

# Homework\_Lesson15\_Report

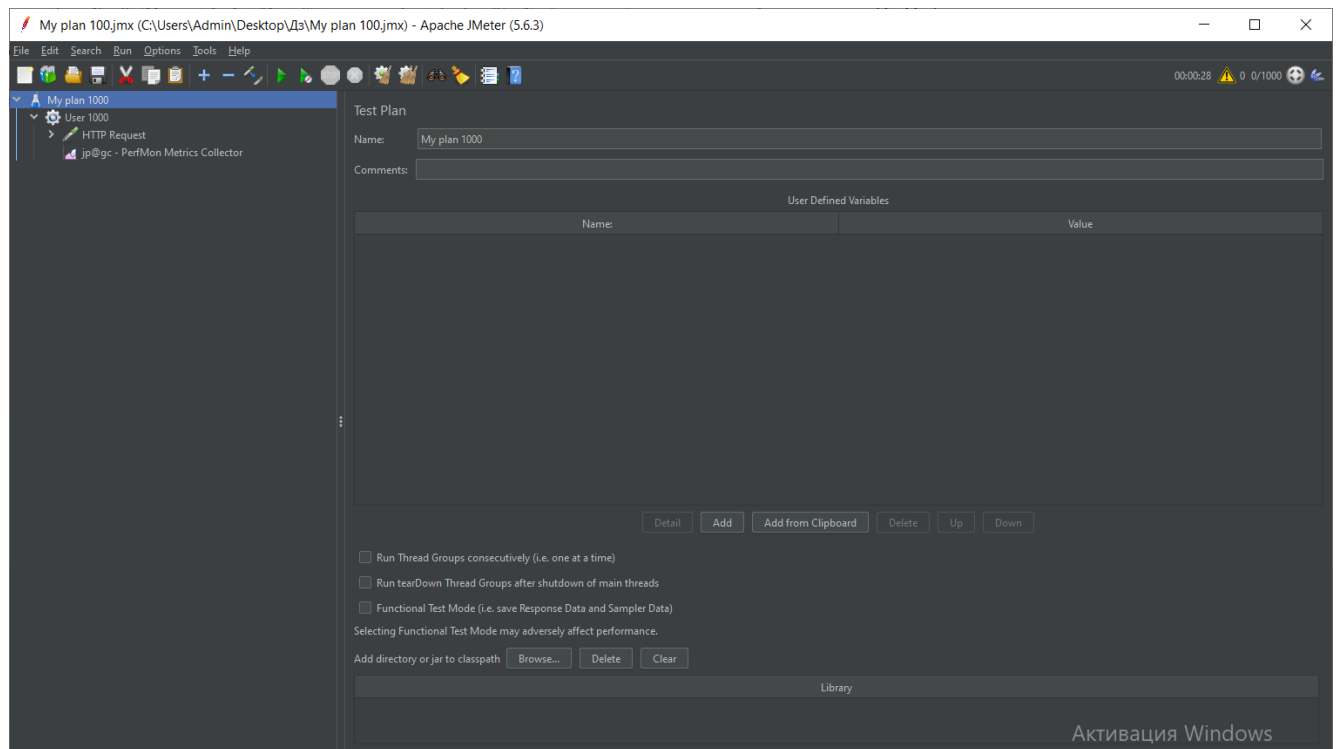
## Домашнее задание No15

Цель: выполнить тестирование производительности вашего веб- сервера, чтобы определить, как он работает при большом объеме запросов и нагрузке.

Задания:

1. Установите и настройте инструменты тестирования производительности, такие как Apache JMeter или Gatling. (лучше JMeter)
2. Определите основные метрики производительности, которые нужно измерить, например, время отклика, задержку, пропускную способность и загрузку процессора.
3. Создайте тестовые сценарии: просмотр страницы, с наличием определённого слова. Протестировать на 100,1'000,10'000 подключений.
4. Задайте параметры для тестовых сценариев, например, количество пользователей, количество запросов в единицу времени, длительность теста и т.д.
5. Запустите тест и отслеживайте производительность сервера в режиме реального времени, собирая данные о времени отклика, задержке, пропускной способности и загрузке процессора при различных нагрузках.
- \*\*\* 6. Проанализируйте результаты тестирования, чтобы определить, как сервер работает при различных нагрузках. Определите максимальную нагрузку, которую сервер может выдержать, и выявите узкие места в инфраструктуре.

## Установим JMeter



Определите основные метрики:

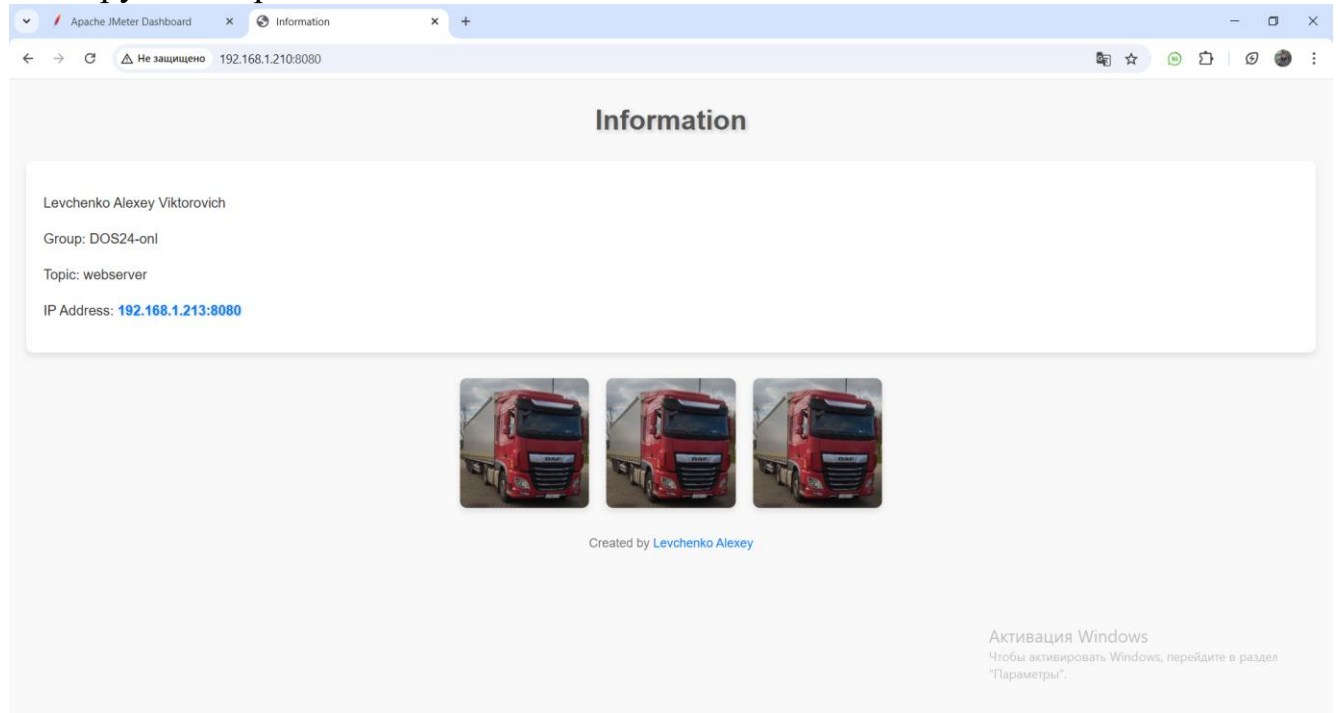
Время отклика (Response Time) - Среднее 900 мс, Максимальная 1300 мс;

Пропускная способность (Throughput) - 400 запросов в сек;

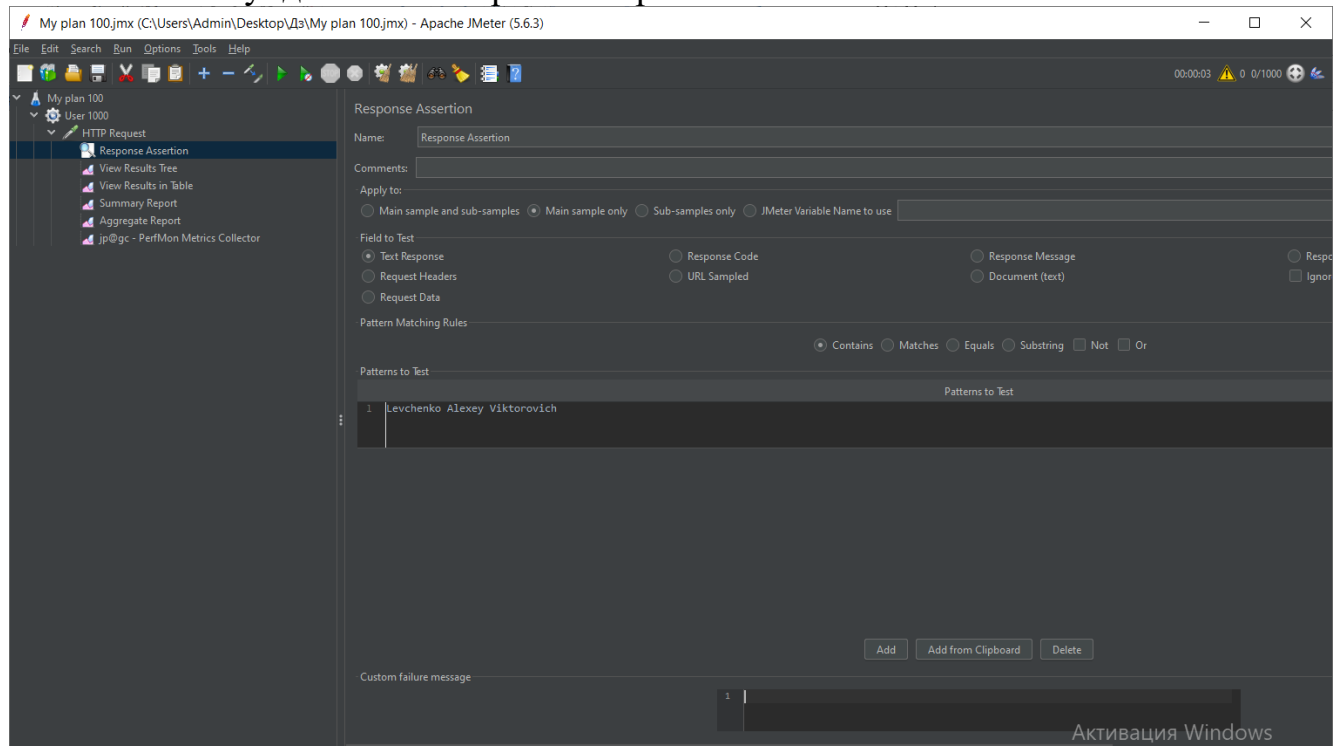
Ошибки (Error Rate) - 0,5 %;

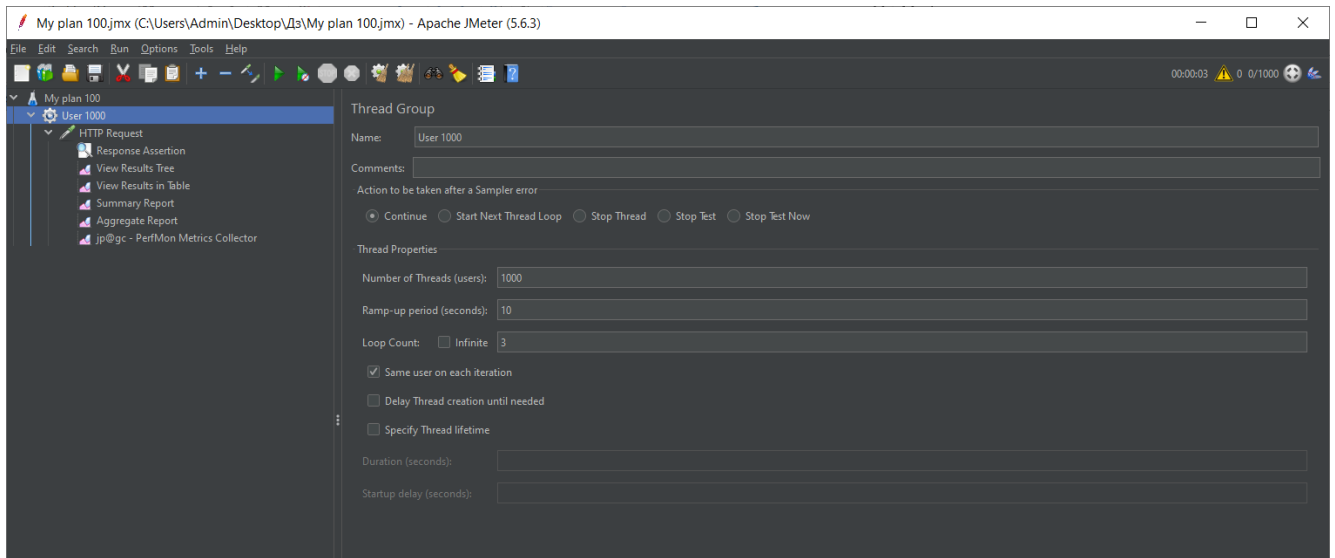
Загрузка процессора (Cpu load) - 80 %.

## Тестируемая страничка

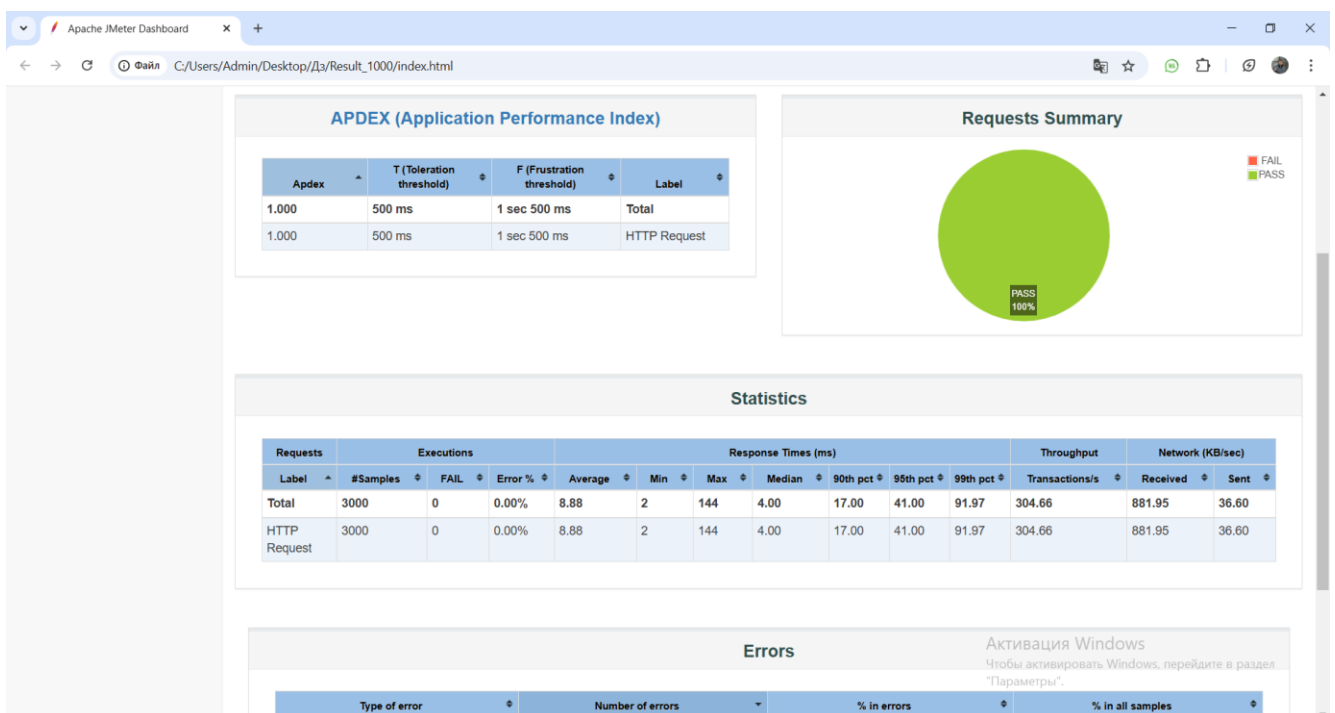
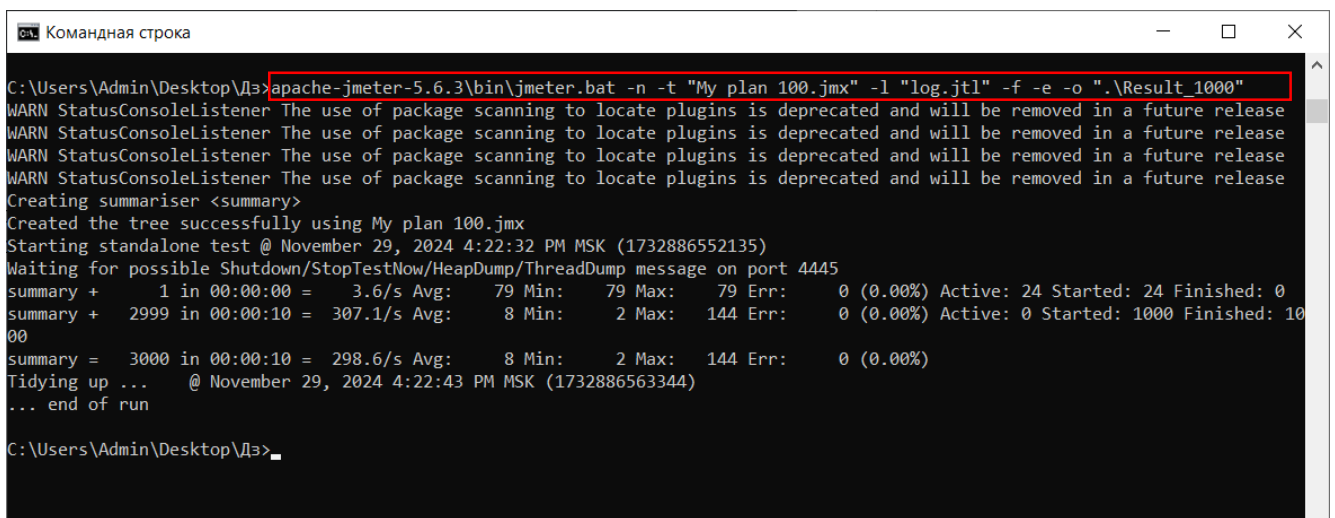


Создаем тестовый сценарий с наличием слова Levchenko Alexey Viktorovich. Для 100, 1000, 10 000. Также зададим параметры что бы пользователи подключались в течении 10 секунд и это повторялось 3 раза.



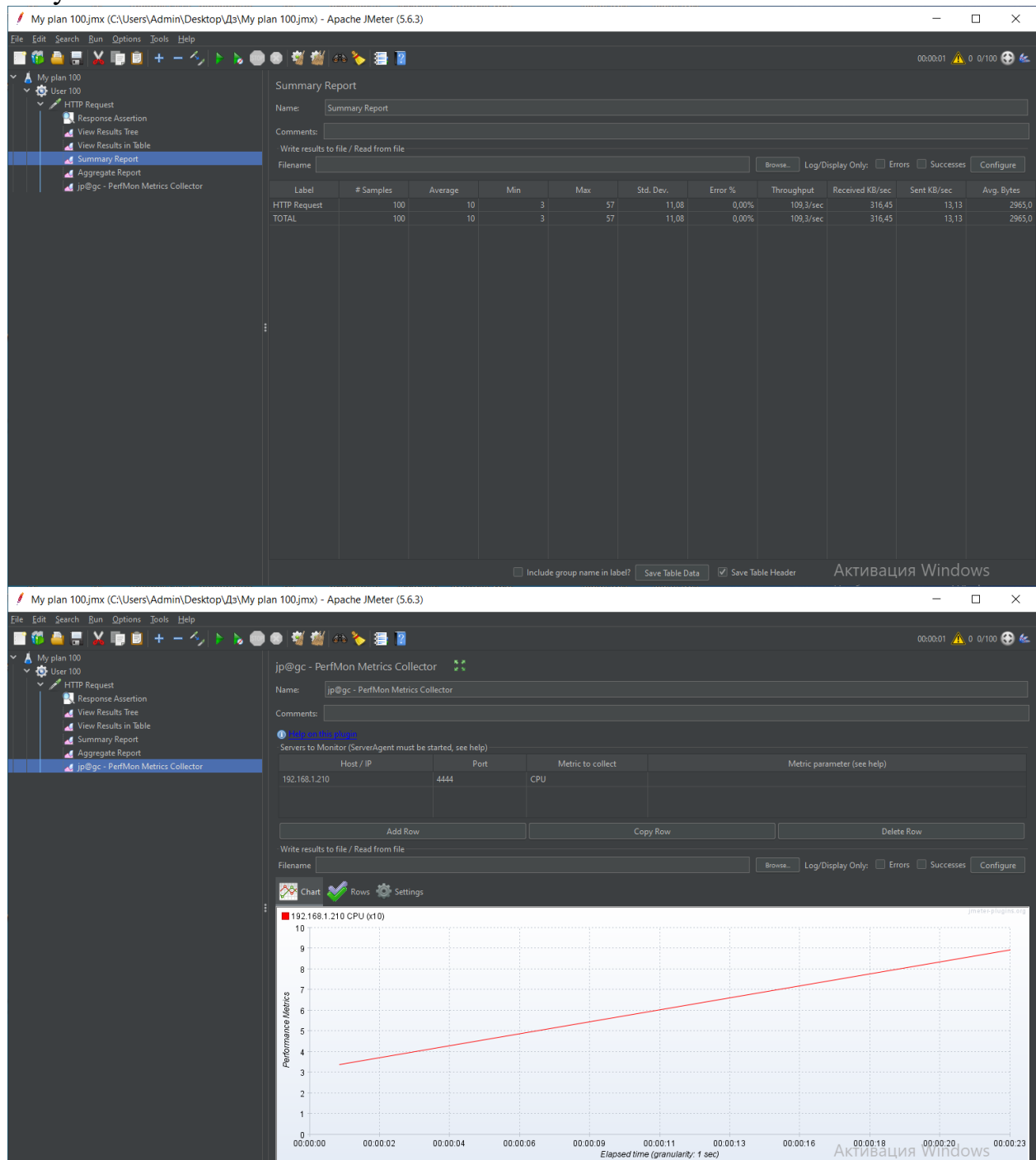


Также запустим наш тест через командную строку. И посмотрим отчет который нам сформируется в html формате.

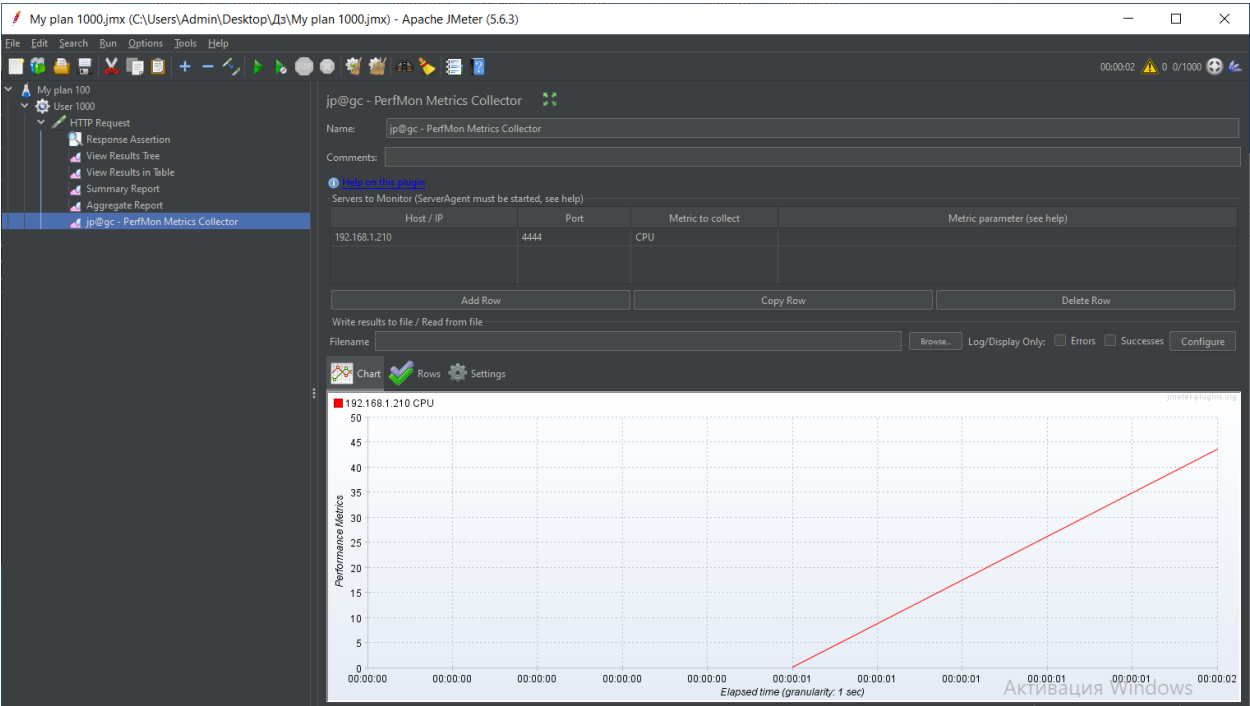
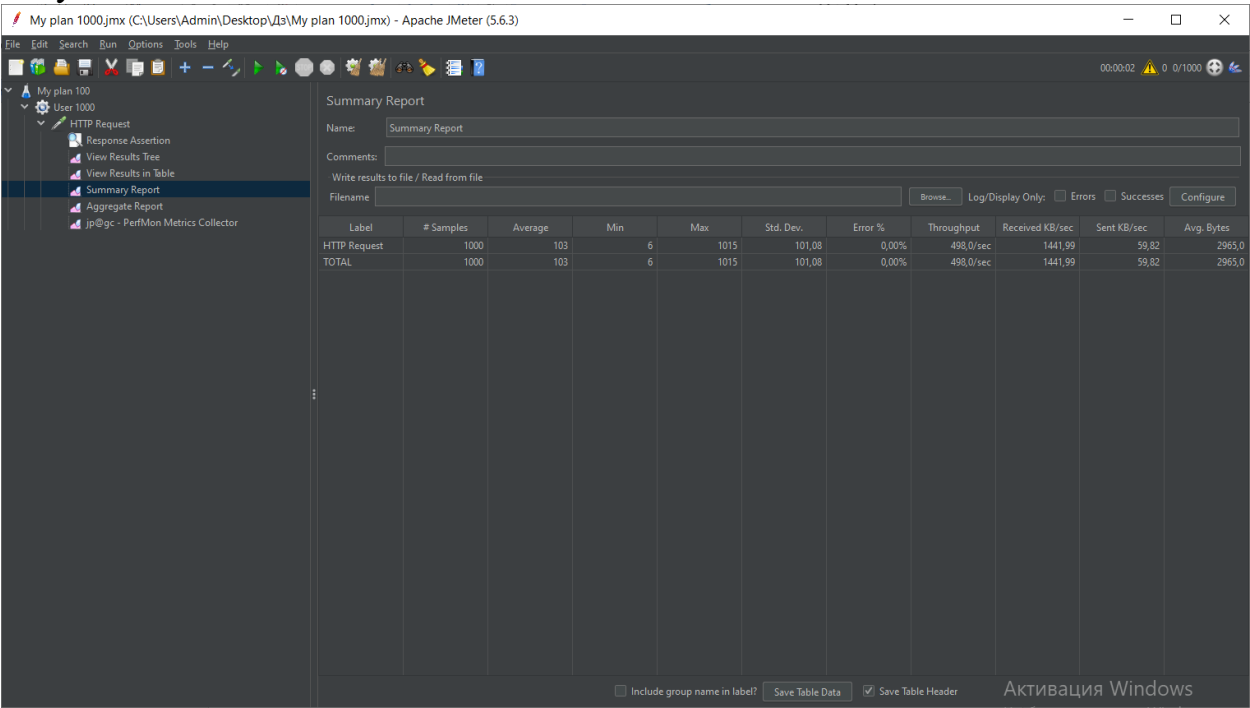


Для получения данных сру установим плагин PerfMon Metrics Collector. Перед установкой нам необходимо скачать и закинуть в папку с программой по пути lib/ext plugins-manager.jar. После этого мы сможем ставить дополнительные плагины в jmeter из графического интерфейса. Так же нам нужно скачать <https://jmeter-plugins.org/wiki/PerfMon/> и запустить сервер агента на тестируемой машине. После этого мы сможем получать в jmeter многие параметры такие как загрузка сру, memory, swap, disks i/o.

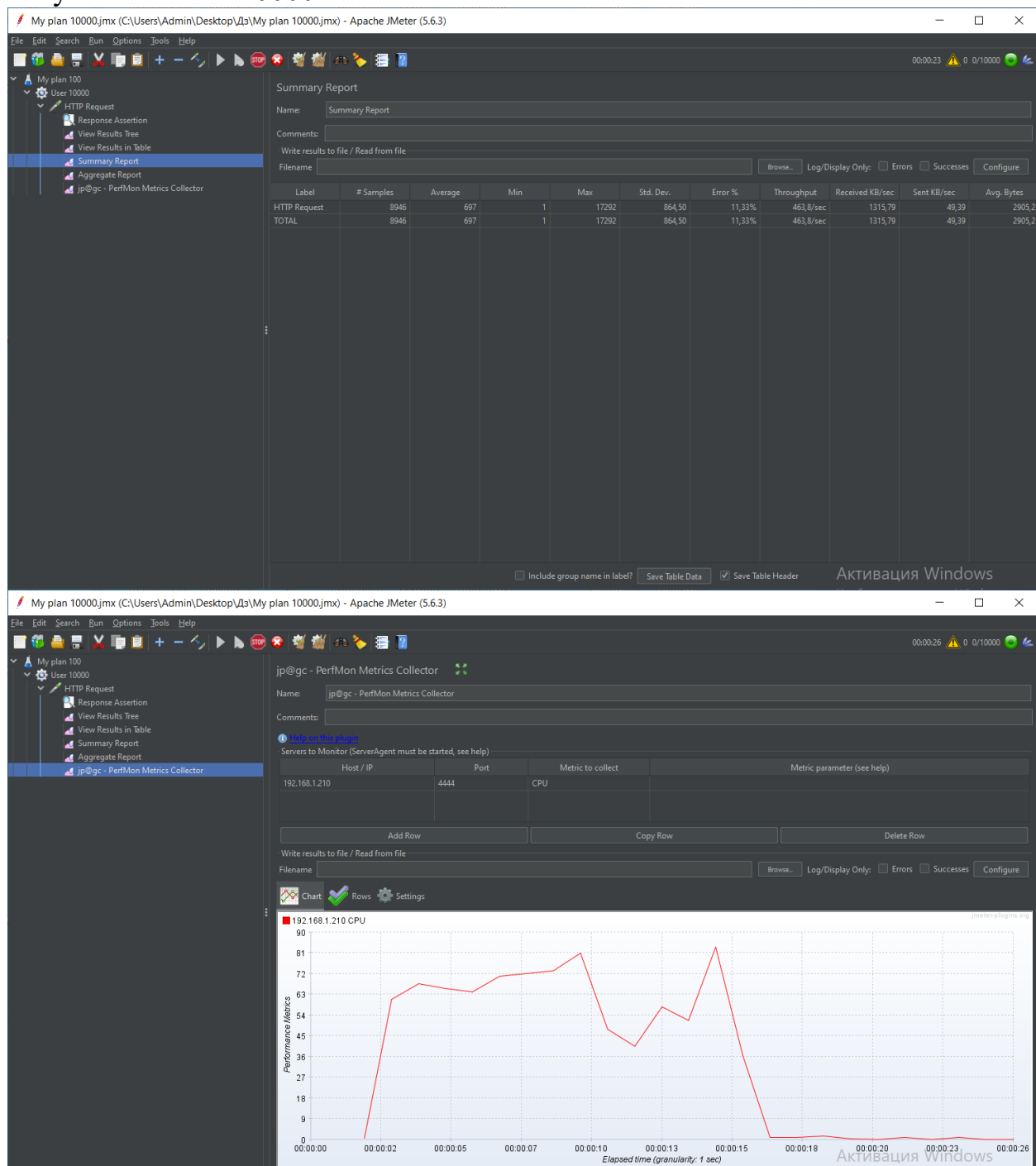
Запускаем тест на 100 пользователей.



# Запустим тест на 1000 пользователей



## Запустим тест на 10000



Приступим к подведению итогов:

100 пользователей

Время отклика (Response Time) - Среднее 8.8 мс, Максимальная 144 мс;

Пропускная способность (Throughput) – 304,6 запросов в сек;

Ошибки (Error Rate) - 0 %;

Загрузка процессора (Cpu load) - 9 %.

При 100 пользователях система работает почти идеально.

1000 пользователей

Время отклика (Response Time) - Среднее 103 мс, Максимальная 1015 мс;

Пропускная способность (Throughput) – 498 запросов в сек;

Ошибки (Error Rate) - 0 %;

Загрузка процессора (Cpu load) - 49 %.

При 1000 пользователях система демонстрирует высокую производительность и стабильность.

10000 пользователей

Время отклика (Response Time) - Среднее 697 мс, Максимальная 17292 мс;

Пропускная способность (Throughput) – 463 запросов в сек;

Ошибки (Error Rate) – 11,33 %;

Загрузка процессора (Cpu load) - 82 %.

При 10 000 пользователях система перегружается. Основные проблемы — рост времени отклика, высокий уровень ошибок и приближение загрузки CPU к критической.

#### **Узкие места:**

##### **1. Рост времени отклика при высокой нагрузке:**

Среднее и максимальное время отклика значительно увеличиваются при росте числа пользователей. Это может быть связано с таймаутами из-за долгих операций и ограниченной пропускной способностью сети.

##### **2. Ошибки при 10 000 пользователях:**

Ошибки появляются только при пиковых нагрузках. Возможные причины могут быть тайм-ауты из-за долгих операций, переполнение очередей обработки запросов.

##### **3. Загрузка CPU:**

При 10 000 пользователях CPU достигает 82 %, что близко к критической отметке. Это может указывать на недостаточное количество выделенных ресурсов для обработки такого количества запросов.