydzielony serwerowi adres ip.

```

CentOS Linux 7 (Core)
Kernel 3.10.0-957.21.3.el7.x86_64 on an x86_64

puppet login: root
Password:
Last login: Thu May 20 11:58:22 on tty1
[root@puppet ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:0e:79:87 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global noprefixroute dynamic enp0s3
        valid_lft 86388sec preferred_lft 86388sec
    inet6 fe80::e7c3:44e3:b24a:f7d3/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:a4:e8:bc brd ff:ff:ff:ff:ff:ff
    inet6 fe80::8c44:3fad:551d:59/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:41:31:3c brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.191/24 brd 192.168.0.255 scope global noprefixroute dynamic enp0s9
        valid_lft 86388sec preferred_lft 86388sec
    inet6 fe80::5bda:5078:527d:efbb/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[root@puppet ~]#

```

Za pomocą komendy instalujemy puppetlabs.

```

[root@puppet ~]# sudo rpm -Uvh https://yum.puppetlabs.com/puppetlabs-release-pc1-el-7.noarch.rpm
Pobieranie https://yum.puppetlabs.com/puppetlabs-release-pc1-el-7.noarch.rpm
ostrzeżenie: /var/tmp/rpm-tmp.UnCtFj: Nagłówek U4 RSA/SHA1 Signature, identyfikator klucza ef8d349f: NOKEY
Przgotowywanie...
Aktualizowanie/instalowanie...
1:puppetlabs-release-pc1-1.1.0-5.el7.noarch.rpm [100%]
[root@puppet ~]# _

```

Instalujemy puppetserver.

```

[root@puppet ~]# sudo rpm -Uvh https://yum.puppetlabs.com/puppetlabs-release-pc1-el-7.noarch.rpm
Pobieranie https://yum.puppetlabs.com/puppetlabs-release-pc1-el-7.noarch.rpm
ostrzeżenie: /var/tmp/rpm-tmp.VnGtFj: Nagłówek U4 RSA/SHA1 Signature, identyfikator klucza ef8d349f:
NOKEY
Przygotowywanie... [100%]
Aktualizowanie/instalowanie...
1:puppetlabs-release-pc1-1.1.0-5.el7.noarch.rpm [100%]
[root@puppet ~]# sudo yum -y install puppetserver

```

Aby serwer mógł otrzymywać informacje/żądania etc, musimy zdez-
aktywować **zaporę ogniową**.

```

[root@puppet ~]# systemctl stop firewalld
[root@puppet ~]# _

```

Za pomocą odpowiednich komend włączamy puppetserver.


```
[root@puppet ~]# systemctl start puppetserver
^C
[root@puppet ~]# systemctl enable puppetserver
Created symlink from /etc/systemd/system/multi-user.target.wants/puppetserver.service to /usr/lib/systemd/system/puppetserver.service.
[root@puppet ~]#
```

Na systemie klienta edytujemy plik `/etc/hosts` poprzez dopisanie jednej linii (adres, który został wpisany, to wcześniej sprawdzony adres serwera).

```
[root@centos7 ~]# mcedit /etc/hosts
```

```
hosts      [-M--] 34 L:[ 1+ 2 3/ 3] *(192 / 192b) <EOF>      [*]IX]
127.0.0.1  localhost localhost.localdomain localhost4 localhost4.localdomain4
::1       localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.0.191 puppet puppet-master
```

Na komputerze klienta również instalujemy puppetlabs, a następnie puppet-agent (agenta).

```
[root@centos7 ~]# rpm -Uvh https://yum.puppetlabs.com/puppetlabs-release-pc1-el-7.noarch.rpm
Pobieranie https://yum.puppetlabs.com/puppetlabs-release-pc1-el-7.noarch.rpm
ostrzeżenie: /var/tmp/rpm-tmp.zF4u11: Nagłówek V4 RSA/SHA1 Signature, identyfikator klucza ef8d349f:
NOKEY
Przygotowywanie...
Aktualizowanie/instalowanie...
1:puppetlabs-release-pc1-1.1.0-5.el7.noarch.rpm [100%]
[root@centos7 ~]# yum -y install puppet-agent
```

Odpowiednią komendą włączamy agenta. Następnie sprawdzamy nazwę hosta. Po włączeniu agenta powinno zostać nawiązanie połączenie SSL oraz

wysłana prośba o podpisanie certyfikatu do serwera.

```
[root@centos7 ~]# /opt/puppetlabs/bin/puppet resource service puppet ensure=running enable=true
Notice: /Service[puppet]/ensure: ensure changed 'stopped' to 'running'
service { 'puppet':
  ensure => 'running',
  enable => 'true',
}
[root@centos7 ~]# hostname
centos7.localdomain
[root@centos7 ~]#
```

Na serwerze sprawdzamy, jakie certyfikaty są obecnie do podpisania. Wybieramy ten o nazwie hosta, którą miał nasz system kliencki. Odpowiednią komendą podpisujemy certyfikat. System ten staje się naszym certyfikowanym klientem.

```
lroot@puppet ~]# sudo /opt/puppetlabs/bin/puppet cert list
"centos7.localdomain" (SHA256) 71:CF:10:D1:34:23:CC:09:13:BC:8C:DA:DD:3F:75:7A:92:6C:0C:7B:EC:D5:4
2:CB:63:20:92:06:E4:5C:F8:0B
lroot@puppet ~]# sudo /opt/puppetlabs/bin/puppet cert sign centos7.localdomain
Signing Certificate Request for:
"centos7.localdomain" (SHA256) 71:CF:10:D1:34:23:CC:09:13:BC:8C:DA:DD:3F:75:7A:92:6C:0C:7B:EC:D5:4
2:CB:63:20:92:06:E4:5C:F8:0B
Notice: Signed certificate request for centos7.localdomain
Notice: Removing file Puppet::SSL::CertificateRequest centos7.localdomain at '/etc/puppetlabs/puppet
/ssl/ca/requests/centos7.localdomain.pem'
lroot@puppet ~]#
```

Na serwerze tworzymy nowy manifest, a następnie edytujemy go. Nasz przykładowy manifest będzie spełniał następującą funkcjonalność - upewni się, że na systemach klienckich (wszystkich - node z parametrem default) w katalogu /home/Marta znajdzie się plik plik.txt o zawartości 'to jest przykładowy manifest' oraz katalog folder. Po zakończeniu edycji, zapisujemy zawartość manifestu.

```
[root@puppet ~]# mcedit /etc/puppetlabs/code/environments/production/manifests/testmanifest.pp

[root@puppet ~]# touch /etc/puppetlabs/code/environments/production/manifests/testmanifest.pp

testmanifest.pp  [-M--] 22 L: [ 1+ 1 2/ 121 *(36 / 183b) 0039 0x027  [*]I[X]
mode default{
file{'/home/Marta/plik':
content=>'to jest przykładowy manifest, milego dnia C:',
ensure=>'present'
}

file{'/home/Marta/folder':
ensure=>'directory',
owner=>'root'
}
}

1Pomoc 2Zapisz 3Zaznacz 4Zastąp 5Kopiuj 6Przen 7Szukaj 8Usuń 9Wdół 10Kończ
```

Po zapisaniu manifestu, przechodzimy na nasz system kliencki. Sprawdzamy aktualną zawartość folderu /home/Marta. Zauważamy, że póki co nie ma tam pozycji wymienionych w poprzednim kroku. Odpowiednią komendą aktualizujemy agenta (wysyłamy do serwera zapytanie, a on następnie porów-

nując stan naszego systemu ze swoimi manifestami, odsyła nam informacje o koniecznych zmianach). Po zakończeniu tej operacji ponownie sprawdzamy zawartość katalogu /home/Marta. Zadeklarowane w manifeście pozycje pojawiły się. Wypisujemy zawartość pliku plik.txt, która również zgodna jest z zadaną w manifeście.

```
[root@centos7 ~]# ls /home/Marta
[root@centos7 ~]# /opt/puppetlabs/bin/puppet agent --test
Info: Using configured environment 'production'
Info: Retrieving pluginfacts
Info: Retrieving plugin
Info: Caching catalog for centos7.localdomain
Info: Applying configuration version '1621587161'
Notice: /Stage[main]/Main/Node[default]/File[/home/Marta/plik]/ensure: defined content as '{md5}db4db823f6e8de63bee7c84c869729a5'
Notice: /Stage[main]/Main/Node[default]/File[/home/Marta/folder]/ensure: created
Notice: Applied catalog in 0.03 seconds
[root@centos7 ~]# ls /home/Marta
folder  plik
[root@centos7 ~]# cat /home/Marta/plik
to jest przykładowy manifest, milego dnia C:[root@centos7 ~]#
```

4 Wnioski

- Podczas tworzenia projektu dowiedzieliśmy się o istnieniu takiego systemu informatycznego jak Puppet, jego plusów oraz możliwych zastosowań i ułatwienia wielu rozwiązań.
- W naszym dwuosobowym zespole jeszcze bardziej udoskonaliłyśmy współpracę na poziomie projektowym i kolejny raz przekonaliśmy się o potędze pracy w grupie.
- Chcemy w przyszłości zainwestować wiele naszego czasu i sił do kontynuacji nauki Zarządzania Systemami Informatycznymi, gdyż jest to mocną podstawą do wielu rozwiązań.