Agenci: dodanie, diagnostyka, utrzymanie

Poziom: operacyjny **Dotyczy: Jenkins LTS, agent SSH na RHEL/Rocky Linux**

Czas: ~40 min

Dodanie agenta to w praktyce dwie osobne roboty — przygotowanie maszyny (dysk, użytkownik, klucze SSH) i konfiguracja node'a w samym Jenkinsie. Ta strona pokazuje obie, razem z diagnostyką najczęstszych powodów, dla których agent nagle staje się offline.

Skróty użyte na tej stronie (7)

Skrót	Co znaczy
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
GB	Gigabyte: miliard bajtów, jednostka pojemności pamięci i plików, tu najczęściej rozmiar modelu albo pamięci karty graficznej
IP	Internet Protocol: adres komputera w sieci; w elektryce: stopień ochrony obudowy (Ingress Protection)
JDK	Java Development Kit: zestaw do uruchamiania i budowania programów w Javie
LTS	Long Term Support: wersja z długim wsparciem poprawkami
RHEL	Red Hat Enterprise Linux: komercyjna dystrybucja Linuksa
SSH	Secure Shell: szyfrowane zdalne logowanie do serwera

Fiszka

Komendy na serwerze agenta wykonujemy z CLI bash z uprawnieniami root (sudo).

TYLKO ODCZYT

Sprawdzenie widocznych dysków: lsblk

```
lsblk
```

Sprawdzamy widoczne dyski.

lsblk = list block devices: pokaż dyski i partycje

ZMIENIA SYSTEM

Utworzenie wolumenu fizycznego: pvcreate

```
pvcreate /dev/sdX
```

(gdzie `/dev/sdX` to nazwa urządzenia blokowego przeznaczonego pod katalog agenta)

pvcreate = physical volume create: oddaj dysk pod LVM

ZMIENIA SYSTEM

Utworzenie grupy wolumenów: vgcreate

```
vgcreate vgJenkins_Agent /dev/sdX
```

Tworzymy grupę wolumenów.

vgcreate = volume group create: utwórz grupę wolumenów

ZMIENIA SYSTEM

Utworzenie wolumenu logicznego: lvcreate

```
lvcreate -l +100%FREE -n lvJenkins_Agent vgJenkins_Agent
```

Tworzymy wolumen logiczny na całej dostępnej przestrzeni tej grupy.

lvcreate = logical volume create: utwórz wolumen logiczny

OSTROŻNIE · NIEODWRACALNE

Założenie systemu plików: mkfs.ext4

```
mkfs.ext4 /dev/mapper/vgJenkins_Agent-lvJenkins_Agent
```

Zakładamy system plików.

mkfs.ext4 = od make filesystem: zakłada na partycji lub wolumenie system plików ext4, niszcząc dotychczasowe dane

TYLKO ODCZYT

Dodanie wpisu w /etc/fstab

```
/dev/mapper/vgJenkins_Agent-lvJenkins_Agent /home/jenkins_agent ext4 defaults 1 2
```

Dodajemy wpis w `/etc/fstab`.

ZMIENIA SYSTEM

Zamontowanie zgodnie z nowym wpisem w fstab: mount -a

```
mount -a
```

Montujemy wolumen zgodnie z nowym wpisem w fstab.

mount = podepnij system plików z urządzenia albo obrazu pod wskazany katalog

TYLKO ODCZYT

Sprawdzenie zamontowania wolumenu: df -h

```
df -h /home/jenkins_agent
```

Sprawdzamy, czy nowy mountpoint się pojawił.

Przykład:

```
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/vgJenkins_Agent-lvJenkins_Agent  9.8G   24K  9.3G   1% /home/jenkins_agent
```

df = disk free: ile miejsca wolnego na dyskach

ZMIENIA SYSTEM

Utworzenie użytkownika systemowego: useradd

```
useradd -d /home/jenkins_agent jenkins_agent
```

Zakładamy dedykowanego użytkownika systemowego, którego katalogiem domowym będzie właśnie zamontowany wolumen.

useradd = user add: tworzy nowe konto systemowe w Linuksie

ZMIENIA SYSTEM

Nadanie uprawnień do katalogu: chown

```
chown -R jenkins_agent:jenkins_agent /home/jenkins_agent
```

Nadajemy właściwe uprawnienia do katalogu.

chown = change owner: zmień właściciela

ZMIENIA SYSTEM

Ograniczenie dostępu do katalogu domowego: chmod

```
chmod 700 /home/jenkins_agent/
```

Ograniczamy dostęp do katalogu domowego wyłącznie dla właściciela.

chmod = change mode: zmień prawa dostępu

ZMIENIA SYSTEM

Instalacja gita: yum install

```
sudo yum install git
```

Instalujemy gita, potrzebnego do checkoutu repozytorium podczas buildów.

yum = Yellowdog Updater Modified: starszy menedżer pakietów RHEL

ZMIENIA SYSTEM

Dodanie użytkownika do grupy docker: usermod

```
usermod -aG docker jenkins_agent
```

Jeśli buildy na tym agencie mają budować obrazy kontenerów, dopisujemy użytkownika do grupy `docker`.

usermod = zmień ustawienia istniejącego konta systemowego, na przykład przynależność do grupy

TYLKO ODCZYT

Wejście do kontenera aplikacji Jenkins (wymiana kluczy SSH): docker exec

```
sudo docker exec -it jenkins_app /bin/bash
```

Połączenie wykonujemy z wnętrza kontenera aplikacji Jenkins, jeśli kontroler jest uruchomiony jako kontener.

SSH = Secure Shell: szyfrowane zdalne logowanie do serwera · **docker** = narzędzie do kontenerów

TYLKO ODCZYT

Połączenie SSH z serwerem agenta: ssh

```
ssh jenkins_agent@app01.ipa.zielona.example
```

Z poziomu tej powłoki łączymy się na docelowego użytkownika na serwerze agenta.

Przykład pierwszego połączenia:

```
The authenticity of host 'app01.ipa.zielona.example (10.20.10.41)' can't be established.
ED25519 key fingerprint is SHA256:mjt4fQzt+BapuKXEQqgPVWQXGnTheNfqd1nkQd94Ri18.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'app01.ipa.zielona.example,10.20.10.41' (ED25519) to the list
of known hosts.
```

SSH = Secure Shell: szyfrowane zdalne logowanie do serwera · **ssh** = Secure Shell: szyfrowane zdalne logowanie

TYLKO ODCZYT

Wejście do kontenera aplikacji Jenkins (naprawa known_hosts): docker exec

```
sudo docker exec -it jenkins_app /bin/bash
```

W ramach naprawy wchodzimy do kontenera aplikacji Jenkins.

docker = narzędzie do kontenerów

TYLKO ODCZYT

Usunięcie przeterminowanego wpisu z known_hosts: ssh-keygen -R

```
ssh-keygen -f "/var/jenkins_home/.ssh/known_hosts" -R "app01.ipa.zielona.example"
```

Usuwamy przeterminowany wpis ze znanych hostów, dokładnie komendą podpowiedzianą w logu.

ssh-keygen = wygeneruj parę kluczy SSH albo sprawdź odcisk klucza

TYLKO ODCZYT

Ręczne zaakceptowanie nowego klucza hosta: ssh

```
ssh app01.ipa.zielona.example
```

Łączymy się ręcznie w celu zaakceptowania nowego odcisku klucza, zanim zrobi to sam Jenkins.

ssh = Secure Shell: szyfrowane zdalne logowanie

TYLKO ODCZYT

Sprawdzenie zajętości wolumenu agenta: df -h

```
df -h /home/jenkins_agent
```

Sprawdzamy zajętość wolumenu.

df = disk free: ile miejsca wolnego na dyskach

□□ TYLKO ODCZYT

Sprawdzenie rzeczywistej ścieżki Javy: readlink

```
readlink -f /etc/alternatives/jre/bin/java
```

Sprawdzamy, gdzie faktycznie stoi Java.

□□ **readlink** = pokaż, na co wskazuje dowiązanie symboliczne

□□ OSTROŻNIE · NIEODWRACALNE

Usunięcie przestarzałego remoting.jar: rm

```
rm -f /home/jenkins_agent/remoting.jar
```

Usuwamy przestarzały plik z katalogu roboczego agenta, żeby wymusić świeżą kopię przy następnym uruchomieniu.

□□ **rm** = remove: usuń (bez kosza, na zawsze)

Z życia wzięte

□□ Z ŻYCIA WZIĘTE

W poniedziałek rano nowy agent na `app01` upiera się, że jest offline, mimo że proces wygląda na uruchomiony. Tomek sprawdza log node'a i widzi ostrzeżenie o niezgodności klucza SSH, bo `app01` został niedawno przeinstalowany po awarii dysku. Po usunięciu starego wpisu z `known_hosts` kontenera Jenkinsa i akceptacji nowego klucza agent wraca online w kilka sekund.

Po co to istnieje

Jeden agent szybko przestaje wystarczać — kolejne zespoły chcą własnego środowiska buildowego (inny JDK, dostęp do innej sieci, więcej mocy obliczeniowej), a kolejka zadań na współdzielonym agencie robi się długa. Dodanie kolejnego agenta jest jedynym sposobem skalowania Jenkinsa w poziomie, bez dokładania zasobów do samego kontrolera.

Dodanie agenta po SSH

Zanim zaczniemy konfigurację po stronie Jenkinsa, sprawdzamy, że ruch po porcie 22 z serwera kontrolera do serwera agenta jest przepuszczony, a agent ma dostęp do wewnętrznego rejestru obrazów i repozytorium Gita, jeśli buildy z nich korzystają.

Przygotowanie dedykowanego wolumenu

Dobrą praktyką jest wydzielenie osobnego dysku pod katalog domowy agenta (standardowo 10 GB na start) zamiast trzymania buildów na wolumenie systemowym — zgodnie z konwencją nazewniczą `vg<Przeznaczenie>` / `lv<Przeznaczenie>` grupa nazywa się `vgJenkins_Agent`, wolumen `lvJenkins_Agent`.

Wymiana kluczy SSH

Zanim skonfigurujemy node w Jenkinsie, kontroler musi choć raz połączyć się po SSH z serwerem agenta, żeby wymienić klucze i zapisać odcisk hosta w swoim `known_hosts` — inaczej pierwsza próba uruchomienia agenta skończy się błędem weryfikacji klucza opisanym niżej.

Dodanie poświadczeń w Jenkinsie

W Manage Jenkins → Manage Credentials wybieramy odpowiednią domenę (np. Global) i dodajemy nowe poświadczenia:

Pole	Wartość
Kind	Username with password (lub SSH Username with private key)
Scope	Global (Jenkins, nodes, items, all child items, etc)
Username	`jenkins_agent`
ID	`jenkins_agent_app01`
Description	Credentials for SSH connection to jenkins agent on app01

Więcej o typach poświadczeń i ich zakresach na stronie Poświadczenia i bezpieczeństwo.

Konfiguracja node'a

W Manage Jenkins → Nodes → New Node wybieramy typ „Permanent Agent". Zamiast wypełniać konfigurację od zera, wygodnie jest skorzystać z opcji „Copy Existing Node" i wskazać już działającego agenta o zbliżonej roli jako szablon — trzeba wtedy tylko dostosować nazwę hosta, etykiety i poświadczenia.

Najważniejsze pola konfiguracji node'a:

Pole	Znaczenie
Remote root directory	`/home/jenkins_agent` — odpowiednik JENKINS_HOME, tylko po stronie agenta
Labels	etykiety, po których zadania wybierają ten agent, patrz Architektura Jenkinsa: kontroler, agenci, workspace
Usage	„Use this node as much as possible" albo „Only build jobs with label expressions matching this node"
Launch method	„Launch agents via SSH"
Host	`app01.ipa.zielona.example`
Credentials	poświadczenia dodane w kroku wyżej
Host Key Verification Strategy	„Known hosts file Verification Strategy" w normalnej pracy
Availability	„Keep this agent online as much as possible"

Po zapisaniu Jenkins przekierowuje do widoku nowo utworzonego node'a i po chwili powinien pojawić się status online.

Przykład poprawnego zakończenia logu uruchomienia agenta:

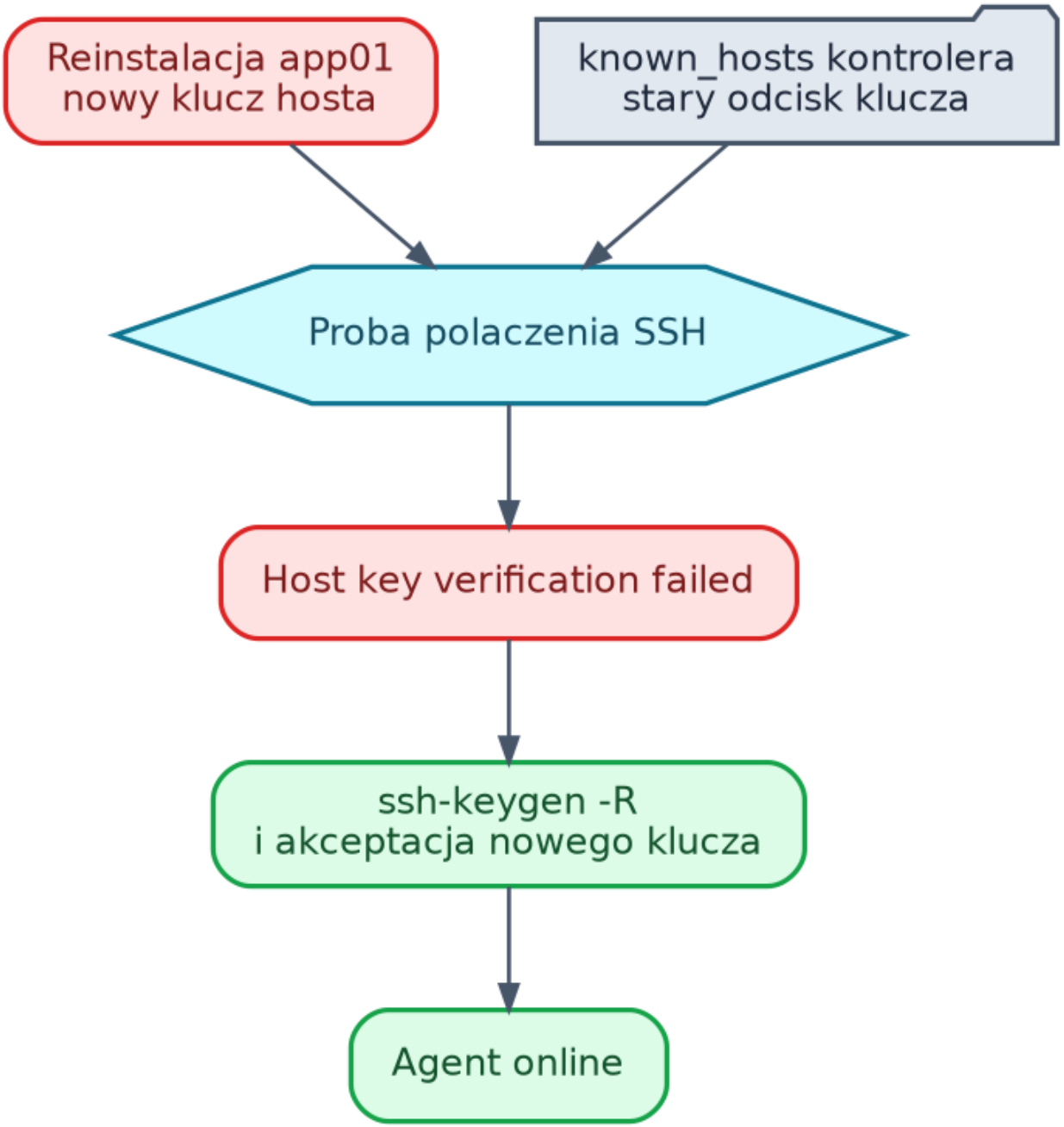
```
Remoting version: 4.5
This is a Unix agent
Agent successfully connected and online
```

Agent w kontenerze

Zamiast trwałej maszyny wirtualnej agent może być kontenerem tworzonym na potrzeby pojedynczego builda przez odpowiednią wtyczkę (Docker, Kubernetes) — konfiguruje się wtedy nie pojedynczy node, tylko szablon (`Cloud`), z którego Jenkins tworzy efemeryczne agenty na żądanie. Zaletą jest zawsze czyste środowisko i brak potrzeby ręcznego utrzymania maszyny; wadą — czas startu kontenera doliczony do każdego builda oraz konieczność trzymania obrazu z odpowiednim zestawem narzędzi (JDK, git, ewentualnie docker-in-docker) zawsze aktualnym.

Diagnostyka

Agent offline z powodu niezgodności klucza SSH



Reinstalacja agenta zmienia klucz hosta, stary odcisk w known_hosts kontrolera odrzuca połączenie, naprawa to usunięcie wpisu i akceptacja nowego klucza (kliknięcie otwiera pełny rozmiar)

> **Pojęcie: known_hosts kontrolera.** Kontener aplikacji Jenkins trzyma własny plik `known_hosts` pod `/var/jenkins_home/.ssh/known_hosts`, niezależny od `known_hosts` hosta, na którym kontener działa. Gdy serwer agenta zostanie przeinstalowany albo zmieni adres IP, jego klucz hosta się zmienia, a stary wpis w tym pliku powoduje odrzucenie połączenia.

Objaw w logu node'a:

```
WARNING: POSSIBLE DNS SPOOFING DETECTED!
The ED25519 host key for app01.ipa.zielona.example has changed,
and the key for the corresponding IP address has a different value.
Offending key for IP in /var/jenkins_home/.ssh/known_hosts: 19
Host key verification failed.
```

Po potwierdzeniu (`yes`) wracamy do konfiguracji node'a w Jenkinsie i klikamy „Launch agent" ponownie — status powinien przejść na online. Jeśli node tymczasowo miał ustawioną „Non verifying Verification Strategy" jako obejście, po naprawie przełączamy go z powrotem na „Known hosts file Verification Strategy" i zapisujemy — inaczej weryfikacja klucza hosta jest wyłączona na stałe, co jest ryzykiem bezpieczeństwa, nie tylko obejściem.

Brak miejsca na wolumenie agenta

Workspace kolejnych buildów i cache narzędzi budowania potrafią wypełnić dedykowany wolumen szybciej niż się wydaje.

Jeśli wolumen jest pełny, pierwszym krokiem jest wyczyszczenie workspace nieużywanych już zadań (patrz Architektura Jenkinsa: kontroler, agenci, workspace), a dopiero potem — jeśli to nie wystarcza — powiększenie wolumenu logicznego, tak jak opisuje strona Powiększanie przestrzeni dyskowej bez restartu.

Zła ścieżka JDK na agencie

Objaw w logu uruchomienia agenta:

```
[SSH] Checking java version of /home/jenkins_agent/jdk/bin/java
bash: /home/jenkins_agent/jdk/bin/java: no such file or directory
```

Jenkins próbuje uruchomić agenta interpreterem Javy wskazanym w konfiguracji node'a (Node Properties → Tool Locations → JAVA_HOME), a ta ścieżka nie istnieje na danym serwerze.

I poprawiamy pole `JAVA_HOME` w konfiguracji node'a na rzeczywistą ścieżkę zainstalowanego JDK — po zapisaniu Jenkins ponawia próbę uruchomienia agenta automatycznie.

Rozjazd wersji remoting.jar

> **Pojęcie: remoting.jar.** Biblioteka, którą kontroler kopiuje na agenta przy każdym uruchomieniu połączenia SSH i która faktycznie realizuje komunikację między obydwoma procesami. Wersja `remoting.jar` jest przywiązana do wersji kontrolera — stary plik pozostawiony na agencie z czasów innej wersji Jenkinsa bywa źródłem niejasnych błędów protokołu, mimo że samo połączenie SSH działa poprawnie.

Sprawdzenie wersji zgłaszanej przy starcie agenta w logu node'a:

```
Remoting version: 4.5
This is a Unix agent
```

Jeśli wersja w logu wygląda podejrzanie nisko w stosunku do aktualnej wersji kontrolera, usuwamy przestarzały plik z katalogu roboczego agenta, żeby wymusić świeżą kopię przy następnym uruchomieniu.

Kolejne uruchomienie node'a (przyciskiem „Launch agent") skopiuje aktualną wersję dopasowaną do kontrolera.

Częste błędy

Zmiana Host Key Verification Strategy na „Non verifying" na stałe. To ustawienie jest wygodnym obejściem na czas diagnozowania problemu z kluczem hosta, ale wyłącza sprawdzanie tożsamości serwera agenta przy każdym kolejnym połączeniu — pozostawione na stałe otwiera furtkę na atak man-in-the-middle na ruch SSH między kontrolerem a agentem.

Próba ponownego uruchomienia agenta, który wciąż jest „connected". Log zgłasza wtedy `The Agent is connected, disconnect it before to try to connect it again` — trzeba go najpierw jawnie rozłączyć z poziomu widoku node'a, zanim Jenkins zgodzi się nawiązać połączenie od nowa.

Zapominanie o wymianie kluczy SSH przed pierwszą konfiguracją node'a. Kolejność ma znaczenie — jeśli kontroler nigdy wcześniej nie łączył się z danym serwerem, pierwsza próba uruchomienia agenta z poziomu Jenkinsa kończy się błędem weryfikacji klucza, bo host nie istnieje

jeszcze w `known_hosts` kontenera.

□□ Ćwiczenia

> Zadania sprawdzające.

1. Opisujemy kolejność czynności potrzebnych do dodania nowego agenta SSH, od przygotowania dysku po pierwszy status online w Jenkinsie.
2. Agent zgłasza `Host key verification failed` po reinstalacji serwera. Wyjaśniamy, dlaczego doszło do tego błędu i jakie dwie komendy trzeba wykonać wewnątrz kontenera Jenkinsa, żeby go naprawić.
3. Wolumen `vgJenkins_Agent` jest zapełniony w 95%. Wskazujemy, co sprawdzamy najpierw, zanim sięgniemy po `lvextend`.
4. Wyjaśniamy, dlaczego zostawienie node'a na stałe z „Non verifying Verification Strategy" jest problemem bezpieczeństwa, nie tylko niedopracowaną konfiguracją.

Powiązane: Architektura Jenkinsa: kontroler, agenci, workspace · Poświadczenia i bezpieczeństwo · Utrzymanie i diagnostyka · Powiększanie przestrzeni dyskowej bez restartu

---

Revision #37
Created 2026-09-25 07:39:52 UTC by Tomasz Konieczny
Updated 2026-10-26 14:14:58 UTC by Tomasz Konieczny