

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б / -2021
	Екземпляр № 1	Арк 11 / 1

Лабораторна робота № 5. Створення CI/CD конвеєра з використанням Jenkins

Метою даної лабораторної роботи є отримання базових навичок по роботі з Jenkins.

Завдання на лабораторну роботу

- Запустити DEVASC VM
- Створити commit для Sample app на Git
- Замінити Sample App та надіслати зміни на Git
- Завантажити та запустити образ Jenkins Docker
- Налаштувати Jenkins
- Використати Jenkins для запуску на виконання збірки застосунку
- Використати Jenkins для тестування збірки
- Створити конвеєр в Jenkins.

Хід роботи:

1. Запустити DEVASC VM.

2. Створити commit для Sample App на Git

2.1 Увійдіть до GitHub та створіть новий репозиторій.

- a. Увійдіть за адресою <https://github.com/> зі своїми обліковими даними.
- b. Натисніть кнопку **New repository** або натисніть на піктограму «+» у верхньому правому куті та виберіть **New repository**.
- c. Створіть репозиторій, використовуючи наступну інформацію:
 - Назва репозиторію: **sample-app**
 - Опис: **Explore CI/CD with GitHub and Jenkins**
 - Public/Private: **Private**
- d. Виберіть: **Create repository**

2.2 Налаштуйте свої облікові дані Git локально у віртуальній машині.

Відкрийте вікно терміналу з **VS Code** на віртуальній машині DEVASC. Використайте своє ім'я замість "Sample User" для імені в лапках " ". Використайте @example.com для вашої адреси електронної пошти.

```
devasc@labvm:~$ git config --global user.name "Sample User"
```

```
devasc@labvm:~$ git config --global user.email sample@example.com
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 2

2.3 Ініціалізуйте каталог як репозиторій Git.

Ви будете використовувати файли `sample-app`, створені вами в попередній лабораторній роботі. Крім того, ці файли для вашої зручності зберігаються в каталозі `/labs/devnet-src/jenkins/sample-app`. Перейдіть до каталогу `jenkins/sample-app` та ініціалізуйте його як репозиторій Git.

```
devasc@labvm:~$ cd labs/devnet-src/jenkins/sample-app/  
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git init  
Initialized empty Git repository in /home/devasc/labs/devnet-src/jenkins/sample-app/.git/  
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

2.4 Оберіть тип репозиторію Git, вказавши на репозиторій GitHub.

Використайте команду `git remote add`, щоб додати URL-адресу Git із віддаленим псевдонімом “origin” та вкажіть на щойно створений репозиторій GitHub. Використовуючи URL-адресу репозиторію Git, який ви створили на Кроці 1, вам потрібно буде лише замінити `github-username` в наступній команді вашим іменем користувача GitHub.

Примітка: Ваше ім'я користувача GitHub з врахуванням реєстру

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git remote add origin  
https://github.com/github-username/sample-app.git  
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

2.5 Проіндексуйте, створіть коміт та надішліть файли `sample-app` до репозиторію GitHub.

а. Використовуйте команду `git add` для розміщення індексованих файлів в каталозі `jenkins/sample-app`. Використовуйте аргумент зірочка (*) для індексування всіх файлів у поточному каталозі.

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git add *
```

б. Використовуйте команду `git status`, щоб переглянути файли та каталоги, які індексовані та підготовлені до коміту у вашому репозиторії GitHub.

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git status  
On branch master  
  
No commits yet
```

Changes to be committed:

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 3

```
(use "git rm --cached <file>..." to unstage)
new file:   sample-app.sh
new file:   sample_app.py
new file:   static/style.css
new file:   templates/index.html
```

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

с. Використайте команду **git commit**, щоб зафіксувати індексовані файли та розпочати відстеження змін. Додайте повідомлення на ваш вибір або використайте таке, як тут запропоновано.

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git commit -m "Committing
sample-app files."
[master 4030ab6] Committing sample-app files
4 files changed, 46 insertions(+)
create mode 100644 sample-app.sh
create mode 100644 sample_app.py
create mode 100644 static/style.css
create mode 100644 templates/index.html
```

д. Використайте команду **git push**, щоб перемістити ваші локальні файли sample-app до вашого репозиторію GitHub.

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git push origin master
Username for 'https://github.com': username
Password for 'https://AllJohns@github.com': password
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (8/8), 1.05 KiB | 1.05 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0)
To https://github.com/AllJohns/sample-app.git
   d0eel4a..4030ab6 master -> master
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

Примітка: Якщо замість запиту на ваше ім'я користувача ви отримуєте від VS Code повідомлення:

The extension 'Git' wants to sign in using GitHub, отже, ви неправильно налаштували свої облікові

дані GitHub на Кроці 2 та/або URL-адресу GitHub на Кроці 4. URL-адреса повинна містити правильне ім'я користувача з урахуванням регістру та ім'я репозиторію, який ви створили на кроці 1. Для того, щоб скасувати попередню

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 4

команду **git add**, використовуйте команду **git remote rm origin**. Потім поверніться до Кроку 2, щоб переконатись в тому, що ви ввели правильні облікові дані, а на Кроці 4 - правильну URL-адресу.

Примітка: Якщо після введення імені користувача та пароля ви отримаєте помилку - «fatal error», в якій зазначено, що репозиторій не знайдений, то скоріше за все, ви надіслали неправильну URL-адресу. Вам потрібно буде скасувати свою команду **git add** за допомогою команди **git remote rm origin**

3: Змінити Sample App та надіслати зміни на Git

3.1 Відкрийте файли sample-app

Переконайтесь, що ви все ще перебуваєте в каталозі `~/labs/devnet-src/jenkins/sample-app`, оскільки ці файли, пов'язані з вашим репозиторієм GitHub. Відкрийте для редагування як `sample_app.py`, так і `sample-app.sh`.

3.2: Відредагуйте файли sample-app

а. В `sample_app.py` змініть один екземпляр порту 8080 на 5050, як це показано нижче.

```
from flask import Flask
from flask import request
from flask import render_template

sample = Flask(__name__)

@sample.route("/")
def main():
    return render_template("index.html")

if __name__ == "__main__":
    sample.run(host="0.0.0.0", port=5050)
```

б. В `sample-app.sh` змініть три екземпляри порту 8080 на 5050, як це показано нижче.

```
#!/bin/bash

mkdir tempdir
mkdir tempdir/templates
mkdir tempdir/static

cp sample_app.py tempdir/.
cp -r templates/* tempdir/templates/.
cp -r static/* tempdir/static/.
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 5

```

echo "FROM python" >> tempdir/Dockerfile
echo "RUN pip install flask" >> tempdir/Dockerfile
echo "COPY ./static /home/myapp/static/" >> tempdir/Dockerfile
echo "COPY ./templates /home/myapp/templates/" >> tempdir/Dockerfile
echo "COPY sample_app.py /home/myapp/" >> tempdir/Dockerfile
echo "EXPOSE 5050" >> tempdir/Dockerfile
echo "CMD python3 /home/myapp/sample_app.py" >> tempdir/Dockerfile

cd tempdir
docker build -t sampleapp .

docker run -t -d -p 5050:5050 --name samplerunning sampleapp
docker ps -a

```

3.3: Зберіть і перевірте sample-app.

а. Введіть команду **bash** для того, щоб створити свій застосунок, який використовуватиме новий порт 5050.

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ bash ./sample-app.sh
```

```
Sending build context to Docker daemon 6.144kB
```

```
Step 1/7 : FROM python
```

```
---> 4f7cd4269fa9
```

```
Step 2/7 : RUN pip install flask
```

```
---> Using cache
```

```
---> 57a74c0dff93
```

```
Step 3/7 : COPY ./static /home/myapp/static/
```

```
---> Using cache
```

```
---> e70310436097
```

```
Step 4/7 : COPY ./templates /home/myapp/templates/
```

```
---> Using cache
```

```
---> e41ed6d0f933
```

```
Step 5/7 : COPY sample_app.py /home/myapp/
```

```
---> 0a8d152f78fd
```

```
Step 6/7 : EXPOSE 5050
```

```
---> Running in d68f6bfbcbfb
```

```
Removing intermediate container d68f6bfbcbfb
```

```
---> 04fa04alc3d7
```

```
Step 7/7 : CMD python3 /home/myapp/sample_app.py
```

```
---> Running in ed48fdb031b
```

```
Removing intermediate container ed48fdb031b
```

```
---> ec9f34fa98fe
```

```
Successfully built ec9f34fa98fe
```

```
Successfully tagged sampleapp:latest
```

```
d957a4094c1781ccd7d86977908f5419a32c05a2a1591943bb44eeb8271c02dc
```

```
d957a4094c1781ccd7d86977908f5419a32c05a2a1591943bb44eeb8271c02dc
```

```
CONTAINER ID          IMAGE                COMMAND              CREATED
```

```
STATUS                PORTS              NAMES
```

```
d957a4094c17          sampleapp          "/bin/sh -c 'python ..." 1 second ago
```

```
Up Less than a second 0.0.0.0:5050->5050/tcp samplerunning
```

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 6

б. Відкрийте вкладку браузера та перейдіть до localhost:5050. Ви маєте побачити повідомлення **You are calling me from 172.17.0.1**.

с. Вимкніть сервер, як тільки переконаєтеся в тому, що він працює на порту 5050. Поверніться до вікна терміналу, в якому працює сервер, і натисніть CTRL+C, щоб зупинити сервер.

3.4 Надішліть свої зміни до GitHub.

а. Тепер ви готові надіслати свої зміни до вашого репозиторію GitHub. Введіть наступні команди:

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git add *
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   sample-app.sh
    modified:   sample_app.py
    new file:   tempdir/Dockerfile
    new file:   tempdir/sample_app.py
    new file:   tempdir/static/style.css
    new file:   tempdir/templates/index.html

devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git commit -m "Changed
port from 8080 to 5050."
[master 98d9b2f] Changed port from 8080 to 5050.
 6 files changed, 33 insertions(+), 3 deletions(-)
 create mode 100644 tempdir/Dockerfile
 create mode 100644 tempdir/sample_app.py
 create mode 100644 tempdir/static/style.css
 create mode 100644 tempdir/templates/index.html
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ git push origin master
Username for 'https://github.com': username
Password for 'https://AllJohns@github.com': password
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 748 bytes | 748.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/AllJohns/sample-app.git
   a6b6b83..98d9b2f master -> master
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

б. Ви можете переконаватися в тому, що ваш репозиторій GitHub оновлений, якщо переглянете посилання <https://github.com/github-user/sample-app>. Ви побачите своє нове повідомлення «Changed port from 8080 to 5050.», а також

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 7

оновлену останню мітку часу коміту.

4. Завантажити та запустити образ Jenkins Docker

4.1 Завантажте образ Jenkins Docker.

Образ Jenkins Docker зберігається за посиланням: <https://hub.docker.com/r/jenkins/jenkins>. На момент написання даної лабораторної роботи на цьому веб-сайті вказано, що ви використовуєте команду **docker pull jenkins jenkins** для завантаження останньої версії контейнера Jenkins. Ви маєте отримати результат, подібний до наступного:

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker pull
jenkins/jenkins:lts
lts: Pulling from jenkins/jenkins
3192219afd04: Pulling fs layer
17c160265e75: Pulling fs layer
cc4fe40d0e61: Pulling fs layer
9d647f502a07: Pulling fs layer
d108b8c498aa: Pulling fs layer
1bfe918b8aa5: Pull complete
dafala7c0751: Pull complete
650a236d0150: Pull complete
cba44e30780e: Pull complete
52e2f7d12a4d: Pull complete
d642af5920ea: Pull complete
e65796f9919e: Pull complete
9138dabbc5cc: Pull complete
f6289c08656c: Pull complete
73d6b450f95c: Pull complete
a8f96fbec6a5: Pull complete
9b49calb4e3f: Pull complete
d9c8f6503715: Pull complete
20fe25b7b8af: Pull complete
Digest: sha256:717dcbe5920753187a20ba43058ffd3d87647fa903d98cde64dda4f4c82c5c48
Status: Downloaded newer image for jenkins/jenkins:lts
docker.io/jenkins/jenkins:lts
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

4.2 Запустіть контейнер Jenkins Docker.

Введіть наступну команду в один рядок. Можливо, вам доведеться скопіювати її в текстовий редактор у випадку, якщо ви переглядаєте PDF-версію цієї лабораторної роботи для того, щоб уникнути розривів рядків. Ця команда запустить контейнер Jenkins Docker, а потім дозволить виконувати команди Docker

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 8

на вашому сервері Jenkins.

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker run --rm -u root -p
8080:8080 -v jenkins-data:/var/jenkins_home -v $(which
docker):/usr/bin/docker -v /var/run/docker.sock:/var/run/docker.sock -v
"$HOME":/home --name jenkins_server jenkins/jenkins:lts
```

В цій команді запуску **docker run**, використовуються наступні параметри:

- **--rm** - Даний параметр автоматично видаляє контейнер Docker після закінчення його запуску.
- **-u** - Даний параметр визначає користувача. Ви хочете, щоб цей контейнер Docker запускався як root, щоб усі команди Docker, введені в сервер Jenkins, були дозволеними.
- **-p** - Даний параметр визначає порт, на якому сервер Jenkins буде працювати локально.
- **-v** - Дані параметри зв'язують монтовані томи, які необхідні для Jenkins і Docker. Перший **-v** визначає, де будуть зберігатися дані Jenkins. Другий **-v** визначає, де взяти Docker, щоб ви могли запустити Docker всередині контейнера Docker, на якому запуснений сервер Jenkins. Третій **-v** визначає змінну PATH для домашнього каталогу.

4.3 Переконайтеся, що сервер Jenkins запущений.

Тепер сервер Jenkins має працювати. Скопіюйте пароль адміністратора, який відображається у вихідних даних, як показано нижче.

Не вводьте жодних команд у цьому вікні сервера. Якщо ви випадково зупините сервер Jenkins, вам необхідно буде повторно ввести команду **docker run** з Кроку 2, наведеного вище. Після початкового встановлення, пароль адміністратора відображається як показано нижче.

<output omitted>

```
*****
*****
```

Jenkins initial setup is required. An admin user has been created and a password generated.

Please use the following password to proceed to installation:

```
77dc402e31324c1b917f230af7bfebf2 <--Your password will be different
```

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

```
*****
*****
```

<output omitted>

```
2020-05-12 16:34:29.608+0000 [id=19] INFO hudson.WebAppMain$3#run: Jenkins is fully up
and running
```

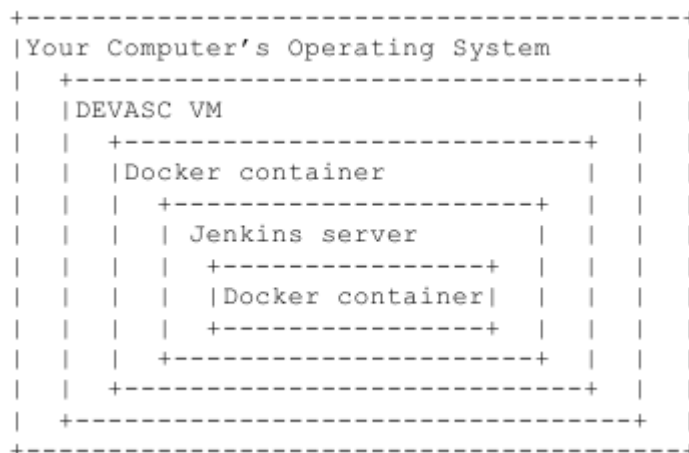

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 9

Примітка: Якщо ви загубите пароль, або він не відображається, як показано вище, або вам потрібно перезапустити сервер Jenkins, ви завжди зможете отримати пароль, відкривши командний рядок контейнера Jenkins Docker. Створіть друге вікно терміналу в VS Code та введіть наступні команди, щоб не зупиняти сервер Jenkins:

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker exec -it
jenkins_server /bin/bash
root@19d2a847a54e:/# cat /var/jenkins_home/secrets/initialAdminPassword
77dc402e31324c1b917f230af7bfebf2
root@19d2a847a54e:/# exit
exit
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$
```

Примітка: Ідентифікатор вашого контейнера (19d2a847a54e виділено вище) і пароль будуть відрізнятися.

4.4 Дослідіть рівні абстракції, які зараз працюють на вашому комп'ютері. На схемі ASCII, яка наведена нижче, відображені рівні абстракції в цій реалізації Docker-inside-Docker (dind). Такий рівень складності не є незвичним для сучасних мереж та хмарних інфраструктур



5. Налаштувати Jenkins

5.1 Відкрийте вкладку веб-переглядача.

Перейдіть до <http://localhost:8080/> та увійдіть за допомогою скопійованого пароля адміністратора.

5.2 Встановіть рекомендовані плагіни Jenkins.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б / -2021
	Екземпляр № 1	Арк 11 / 10

Натисніть **Install suggested plugins** та дочекайтеся допоки Jenkins завантажить і встановить плагіни. Під час встановлення, у вікні терміналу ви побачите повідомлення журналу. Переконайтеся, що ви не закриваєте це вікно терміналу. Для доступу до командного рядка, ви можете відкрити інше вікно терміналу.

5.3 Пропустіть створення нового користувача-адміністратора.

Після завершення встановлення вам відкриється вікно **Create First Admin User**. На даний момент натисніть **Skip and continue as admin** в нижній частині

5.4 Пропустіть створення конфігурації екземпляру.

У вікні **Instance Configuration** залишіть все без змін. Внизу натисніть **Save and Finish**.

5.5 Почніть використовувати Jenkins.

В наступному вікні, натисніть **Start using Jenkins**. Тепер ви повинні бути на головній панелі керування, на якому є привітальне повідомлення **Welcome to Jenkins!**

6. Використати Jenkins для запуску на виконання збірки застосунку

Основним елементом Jenkins є «завдання» (job, також відоме як «проект»). Ви можете створювати завдання, які виконують різноманітні задачі, зокрема такі:

- Отримати код зі репозиторію керування вихідним кодом, такого як GitHub.
- Створити застосунок за допомогою скрипта або інструменту збирання (build).
- Зробити упакування застосунку та запустити його на сервері

6.1 Створіть нове завдання.

a. Натисніть на посилання **Create a job**, яке знаходиться під повідомленням **Welcome to Jenkins!**. Крім того, ви можете натиснути на кнопку **New Item** в меню ліворуч.

b. В полі **Enter an item name** введіть **BuildAppJob**.

c. Виберіть як тип завдання: **Вільний стиль проекту (Freestyle project)**. В описі аббревіатура SCM означає керування конфігурацією програмного забезпечення (software configuration management), що є класифікацією програмного забезпечення, яке відповідає за відстеження та контроль змін у програмному забезпеченні.

d. Прокрутіть до самого низу та натисніть кнопку **OK**.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 11

6.2 Налаштуйте Jenkins BuildAppJob.

Ви знаходитесь у вікні конфігурації, де ви можете ввести деталі про своє завдання. Вкладки вгорі - це лише ярлики до розділів нижче. Натисніть по всім вкладкам, щоб переглянути параметри, які ви можете налаштувати. Для цього простого завдання вам потрібно додати лише кілька деталей конфігурації.

a. Натисніть на вкладку **General** та додайте опис свого завдання. Наприклад, "**My first Jenkins job.**"

b. Перейдіть на вкладку **Source Code Management** та оберіть перемикач **Git**. В полі URL-адреси репозиторію додайте посилання на репозиторій GitHub для sample-app, будьте уважні щоб ввести ваше ім'я користувача з урахуванням реєстру. Обов'язково додайте розширення .git в кінці URL-адреси. Наприклад:

```
https://github.com/github-ім'я користувача/sample-app.git
```

c. Для **Credentials** натисніть кнопку **Add** та виберіть **Jenkins**.

d. У діалоговому вікні **Add Credentials** введіть ім'я користувача та пароль GitHub і натисніть кнопку **Add**.

Примітка: Ви отримаєте повідомлення про помилку про те, що з'єднання не вдалося. Це тому, що ви ще не вибрали облікові дані.

e. У спливаючому вікні **Credentials**, в якому наразі вказано **None**, виберіть облікові дані, які ви щойно налаштували.

f. Після того, як ви додали правильну URL-адресу та облікові дані, Jenkins перевіряє доступ до репозиторію. У вас не повинно бути повідомлень про помилки. Якщо такі повідомлення є, перевірте свою URL-адресу та облікові дані. Вам доведеться **Додати** їх знову, оскільки на даний момент немає можливості видалити ті, які ви ввели раніше.

g. У верхній частині вікна конфігурації **BuildAppJob** перейдіть на вкладку **Build**.

h. У спливаючому вікні **Add build step** виберіть **Execute shell**.

i. В полі **Command** введіть команду, яку ви використовуєте для запуску збірки сценарію sample-app.sh.

```
bash ./sample-app.sh
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б / -2021
	Екземпляр № 1	Арк 11 / 12

і. Натисніть кнопку **Save**. Ви повертаєтеся до панелі керування Jenkins з вибраним **BuildAppJob**.

6.3 Зберіть застосунок в Jenkins.

Ліворуч натисніть **Build Now**, щоб розпочати роботу. Jenkins завантажить ваш репозиторій Git та виконає команду збірки **bash ./sample-app.sh**. Ваша збірка має бути успішною, оскільки ви нічого не змінювали в коді з Частини 3, коли ви модифікували код.

6.4 Отримайте доступ до деталей збірки.

Зліва в розділі **Build History** натисніть номер збірки, який має бути **#1**, якщо ви не створювали застосунок кілька разів.

6.5 Перегляньте вихідні дані консолі.

У лівій частині натисніть кнопку **Виведення консолі (Console Output)**. Ви маєте отримати результат, подібний до наведеного нижче. Зверніть увагу на повідомлення про успіх внизу, а також результат команди **docker ps -a**. Працюють два Docker контейнери: один для вашого sample-app, що працює на локальному порту 5050, і один для Jenkins на локальному порту 8080.

```
Started by user admin
Running as SYSTEM

Building in workspace /var/jenkins_home/workspace/BuildAppJob
using credential 0cf684ea-48a1-4e8b-ba24-b2falc5aa3df
Cloning the remote Git repository
Cloning repository https://github.com/github-user/sample-app
  > git init /var/jenkins_home/workspace/BuildAppJob # timeout=10
Fetching upstream changes from https://github.com/github-user/sample-app
  > git --version # timeout=10
using GIT_ASKPASS to set credentials
  > git fetch --tags --progress -- https://github.com/github-user/sample-app
+refs/heads/*:refs/remotes/origin/* # timeout=10
  > git config remote.origin.url https://github.com/github-user/sample-app # timeout=10
  > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* #
timeout=10
  > git config remote.origin.url https://github.com/github-user/sample-app # timeout=10
Fetching upstream changes from https://github.com/github-user/sample-app
using GIT_ASKPASS to set credentials
  > git fetch --tags --progress -- https://github.com/github-user/sample-app
+refs/heads/*:refs/remotes/origin/* # timeout=10
  > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
  > git rev-parse refs/remotes/origin/origin/master^{commit} # timeout=10
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 13

```

Checking out Revision 230ca953ce83b5d6bdb8f99f11829e3a963028bf
(refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 230ca953ce83b5d6bdb8f99f11829e3a963028bf # timeout=10
Commit message: "Changed port numbers from 8080 to 5050"
> git rev-list --no-walk 230ca953ce83b5d6bdb8f99f11829e3a963028bf # timeout=10
[BuildAppJob] $ /bin/sh -xe /tmp/jenkins1084219378602319752.sh
+ bash ./sample-app.sh
Sending build context to Docker daemon 6.144kB

```

```

Step 1/7 : FROM python
---> 4f7cd4269fa9
Step 2/7 : RUN pip install flask
---> Using cache
---> 57a74c0dff93
Step 3/7 : COPY ./static /home/myapp/static/
---> Using cache
---> aee4eb712490
Step 4/7 : COPY ./templates /home/myapp/templates/
---> Using cache
---> 594cdc822490
Step 5/7 : COPY sample_app.py /home/myapp/
---> Using cache
---> a001df90cf0c
Step 6/7 : EXPOSE 5050
---> Using cache
---> eae896e0a98c
Step 7/7 : CMD python3 /home/myapp/sample_app.py
---> Using cache
---> 272c61fddb45

```

Successfully built 272c61fddb45

Successfully tagged sampleapp:latest

```
9c8594e62079c069baf9a88a75c13c8c55a3aeaddde6fd6ef54010953c2d3fbb
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
9c8594e62079	sampleapp	"/bin/sh -c 'python ..."	Less than a second
ago	Up Less than a second	0.0.0.0:5050->5050/tcp	samplerunning
e25f233f9363	jenkins/jenkins:lts	"/sbin/tini -- /usr/..."	29 minutes ago
Up 29 minutes	0.0.0.0:8080->8080/tcp, 50000/tcp	jenkins_server	

Finished: SUCCESS

6.6 Відкрийте іншу вкладку веб-переглядача та перевірте, чи запущений sample app.

Введіть локальну адресу **localhost:5050**. Ви повинні побачити вміст вашого **index.html**, відображеного на світло-блакитному фоні з написом: **You are calling me from 172.17.0.1**, який відображається як **Н1**.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 14

7. Використати Jenkins для тестування збірки

У цій частині ви створите друге завдання, яке перевіряє збірку, щоб переконатися в тому, що вона працює належним чином.

Примітка: Вам потрібно зупинити і видалити **samplerunning** Docker-контейнер.

```
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker stop samplerunning
samplerunning
devasc@labvm:~/labs/devnet-src/jenkins/sample-app$ docker rm samplerunning
samplerunning
```

7.1: Почніть нову завдання для тестування вашого sample-app.

- Поверніться на вкладку веб-переглядача **Jenkins** і натисніть посилання Jenkins у верхньому лівому куті, щоб повернутися на головну панель керування.
- Натисніть на посилання **New Item**, щоб створити нове завдання.
- Поле «Enter an item name» заповніть іменем **TestAppJob**.
- Виберіть **Вільний стиль проекту (Freestyle project)** як тип завдання.
- Прокрутіть до самого низу та натисніть кнопку **ОК**.

7.2 Налаштуйте Jenkins TestAppJob

- Додайте опис для своєї роботи. Наприклад, "My first Jenkins test."
- Залиште для параметра **Source Code Management** значення **None**.
- Перейдіть на вкладку **Build Triggers** та встановіть прапорець - **Збірка, після збірки інших проектів (Build after other projects are built)**. Для вкладки **Projects to watch** введіть **BuildAppJob**.

7.3 Напишіть тестовий скрипт, який повинен працювати після стабільної збірки BuildAppJob.

- Натисніть вкладку **Build**.
- Натисніть на вкладку **Add build step** та виберіть команду **Execute shell**.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 15

с. Введіть наведений нижче скрипт. Команда **If** має бути у одному рядку, включно з **;** **then**. Дана команда буде **grep** виведенням, що повертається від команди **cURL**, щоб побачити, чи повернеться **You are calling me from 172.17.0.1**. Якщо значення «true», скрипт закінчується з кодом «0», що означає, що у збірці **BuildAppJob** немає помилок. Якщо значення «false», скрипт завершує роботу з кодом «1», що означає є збій в **BuildAppJob**

```
if curl http://172.17.0.1:5050/ | grep "You are calling me from 172.17.0.1"; then
    exit 0
else
    exit 1
fi
```

d. Натисніть на **Save**, а потім на посилання ліворуч **Back to Dashboard**.

7.4 Запустіть знову завдання **BuildAppJob** в **Jenkins**.

a. Оновіть веб-сторінку за допомогою кнопки оновлення для вашого браузера.

b. Тепер ви маєте побачити свої два завдання, наведені в таблиці. Для завдання **BuildAppJob** натисніть кнопку праворуч (годинник зі стрілкою).

7.5 Переконайтеся, що обидва завдання виконані.

Якщо все буде добре, ви повинні побачити позначку часу для оновлення стовпця **Last Success** як для **BuildAppJob**, так і для **TestAppJob**. Це означає, що ваш код для обох завдань виконувався без помилок. Але ви також можете переконатися в цьому самі.

Примітка: Якщо мітки часу не оновлюються, переконайтеся, що ввімкнено автоматичне оновлення, натиснувши посилання у верхньому правому кутку.

a. Натисніть посилання для **TestAppJob**. В розділі **Permalinks** виберіть посилання для останньої збірки та натисніть **Console Output**. Ви маєте отримати результат, подібний до наступного:

```
Started by upstream project "BuildAppJob" build number 13
originally caused by:
  Started by user admin
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/TestAppJob
[TestAppJob] $ /bin/sh -xe /tmp/jenkins1658055689664198619.sh
+ grep You are calling me from 172.17.0.1
+ curl http://172.17.0.1:5050/
```

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 16

```

% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total             Spent    Left     Speed

  0      0    0     0     0     0     0     0  ---:--:--  ---:--:--  ---:--:--    0
100    177  100    177    0     0  29772     0  ---:--:--  ---:--:--  ---:--:--  35400
  <h1>You are calling me from 172.17.0.1</h1>
+ exit 0
Finished: SUCCESS

```

b. Немає потреби перевіряти запуск `sample app`, оскільки **TestAppJob** це вже зробив за вас. Проте, ви можете відкрити вкладку браузера для **172.17.0.1:5050**, щоб побачити, що він дійсно працює.

8. Створити конвеєр в Jenkins

Хоча наразі ви можете запустити свої дві роботи, просто натиснувши кнопку «Build Now» для **BuildAppJob**, проте проекти з розробки програмного забезпечення, як правило, є набагато складнішими. Ці проекти можуть отримати значну користь від автоматизації збірок для неперервної інтеграції змін коду та постійного створення збірок розробки, готових до розгортання. В цьому і полягає суть CI/CD. Конвеєр можна автоматизувати для запуску на основі різноманітних тригерів, включаючи періодичні, на основі опитування GitHub щодо наявності змін або зі скрипту, запущеного віддалено.

8.1 Створіть завдання конвеєра.

- Натисніть на посилання **Jenkins** у верхньому лівому куті, а потім **New Item**.
- У полі **Enter an item name** введіть **SamplePipeline**.
- Виберіть **Конвеєр (Pipeline)** як тип завдання.
- Прокрутіть до самого низу та натисніть кнопку **ОК**.

8.2 Налаштуйте завдання SamplePipeline.

- Вгорі натисніть вкладки та перегляньте кожний розділ сторінки налаштування. Зверніть увагу, що існує безліч різних способів запуску збірки. Для завдання **SamplePipeline** ви запускатимете його вручну.
- В розділі **Pipeline** додайте наведений нижче скрипт.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 17

```
node {
  stage('Preparation') {
    catchError(buildResult: 'SUCCESS') {
      sh 'docker stop samplerunning'
      sh 'docker rm samplerunning'
    }
  }
  stage('Build') {
    build 'BuildAppJob'
  }
  stage('Results') {
    build 'TestAppJob'
  }
}
```

Цей скрипт виконує наступні дії:

- Він створює збірку одного вузла, на відміну від розподіленої або багатовузлової. Розподілені або багатовузлові конфігурації призначені для конвеєрів більш великих розмірів, ніж той, який ви будете в цій лабораторній роботі, та не є предметом вивчення даного курсу.
- На етапі **Preparation, SamplePipeline** спочатку переконається, що будь-які попередні екземпляри контейнера-докера **BuildAppJob** зупинені та видалені. Але якщо ще досі є запущений контейнер, ви отримаєте повідомлення про помилку. Тому, ви використаєте функцію **catchError**, щоб перехоплювати будь-які помилки та повертати значення “SUCCESS”. Це забезпечить перехід конвеєра до наступного етапу.
- На етапі **Build, SamplePipeline** побудує ваш **BuildAppJob**.
- На етапі **Results, SamplePipeline** побудує ваш **TestAppJob**.

с. Натисніть кнопку **Save**, і ви повернетесь до панелі керування Jenkins для завдання **SamplePipeline**.

8.3 Запустіть SamplePipeline.

Ліворуч натисніть на кнопку **Build Now**, щоб запустити завдання **SamplePipeline**. Якщо ви написали свій скрипт для конвеєру без помилок, тоді в режимі **Перегляд етапу (Stage View)** повинні відобразитися три зелені прямокутники з кількістю секунд, які були витрачені на побудову кожного етапу. Якщо ж ні, то натисніть «Configure» - ліворуч, щоб повернутися до налаштування **SamplePipeline** та перевірити свій скрипт конвеєра.

Житомирська політехніка	МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА» Система управління якістю відповідає ДСТУ ISO 9001:2015	Ф-22.06- 05.01/121.00.1/Б /-2021
	Екземпляр № 1	Арк 11 / 18

8.4 Перевірте виведення SamplePipeline.

Натисніть на останнє посилання збірки в розділі **Permalinks**, а потім виберіть **Console Output**. Ви маєте отримати результат, подібний до наведеного нижче.

```

Started by user admin
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/jenkins_home/workspace/SamplePipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Preparation)
[Pipeline] catchError
[Pipeline] {
[Pipeline] sh
+ docker stop samplerunning
samplerunning
[Pipeline] sh
+ docker rm samplerunning
samplerunning
[Pipeline] }
[Pipeline] // catchError
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] build (Building BuildAppJob)
Scheduling project: BuildAppJob
Starting building: BuildAppJob #15
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Results)
[Pipeline] build (Building TestAppJob)
Scheduling project: TestAppJob
Starting building: TestAppJob #18
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Запишіть висновки по виконаній роботі.