Да, **использовать Selenium на сервере Яндекс Cloud (MyCloud)** **можно**, и это вполне рабочий вариант. Но нужно правильно настроить окружение — особенно если ты используешь **Ubuntu-сервер без графического интерфейса (GUI)**, как это обычно бывает на виртуальных машинах (ВМ) в облаке.

**✅ Что ты получаешь в MyCloud**

Если ты работаешь с **ВМ на базе Ubuntu/Debian**, у тебя есть обычный Linux-сервер, на который можно установить:

* Google Chrome или Chromium
* ChromeDriver
* Python с библиотекой selenium

**🔧 Пошаговая инструкция: установка Selenium на сервере Яндекс Cloud**

**🔹 1. Установи Python, pip и виртуальное окружение (если ещё нет)**

sudo apt update

sudo apt install -y python3 python3-pip python3-venv

python3 -m venv venv

source venv/bin/activate

**🔹 2. Установи Google Chrome и ChromeDriver**

**✅ Установка Google Chrome:**

wget https://dl.google.com/linux/direct/google-chrome-stable\_current\_amd64.deb

sudo apt install -y ./google-chrome-stable\_current\_amd64.deb

**✅ Установка ChromeDriver:**

Узнай текущую версию Chrome:

google-chrome --version

Затем скачай **соответствующую версию ChromeDriver** отсюда:  
👉 <https://googlechromelabs.github.io/chrome-for-testing/>

Пример:

wget https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/123.0.6312.86/linux64/chromedriver-linux64.zip

unzip chromedriver-linux64.zip

chmod +x chromedriver-linux64/chromedriver

sudo mv chromedriver-linux64/chromedriver /usr/local/bin/chromedriver

**🔹 3. Установи Python-библиотеки**

pip install selenium

**🔹 4. Пиши Python-скрипт с headless Chrome**

Пример main.py:

from selenium import webdriver

from selenium.webdriver.chrome.options import Options

options = Options()

options.add\_argument("--headless") # без графического интерфейса

options.add\_argument("--no-sandbox")

options.add\_argument("--disable-dev-shm-usage")

driver = webdriver.Chrome(options=options)

driver.get("https://example.com")

print(driver.title)

driver.quit()

**🔹 5. Запусти скрипт:**

python main.py

**✅ Проверка успешности**

Если ты видишь:

Example Domain

…значит Selenium успешно отработал в headless-режиме.

**⚠️ Возможные ошибки и их решения**

| **Ошибка** | **Причина** | **Решение** |
| --- | --- | --- |
| selenium.common.exceptions.WebDriverException | Нет chromedriver или он не совпадает с версией Chrome | Убедись, что Chrome и ChromeDriver одной версии |
| cannot open display | Нет GUI | Обязательно использовать --headless |
| DevToolsActivePort | Браузер не запущен в sandbox-режиме | Добавь --no-sandbox, --disable-dev-shm-usage |

**✅ Итог**

Selenium отлично работает на сервере в Яндекс Cloud, **если ты установишь headless Chrome + ChromeDriver**, и **настроишь браузер без GUI**.

Хочешь — могу помочь собрать Docker-файл или bash-скрипт для быстрой установки всего стека. Продолжим?

Да, **можно использовать selenium-stealth** на сервере, включая Яндекс Cloud (MyCloud) — это популярная библиотека, которая помогает **маскировать автоматический запуск браузера** от систем антиботов.

**📌 Что такое selenium-stealth?**

Это расширение для Selenium, которое **меняет поведение браузера**, чтобы он выглядел как обычный пользовательский. Особенно полезно при парсинге сайтов, защищённых антиботами (например, Cloudflare, Distil, бот-детекторы JS).

**✅ Как использовать selenium-stealth на сервере (например, Ubuntu в MyCloud)**

**🔧 Пошаговая инструкция:**

**🔹 1. Установи всё как для обычного headless Chrome:**

Смотри [предыдущий ответ](https://platform.openai.com/), если ты этого ещё не сделал:

* Google Chrome
* ChromeDriver
* Python
* pip install selenium

**🔹 2. Установи selenium-stealth**

pip install selenium-stealth

**🔹 3. Пример рабочего скрипта с selenium-stealth в headless-режиме:**

from selenium import webdriver

from selenium.webdriver.chrome.options import Options

from selenium\_stealth import stealth

options = Options()

options.add\_argument("--headless") # для серверов без GUI

options.add\_argument("--no-sandbox")

options.add\_argument("--disable-dev-shm-usage")

options.add\_argument("--disable-blink-features=AutomationControlled")

driver = webdriver.Chrome(options=options)

# Применяем selenium-stealth

stealth(driver,

languages=["en-US", "en"],

vendor="Google Inc.",

platform="Win32",

webgl\_vendor="Intel Inc.",

renderer="Intel Iris OpenGL Engine",

fix\_hairline=True,

)

driver.get("https://httpbin.org/headers")

print(driver.page\_source)

driver.quit()

**✅ Что делает selenium-stealth:**

* Удаляет navigator.webdriver = true
* Подменяет заголовки navigator.platform, navigator.languages, navigator.vendor
* Меняет поведение JS API WebGL, hairline detection и др.
* Имитирует поведение настоящего браузера

**⚠️ Замечания при запуске на сервере:**

| **Проблема** | **Решение** |
| --- | --- |
| Браузер крашится на старте | Убедись в наличии --headless --no-sandbox |
| DevToolsActivePort ошибка | Добавь --disable-dev-shm-usage |
| Не находит chromedriver | Убедись, что он в $PATH или явно указать путь |

**💡 Альтернатива: undetected-chromedriver**

Если selenium-stealth не помогает (некоторые сайты всё ещё блокируют), используй:

pip install undetected-chromedriver

Пример:

import undetected\_chromedriver as uc

driver = uc.Chrome(headless=True)

driver.get("https://example.com")

print(driver.page\_source)

driver.quit()

**✅ Вывод:**

Да, selenium-stealth работает на сервере (в том числе Яндекс Cloud). Просто запускай Chrome в headless-режиме, устанавливай selenium-stealth, и твой браузер будет выглядеть более "натурально".

Если хочешь — могу помочь собрать скрипт установки всего окружения или Docker-контейнер. Продолжим?