Instalacja TeamCity bez Dockera

Typ: procedura **Dotyczy: TeamCity 2026.2, dystrybucja .tar.gz, Rocky Linux 9**

Czas: ~30 min

Efekt: serwer TeamCity uruchomiony jako usługa systemd na dedykowanym koncie systemowym, bez Dockera, z Javą, firewalld i katalogiem danych dopasowanymi do Rocky Linux 9.

Skróty użyte na tej stronie (4)

Skrót	Co znaczy
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
SELinux	Security-Enhanced Linux: dodatkowa warstwa uprawnień w jądrze RHEL
WWW	World Wide Web: strony i aplikacje otwierane w przeglądarce po HTTP albo HTTPS

Zanim zaczniemy

Komendy wykonujemy z CLI bash jako user z uprawnieniami sudo albo root. Serwer musi rozwiązywać nazwy DNS i mieć ruch wychodzący do `download.jetbrains.com`, żeby pobrać dystrybucję.

Wymaganie	Uwaga
Rocky Linux 9 albo RHEL 9	polecenia SELinux i firewalld poniżej dotyczą tej rodziny systemów
Java 21 (OpenJDK)	TeamCity nie ma własnej Javy w tym pakiecie, w przeciwieństwie do obrazu Dockera
Ruch wychodzący do `download.jetbrains.com`	pobranie dystrybucji `.tar.gz`
Wolny port `8111`	interfejs WWW serwera
Baza danych osiągalna z tego serwera	PostgreSQL na `pg01`, patrz Instalacja TeamCity w Dockerze po kroki założenia bazy

Kiedy wybieramy ten wariant zamiast Dockera:

Sytuacja	Docker	Bez Dockera
Mamy już standard Dockera w firmie	tak, spójnie z resztą aplikacji	dodatkowy wyjątek do utrzymania
Aktualizacja ma być podmianą jednego katalogu, bez rejestru obrazów	trzeba śledzić tagi obrazu	podmiana dystrybucji, prościej do audytu
Zespół zna dobrze systemd, SELinux i firewalld	mniej istotne, izolacja z kontenera	pełna kontrola, ale więcej ręcznej pracy

Przebieg

1. Instalacja Javy

```
sudo dnf install -y java-21-openjdk
```

TeamCity 2026.2 wymaga Javy w wersji 21: starsza Java kończy się błędem już przy pierwszym uruchomieniu skryptu startowego, nie dopiero przy pracy serwera.

2. Utworzenie konta systemowego teamcity

```
sudo useradd --system --shell /bin/bash --comment "TeamCity Server" --create-home --home-dir /var/lib/teamcity teamcity
```

Konto `teamcity` uruchamia usługę i jest właścicielem katalogu danych, bez prawa logowania hasłem.

3. Pobranie i rozpakowanie dystrybucji

```
curl -L -o TeamCity.tar.gz https://download.jetbrains.com/teamcity/TeamCity-2026.2.tar.gz
sudo mkdir -p /opt/teamcity
sudo tar -xzf TeamCity.tar.gz -C /opt/teamcity --strip-components=1
sudo chown -R teamcity:teamcity /opt/teamcity
```

Dystrybucja `.tar.gz` zawiera już dodatkowo agenta budującego, przydatnego do szybkich testów na tej samej maszynie: instalację osobnego agenta na innej maszynie opisuje strona Agenci TeamCity: dodanie, autoryzacja, pule.

4. Katalog danych

```
sudo mkdir -p /var/lib/teamcity/.BuildServer
sudo chown -R teamcity:teamcity /var/lib/teamcity
sudo chmod -R 750 /var/lib/teamcity
```

> **Pojęcie: katalog danych TeamCity.** Miejsce, w którym serwer trzyma konfigurację projektów, poświadczenia i klucz szyfrujący, niezależnie od samej instalacji programu w `/opt/teamcity`. Domyślnie to `.BuildServer` w katalogu domowym konta uruchamiającego usługę: utrata tego katalogu bez kopii zapasowej, opisanej na stronie Kopia zapasowa i aktualizacja TeamCity, oznacza utratę całej historii i konfiguracji.

5. Jednostka systemd

Plik `/etc/systemd/system/teamcity.service`:

```
[Unit]
Description=TeamCity Server
After=network.target

[Service]
Type=forking
User=teamcity
Group=teamcity
Environment=JAVA_HOME=/usr/lib/jvm/jre-21
Environment=TEAMCITY_DATA_PATH=/var/lib/teamcity/.BuildServer
ExecStart=/opt/teamcity/bin/teamcity-server.sh start
ExecStop=/opt/teamcity/bin/teamcity-server.sh stop
RestartSec=5s
Restart=on-failure

[Install]
```

```
WantedBy=multi-user.target
```

Włączamy i uruchamiamy usługę:

```
sudo systemctl daemon-reload
sudo systemctl enable --now teamcity
```

Sprawdzamy status:

```
sudo systemctl status teamcity
```

Przykład:

```
● teamcity.service - TeamCity Server
     Loaded: loaded (/etc/systemd/system/teamcity.service; enabled; preset: disabled)
     Active: active (running) since Sun 2026-09-27 11:02:10 UTC; 14s ago
```

6. SELinux i firewalld na Rocky 9

Otwieramy port WWW na firewalld:

```
sudo firewall-cmd --add-port=8111/tcp --permanent
sudo firewall-cmd --reload
```

Weryfikujemy otwarte porty:

```
sudo firewall-cmd --list-ports
```

Przykład:

```
8111/tcp
```

Sprawdzamy kontekst SELinux katalogu danych, gdyby serwer nie mógł do niego zapisywać mimo poprawnych uprawnień systemowych:

```
ls -Z /var/lib/teamcity
```

Jeśli kontekst różni się od oczekiwanego typu dla katalogów domowych, przywracamy domyślny:

```
sudo restorecon -Rv /var/lib/teamcity
```

Weryfikacja

```
curl -I http://localhost:8111
```

Przykład:

```
HTTP/1.1 200 OK
```

Sprawdzamy logi serwera, gdyby coś odbiegało od oczekiwań:

```
sudo tail -n 50 /opt/teamcity/logs/teamcity-server.log
```

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: pierwszy ekran kreatora TeamCity otwarty pod `http://ci01:8111`, ekran akceptacji licencji z widocznym numerem wersji „2026.2".

Wycofanie zmian

```
sudo systemctl disable --now teamcity
sudo rm -f /etc/systemd/system/teamcity.service
sudo systemctl daemon-reload
sudo rm -rf /opt/teamcity /var/lib/teamcity
sudo userdel -r teamcity
```

Gdy coś nie zadziała

Objaw	Przyczyna	Naprawa
`systemctl status teamcity` pokazuje `failed` zaraz po starcie	zła wersja Javy albo brak `JAVA_HOME` widocznego dla usługi	`journalctl -u teamcity`, sprawdzamy `java -version` na koncie `teamcity`
`curl -I http://localhost:8111` zwraca `Connection refused`, mimo że usługa działa	firewalld blokuje port	`firewall-cmd --list-ports`, dodajemy `8111/tcp`
Serwer startuje, ale zapisuje dane w nieoczekiwanym katalogu	`TEAMCITY_DATA_PATH` nieustawiony albo inny katalog domowy konta `teamcity`	sprawdzamy zmienną w jednostce systemd i katalog domowy z `getent passwd teamcity`

Objaw	Przyczyna	Naprawa
Błąd `Permission denied` przy zapisie do `.BuildServer` mimo poprawnego `chown`	kontekst SELinux katalogu niezgodny z oczekiwanym	`ls -Z /var/lib/teamcity`, `restorecon -Rv /var/lib/teamcity`

Powiązane: Instalacja TeamCity w Dockerze · Agenci TeamCity: dodanie, autoryzacja, pule · Instalacja Gitea bez Dockera

Źródła: jetbrains.com/help/teamcity, rozdział „Install TeamCity Server on Linux or macOS" · jetbrains.com/help/teamcity, rozdział „Start TeamCity Server" · jetbrains.com/help/teamcity, „Cheatsheet: Updating Java on Server and Agent Machines" · `man 8 firewall-cmd`

Revision #31
Created 2026-09-27 11:19:33 UTC by Tomasz Konieczny
Updated 2026-10-26 14:17:43 UTC by Tomasz Konieczny