

- **Name:** Iskander Nafikov
- **E-mail:** i.nafikov@innopolis.university
- **GitHub:** <https://github.com/iskanred>
- **DockerHub:** <https://hub.docker.com/repository/docker/iskanred>
- **Username:** i.nafikov
- **Hostname:** macbook-KN70WX2PH

Task 1: Infra Deployment

1.

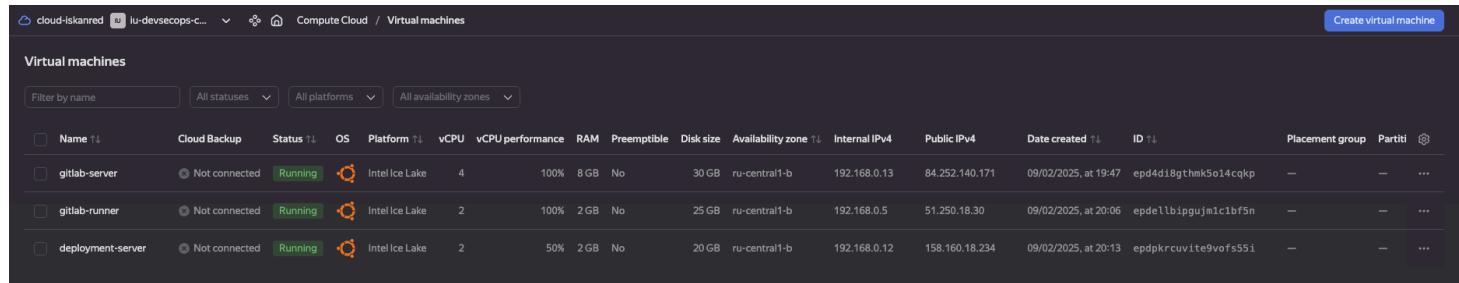
Task description

Deploy three VMs that you will be using as Gitlab Server, Gitlab Runner, and the deployment server. Make sure VMs can reach each other.

Yandex Cloud

I decided to create VMs using Yandex Cloud

- Below is the configuration of my VMs



The screenshot shows a list of virtual machines in the Yandex Cloud Compute Cloud interface. The table has columns for Name, Cloud Backup, Status, OS, Platform, vCPU, vCPU performance, RAM, Preemptible, Disk size, Availability zone, Internal IPv4, Public IPv4, Date created, ID, Placement group, and Partition. Three VMs are listed:

| Name | Status | OS | Platform | vCPU | vCPU performance | RAM | Preemptible | Disk size | Availability zone | Internal IPv4 | Public IPv4 | Date created | ID | Placement group | Partition | |
|-------------------|---------|------------------|----------------|------|------------------|------|-------------|-----------|-------------------|---------------|----------------|----------------------|----------------------|-----------------|-----------|-----|
| gitlab-server | Running | Ubuntu 24.04 LTS | Intel Ice Lake | 4 | 100% | 8 GB | No | 30 GB | ru-central1-b | 192.168.0.13 | 84.252.140.171 | 09/02/2025, at 19:47 | epd4di8gthmk5o14cqkp | — | — | ... |
| gitlab-runner | Running | Ubuntu 24.04 LTS | Intel Ice Lake | 2 | 100% | 2 GB | No | 25 GB | ru-central1-b | 192.168.0.5 | 51.250.18.30 | 09/02/2025, at 20:06 | epdellbipgujm1c1bf5n | — | — | ... |
| deployment-server | Running | Ubuntu 24.04 LTS | Intel Ice Lake | 2 | 50% | 2 GB | No | 20 GB | ru-central1-b | 192.168.0.12 | 158.160.18.234 | 09/02/2025, at 20:13 | epdpkrcuvite9vofs5i | — | — | ... |

- **gitlab-server** (84.252.140.171) is an Ubuntu 24.04 LTS virtual machine that acts as a GitLab Server.
- **gitlab-runner** (51.250.18.30) is a Ubuntu 24.04 LTS virtual machine that acts as a Gitlab Runner.
- **deployment-server** (158.160.18.234) is a Ubuntu 24.04 LTS virtual machine that acts as a Deployment Server.

Network connectivity

- Machines can reach each other in the by public IP

```
Terminal gitlab-server ✘ + ~
ubuntu@gitlab-server:~$ ping 51.250.98.249
PING 51.250.98.249 (51.250.98.249) 56(84) bytes of data.
64 bytes from 51.250.98.249: icmp_seq=1 ttl=60 time=0.715 ms
^C
--- 51.250.98.249 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.715/0.715/0.000 ms
ubuntu@gitlab-server:~$ ping 158.160.18.234
PING 158.160.18.234 (158.160.18.234) 56(84) bytes of data.
64 bytes from 158.160.18.234: icmp_seq=1 ttl=60 time=1.78 ms
^C
--- 158.160.18.234 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.776/1.776/1.776/0.000 ms
ubuntu@gitlab-server:~$
```

```
Terminal gitlab-server ✘ gitlab-runner ✘ + ~
ubuntu@gitlab-runner:~$ ping 158.160.18.234
PING 158.160.18.234 (158.160.18.234) 56(84) bytes of data.
64 bytes from 158.160.18.234: icmp_seq=1 ttl=60 time=1.07 ms
^C
--- 158.160.18.234 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.066/1.066/1.066/0.000 ms
ubuntu@gitlab-runner:~$
```

2.

Task description

Set up **Gitlab Server** (VM1), and create a docker-compose file with the below configs:

- Pull the Gitlab EE or CE edition
- Name the running container as <stx>-gitlab
- Map container ports 80 and 22 to host machine
- Expose the Gitlab server as <stx>.sne.com . (Hint: find the right env variable to pass to the container to update the configs of Gitlab. Update the hosts file to resolve the mentioned DNS record)
- Bind the necessary directories of the Gitlab server container to the host machine (e.g. logs, app data, configs...)
- Run the docker-compose file and make sure the configs are working.
- Access the Gitlab server and log in, create a project name it as <stx>-repo .

- Firstly, I put st15.sne.com as an address to **gitlab-server** VM to my etc/hosts file

```
Terminal gitlab-server ✘ Local ✘ + ~
~/Study/iu-devsecops-course/lab-03 git:[main]
hostname
macbook-KN70WX2HPH

~/Study/iu-devsecops-course/lab-03 git:[main]
cat /etc/hosts | grep st15.sne.com
84.252.140.171 st15.sne.com

~/Study/iu-devsecops-course/lab-03 git:[main]
```

- After it I installed docker using [this](#) instruction on the gitlab-server VM

```
~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/.ssh/id_ed25519 gitlab_server ubuntu@st15.sne.com
ubuntu@gitlab-server:~$ docker version
Client: Docker Engine - Community
  Version:          27.5.1
  API version:     1.47
  Go version:      go1.22.11
  Git commit:      9f9e405
  Built:           Wed Jan 22 13:41:48 2025
  OS/Arch:         linux/amd64
  Context:         default

Server: Docker Engine - Community
Engine:
  Version:          27.5.1
  API version:     1.47 (minimum version 1.24)
  Go version:      go1.22.11
  Git commit:      4c9b3b0
  Built:           Wed Jan 22 13:41:48 2025
  OS/Arch:         linux/amd64
  Experimental:   false
containerd:
  Version:          1.7.25
  GitCommit:        bcc810d6b9066471b0b6fa75f557a15a1cbf31bb
runc:
  Version:          1.2.4
  GitCommit:        v1.2.4-0-g6c52b3f
docker-init:
  Version:          0.19.0
  GitCommit:        de40ad0
ubuntu@gitlab-server:~$ █
```

- Then using the [instruction](#) I set up the Gitlab directory for configuration:

```
sudo mkdir -p /srv/gitlab
sudo chown ubuntu:ubuntu /srv/gitlab
```

```
Terminal gitlab-server ✘ + ▾
ubuntu@gitlab-server:/srv/gitlab$ pwd
/srv/gitlab
```

- Then I created GITLAB_HOME environment variable in `~/.bashrc`

```
export GITLAB_HOME=/srv/gitlab
```

```
Terminal gitlab-server ✘ + ▾
ubuntu@gitlab-server:~$ cat .bashrc | grep GITLAB
export GITLAB_HOME=/srv/gitlab
ubuntu@gitlab-server:~$ echo $GITLAB_HOME
/srv/gitlab
ubuntu@gitlab-server:~$
```

- I created the following `compose.yaml` file (it is located in my GitHub [repo](#) for the course inside the `lab-03/gitlab-server/` directory):

```
services:
  gitlab:
    image: gitlab/gitlab-ce:17.6.4-ce.0
    container_name: st15-gitlab
    restart: always
    hostname: 'st15.sne.com'
```

```

environment:
  GITLAB_OMNIBUS_CONFIG: |
    external_url 'http://st15.sne.com'

ports:
  # - '443:443'
  - target: 80
    published: 80
  - target: 22
    published: 2222

volumes:
  - '$GITLAB_HOME/config:/etc/gitlab'
  - '$GITLAB_HOME/logs:/var/log/gitlab'
  - '$GITLAB_HOME/data:/var/opt/gitlab'

shm_size: '256m'

```

- You can notice that I changed the port of SSH for the container because by port 22 I connect to this VM through SSH.
- Using `scp` I transferred it to my GitLab Server machine

```

scp -i ~/.ssh/id_ed25519_gitlab_server gitlab-server/compose.yaml
ubuntu@st15.sne.com:/srv/gitlab/

```

```

Terminal gitlab-server x Local x + v : -
~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/.ssh/id_ed25519_gitlab_server ubuntu@st15.sne.com
ubuntu@gitlab-server:/srv/gitlab$ cat compose.yaml
services:
  gitlab:
    image: gitlab/gitlab-ce:17.6.4-ce.0
    container_name: st15-gitlab
    restart: always
    hostname: 'st15.sne.com'
    environment:
      GITLAB_OMNIBUS_CONFIG: |
        external_url 'http://st15.sne.com'
    ports:
      # - '443:443'
      - target: 80
        published: 80
      - target: 22
        published: 2222
    volumes:
      - '$GITLAB_HOME/config:/etc/gitlab'
      - '$GITLAB_HOME/logs:/var/log/gitlab'
      - '$GITLAB_HOME/data:/var/opt/gitlab'
    shm_size: '256m'
ubuntu@gitlab-server:/srv/gitlab$

```

- Finally, I run the `compose.yaml`

```

docker compose up -d

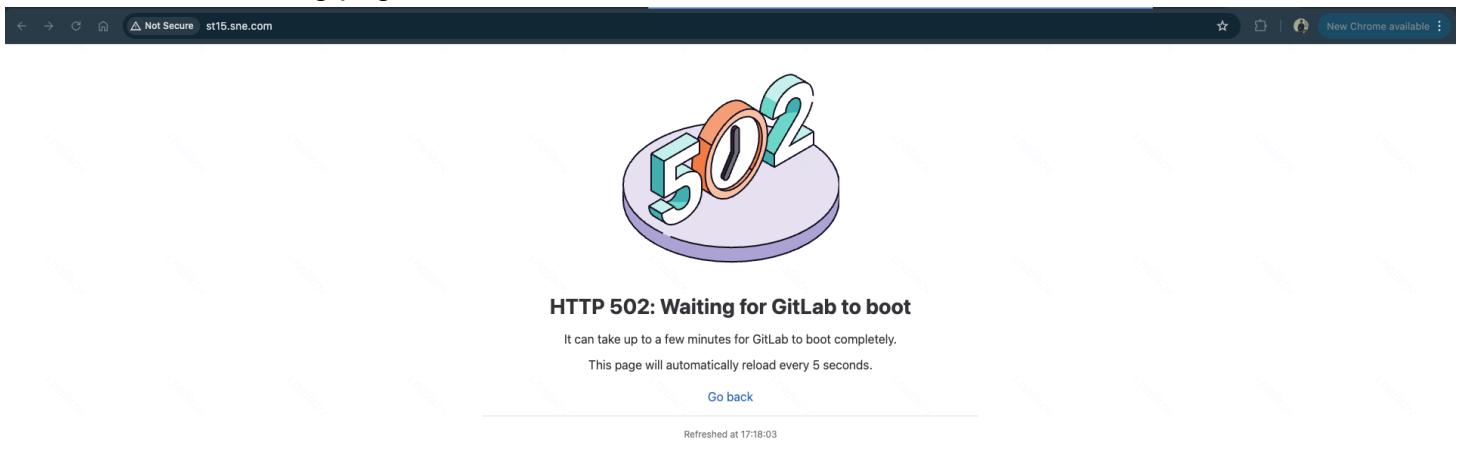
```

```

Terminal gitlab-server x Local x + v : -
~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/.ssh/id_ed25519_gitlab_server ubuntu@st15.sne.com
ubuntu@gitlab-server:/srv/gitlab$ ls
compose.yaml
ubuntu@gitlab-server:/srv/gitlab$ docker compose up -d
[+] Running 2/2
  ✓ Network gitlab_default Created
  ✓ Container st15-gitlab Started
ubuntu@gitlab-server:/srv/gitlab$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
0a0b1df9d2d5 gitlab/gitlab-ce:17.6.4-ce.0 "/assets/wrapper" 6 seconds ago Up 5 seconds (health: starting) 0.0.0.0:80→80/tcp, :::80→80/tcp, 443/tcp, 0.0.0.0:2222→22/tcp, [::]:222→22/tcp
st15-gitlab
ubuntu@gitlab-server:/srv/gitlab$

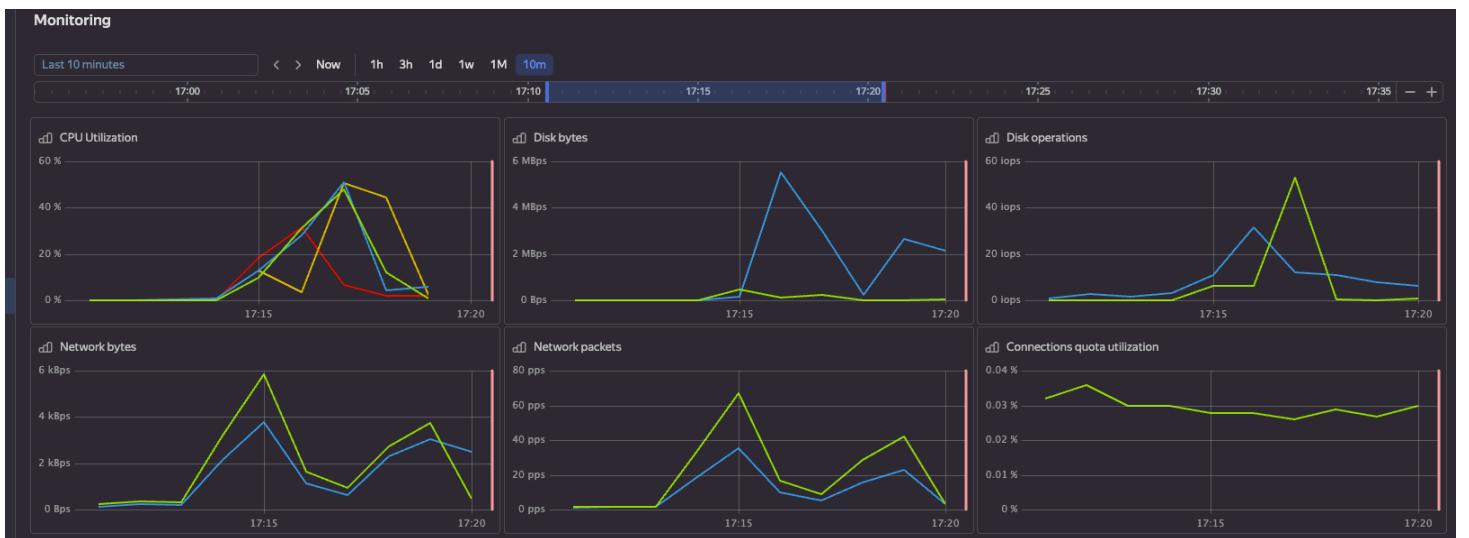
```

- I received the following page in the web-browser in a while



- Also, we can notice that Gitlab consumes a lot of resources

```
Terminal gitlab-server x Local x + ~
~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/.ssh/id_ed25519_gitlab_server ubuntu@st15.sne.com
ubuntu@gitlab-server:/srv/gitlab$ free -h
              total        used        free      shared  buff/cache available
Mem:       7.8Gi       4.3Gi       2.0Gi    125Mi     1.8Gi   3.4Gi
Swap:          0B          0B          0B
ubuntu@gitlab-server:/srv/gitlab$
```



- Finally, the container became "healthy" and I could reach the GitLab

```
Terminal gitlab-server ✘ Local ✘ + ⓘ ~ /Study/iu-devsecops-course/lab-03 git:[main] ssh -i ./ssh/id_ed25519_github_server ubuntu@st15.sne.com ubuntu@gitlab-server:/srv/gitlab$ docker ps CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES 0a0b1df9d2d5 gitlab/gitlab-ce:17.6.4-ce.0 "/assets/wrapper" 6 minutes ago Up 6 minutes (healthy) 0.0.0.0:80→80/tcp, :::80→80/tcp, 443/tcp, 0.0.0.0:2222→22/tcp, ::2222→22/tcp st15-gitlab ubuntu@gitlab-server:/srv/gitlab$
```

Username or primary email

Password

Remember me

[Forgot your password?](#)

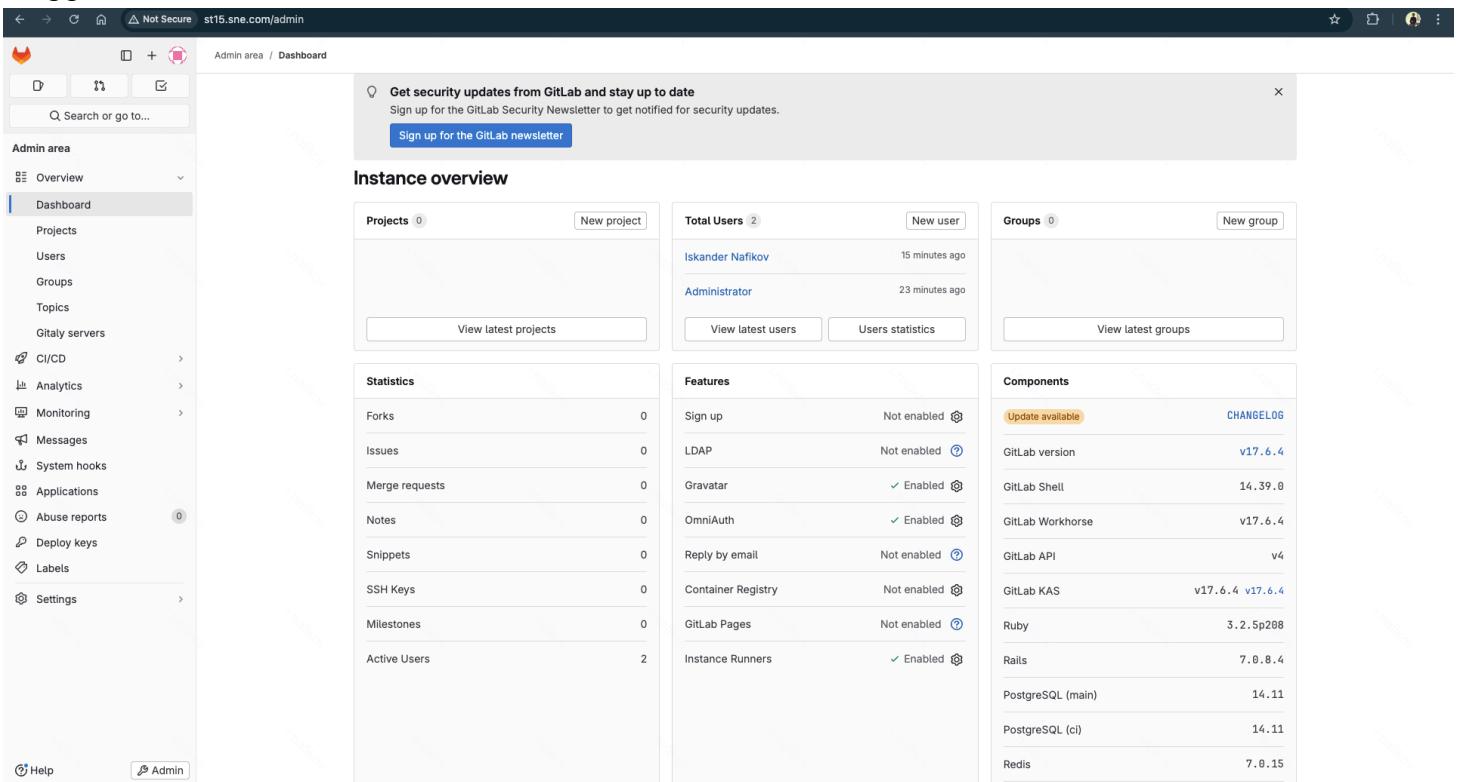
[Sign in](#)

Don't have an account yet? [Register now](#)

Explore Help About GitLab Community forum

English ▾

- I logged in as an administrator



The screenshot shows the GitLab Admin area dashboard. On the left, a sidebar lists various administrative sections: Overview, Dashboard, Projects, Users, Groups, Topics, Gitaly servers, CI/CD, Analytics, Monitoring, Messages, System hooks, Applications, Abuse reports (0), Deploy keys, Labels, Settings, Help, and Admin. The 'Dashboard' section is currently selected. The main content area is titled 'Instance overview' and displays several cards:

- Projects**: 0 projects, New project button.
- Total Users**: 2 users, Iskander Nafikov (15 minutes ago), Administrator (23 minutes ago). Buttons: View latest users, Users statistics.
- Groups**: 0 groups, New group button. Button: View latest groups.
- Statistics**: Forks (0), Issues (0), Merge requests (0), Notes (0), Snippets (0), SSH Keys (0), Milestones (0), Active Users (2).
- Features**: Sign up (Not enabled), LDAP (Not enabled), Gravatar (Enabled), OmniAuth (Enabled), Reply by email (Not enabled), Container Registry (Not enabled), GitLab Pages (Not enabled), Instance Runners (Enabled).
- Components**: Update available (CHangelog v17.6.4), GitLab version (v17.6.4), GitLab Shell (14.39.0), GitLab Workhorse (v17.6.4), GitLab API (v4), GitLab KAS (v17.6.4 v17.6.4), Ruby (3.2.5p208), Rails (7.0.8.4), PostgreSQL (main) (14.11), PostgreSQL (ci) (14.11), Redis (7.0.15).

- And created a project with the name st15-repo

Project

st15-repo

Initial commit i.nafikov authored 9 minutes ago 189943df

| Name | Last commit | Last update |
|-----------|----------------|---------------|
| README.md | Initial commit | 9 minutes ago |

README.md

st15-repo

Getting started

To make it easy for you to get started with GitLab, here's a list of recommended next steps.

Already a pro? Just edit this README.md and make it your own. Want to make it easy? Use the template at the bottom!

Add your files

Create or upload files
 Add files using the command line or push an existing Git repository with the following command:

```
cd existing_repo
git remote add origin http://st15.sne.com/root/st15-repo.git
git branch -M main
git push -uf origin main
```

Integrate with your tools

Set up project integrations

Project information

- 1 Commit
- 1 Branch
- 0 Tags
- 3 KIB Project Storage

README

- + Add LICENSE
- + Add CHANGELOG
- + Add CONTRIBUTING
- + Enable Auto DevOps
- + Add Kubernetes cluster
- + Set up CI/CD
- + Add Wiki
- + Configure Integrations

Created on
February 10, 2025

- In addition I changed my nickname from root to i.nafikov

Iskander Nafikov Admin

Account Groups and projects SSH keys Identities Impersonation Tokens

| | |
|------------------|----------------------------------|
| Iskander Nafikov | Profile |
| | Member since Feb 10, 2025 2:16pm |

Profile page: i.nafikov

| | |
|----------------|---|
| Account | |
| Name: | Iskander Nafikov |
| Username: | i.nafikov |
| Email: | gitlab_admin_055da0@example.com Verified |

3.

Task description

Set up the Gitlab Runner (VM2), don't use the docker approach this time.

- Install and configure shared Gitlab Runner
- Explain what is the Gitlab runner executor and set the executor type to shell
- Set Gitlab Runner tag to <stx>-runner. (You will be using this tag in the pipeline in the coming task)
- Authenticate your Gitlab runner with Gitlab server, and validate.

A **GitLab Runner executor** is the environment in which CI/CD jobs are executed by GitLab Runner. It defines how builds are run, such as whether they are run in a Docker container, a virtual machine, some local machine or directly on a shell.

The **shell executor** runs jobs in a shell on the machine where the runner is installed, executing commands directly in the local operating system's shell. This type is simple and does not require additional virtualisation or containerisation, making it ideal for running scripts quickly and with minimal setup. However, it lacks the isolation and scalability of containerised or virtualised environments.

Installing runner

- Using [this](#) instruction I installed gitlab runner package for apt :

```
ubuntu@gitlab-runner:~$ apt show gitlab-runner
Package: gitlab-runner
Version: 17.8.3-1
Priority: optional
Section: admin
Maintainer: GitLab Inc. <support@gitlab.com>
Installed-Size: 108 MB
Provides: gitlab-ci-multi-runner
Depends: ca-certificates, gitlab-runner-helper-images (= 17.8.3-1), git, curl, tar
Suggests: docker-engine
Conflicts: gitlab-runner-beta, gitlab-ci-multi-runner, gitlab-ci-multi-runner-beta
Replaces: gitlab-ci-multi-runner
Homepage: https://gitlab.com/gitlab-org/gitlab-runner
Download-Size: 26.0 MB
APT-Manual-Installed: yes
APT-Sources: https://packages.gitlab.com/runners/gitlab-runner/ubuntu noble/main amd64 Packages
Description: Gitlab Runner

N: There are 38 additional records. Please use the '-a' switch to see them.
ubuntu@gitlab-runner:~$ gitlab-runner --version
Version: 17.8.3
Git revision: 690ce25c
Git branch: 17-8-stable
GO version: go1.23.2 X:cacheprog
Built: unknown
OS/Arch: linux/amd64
ubuntu@gitlab-runner:~$
```

Registering runner

- For registration I added `st15.sne.com` to the `etc/hosts` of the `gitlab-runner` VM

```
Local gitlab-server gitlab-runner <input type="checkbox" checked="" data-bbox="945 10 955 20" value=""/>
```

~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/.ssh/id_ed25519 gitlab_runner ubuntu@51.250.18.30
ubuntu@gitlab-runner:~\$ cat /etc/hosts | grep st15.sne.com
84.252.140.171 st15.sne.com
ubuntu@gitlab-runner:~\$

- Finally, I registered this runner with a shell executor

```

~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/.ssh/id_ed25519_github_runner ubuntu@51.250.18.30
ubuntu@gitlab-runner:~$ gitlab-runner register --url http://st15.sne.com --token $RUNNER_TOKEN
Runtime platform: arch=amd64 os=linux pid=3640 revision=690ce25c version=17.8.3
WARNING: Running in user-mode.
WARNING: The user-mode requires you to manually start builds processing.
WARNING: $ gitlab-runner run
WARNING: Use sudo for system-mode:
WARNING: $ sudo gitlab-runner ...

Enter the GitLab instance URL (for example, https://gitlab.com):
[http://st15.sne.com]:
Verifying runner... is valid
runner=t3_GybZBu
Enter a name for the runner. This is stored only in the local config.toml file:
[gitlab-runner]:
Enter an executor: ssh, docker, docker-windows, docker-machine, docker-autoscaler, instance, custom, shell, parallels, virtualbox, kubernetes:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!
Configuration (with the authentication token) was saved in "/home/ubuntu/.gitlab-runner/config.toml"
ubuntu@gitlab-runner:~$ 

```

- We can see it on GitLab Server

| System ID | Status | Version | IP Address | Executor | Arch/Platform | Last contact |
|----------------|---|-------------------|--------------|----------|---------------|---------------|
| s_f87c9f11ca63 | Online | 17.8.3 (690ce25c) | 51.250.18.30 | shell | amd64/linux | 4 minutes ago |

Configuring a runner

- Since I would use this runner as a docker build/push mechanism I had to install docker using the same instruction

```

Local gitlab-runner deployment-server gitlab-runner

~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/ssh/id_ed25519_gitlab_runner ubuntu@51.250.18.30
ubuntu@gitlab-runner:~$ docker version
Client: Docker Engine - Community
  Version:          27.5.1
  API version:     1.47
  Go version:      go1.22.11
  Git commit:      9f9e405
  Built:           Wed Jan 22 13:41:48 2025
  OS/Arch:         linux/amd64
  Context:         default

Server: Docker Engine - Community
  Engine:
    Version:          27.5.1
    API version:     1.47 (minimum version 1.24)
    Go version:      go1.22.11
    Git commit:      4c9b3b0
    Built:           Wed Jan 22 13:41:48 2025
    OS/Arch:         linux/amd64
    Experimental:   false
  containerd:
    Version:          1.7.25
    GitCommit:        bcc810d6b9066471b0b6fa75f557a15a1cbf31bb
  runc:
    Version:          1.2.4
    GitCommit:        v1.2.4-0-g6c52b3f
  docker-init:
    Version:          0.19.0
    GitCommit:        de40ad0
ubuntu@gitlab-runner:~$ █

```

- After installation of Docker I had to add gitlab-runner Linux user to the docker group

```

Local gitlab-runner

~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/ssh/id_ed25519_gitlab_runner ubuntu@51.250.18.30
ubuntu@gitlab-runner:~$ cat /etc/group | grep docker
docker:x:987:ubuntu,gitlab-runner
ubuntu@gitlab-runner:~$ █

```

- Finally, I had to add my DockerHub credentials as environment variables to be able to push an image of the application

```

Local gitlab-runner

~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/ssh/id_ed25519_gitlab_runner ubuntu@51.250.18.30
ubuntu@gitlab-runner:~$ grep -o 'DOCKER_.*=' /etc/environment
DOCKER_USER=
DOCKER_PASSWORD=
ubuntu@gitlab-runner:~$ █

```

- In addition, I had to install Java JDK for compiling the source code of the application that I would create using Kotlin/Java stack

```

Local gitlab-runner

~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/ssh/id_ed25519_gitlab_runner ubuntu@51.250.18.30
ubuntu@gitlab-runner:~$ java -version
openjdk version "17.0.14" 2025-01-21
OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-124.04)
OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-124.04, mixed mode, sharing)
ubuntu@gitlab-runner:~$ javac -version
javac 17.0.14
ubuntu@gitlab-runner:~$ █

```

4.

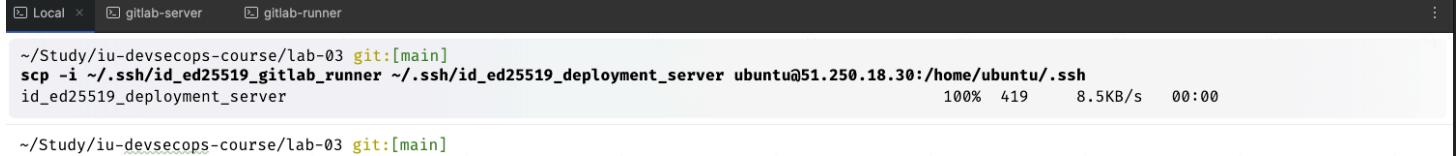
Task description

Set up the Deployment Server (VM3).

a. Set up authentication of your Gitlab runner to be able to deploy to the deployment server.

- I transferred SSH private key for accessing the `deployment-server` to the `gitlab-runner`

```
scp -i ~/.ssh/id_ed25519_gitlab_runner ~/.ssh/id_ed25519_deployment_server  
ubuntu@51.250.18.30:/home/ubuntu/.ssh
```



```
~/Study/iu-devsecops-course/lab-03 git:[main]  
scp -i ~/.ssh/id_ed25519_gitlab_runner ~/.ssh/id_ed25519_deployment_server ubuntu@51.250.18.30:/home/ubuntu/.ssh  
id_ed25519_deployment_server  
100% 419 8.5KB/s 00:00
```

- Then I moved the file with the private key to `home/gitlab-runner/.ssh` and changed its owner because `gitlab-runner` is a Linux user with its own access rights for the files in the system:

```
sudo mv .ssh/id_ed25519_deployment_server /home/gitlab-runner/.ssh/  
sudo chown gitlab-runner /home/gitlab-runner/.ssh/id_ed25519_deployment_server  
sudo chgrp gitlab-runner /home/gitlab-runner/.ssh/id_ed25519_deployment_server
```



```
~/Study/iu-devsecops-course/lab-03/st15-repo git:[main]  
ssh -i ~/.ssh/id_ed25519_gitlab_runner ubuntu@51.250.18.30  
ubuntu@gitlab-runner:~$ sudo mv .ssh/id_ed25519_deployment_server /home/gitlab-runner/.ssh/  
ubuntu@gitlab-runner:~$ sudo ls /home/gitlab-runner/.ssh/  
id_ed25519_deployment_server known_hosts  
ubuntu@gitlab-runner:~$
```



```
~/Study/iu-devsecops-course/lab-03/st15-repo git:[main]  
ssh -i ~/.ssh/id_ed25519_gitlab_runner ubuntu@51.250.18.30  
ubuntu@gitlab-runner:~$ sudo chown gitlab-runner /home/gitlab-runner/.ssh/id_ed25519_deployment_server  
ubuntu@gitlab-runner:~$ sudo chgrp gitlab-runner /home/gitlab-runner/.ssh/id_ed25519_deployment_server  
ubuntu@gitlab-runner:~$ sudo ls -l /home/gitlab-runner/.ssh/id_ed25519_deployment_server  
-rw-r--r-- 1 gitlab-runner gitlab-runner 419 Feb 10 16:47 /home/gitlab-runner/.ssh/id_ed25519_deployment_server  
ubuntu@gitlab-runner:~$
```

- After it I could connect to the deployment-server from the gitlab-runner via SSH

```
Local gitlab-server gitlab-runner x : 

~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/._ssh/id_ed25519_gitlab_runner ubuntu@51.250.18.30
ubuntu@gitlab-runner:~/._ssh$ ls
authorized_keys id_ed25519_deployment_server
ubuntu@gitlab-runner:~/._ssh$ ssh -i id_ed25519_deployment_server ubuntu@158.160.18.234
The authenticity of host '158.160.18.234 (158.160.18.234)' can't be established.
ED25519 key fingerprint is SHA256:q8SZhfjF4JTA0dFXG0bTxfgVP0QXD2dx8daVD+GsMn8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '158.160.18.234' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-52-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Mon Feb 10 04:48:41 PM UTC 2025

 System load: 0.0          Processes:      135
 Usage of /:   15.3% of 19.59GB  Users logged in:    0
 Memory usage: 9%           IPv4 address for eth0: 192.168.0.12
 Swap usage:   0%          

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

10 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun Feb  9 18:12:58 2025 from 95.26.76.76
ubuntu@deployment-server:~$
```

- Also, I installed Docker using the same instruction because I will need to deploy the application using Docker

```
Local deployment-server x : 

~/Study/iu-devsecops-course/lab-03/st15-repo git:[from_scratch_dockerfile]
ssh -i ~/._ssh/id_ed25519_deployment_server ubuntu@158.160.18.234
ubuntu@deployment-server:~$ docker version
Client: Docker Engine - Community
 Version:          27.5.1
 API version:     1.47
 Go version:      go1.22.11
 Git commit:      9f9e405
 Built:            Wed Jan 22 13:41:48 2025
 OS/Arch:          linux/amd64
 Context:          default

Server: Docker Engine - Community
Engine:
 Version:          27.5.1
 API version:     1.47 (minimum version 1.24)
 Go version:      go1.22.11
 Git commit:      4c9b3b0
 Built:            Wed Jan 22 13:41:48 2025
 OS/Arch:          linux/amd64
 Experimental:    false
containerd:
 Version:          1.7.25
 GitCommit:        bcc810d6b9066471b0b6fa75f557a15a1cbf31bb
runc:
 Version:          1.2.4
 GitCommit:        v1.2.4-0-g6c52b3f
docker-init:
 Version:          0.19.0
 GitCommit:        de40ad0
ubuntu@deployment-server:~$ cat /etc/group | grep docker
docker:x:988:ubuntu
ubuntu@deployment-server:~$
```

Task 2: Create CI/CD Pipeline

You can always check the artefacts (such as source code) of the lab in my GitHub repository for the course by this link:

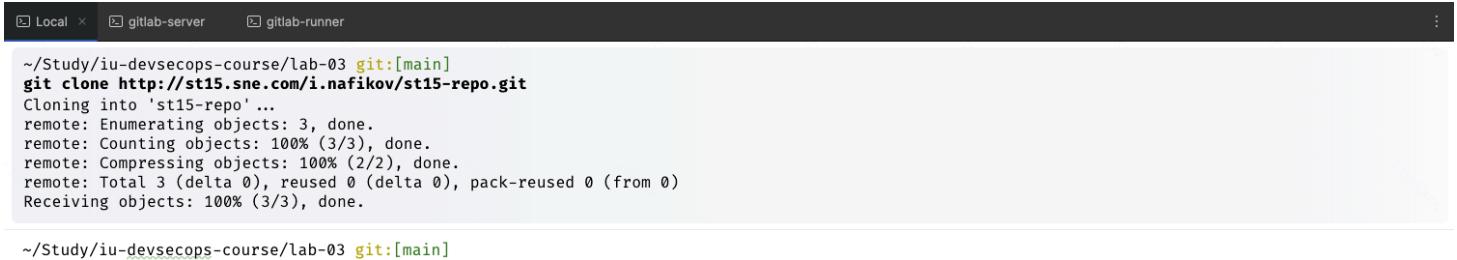
<https://github.com/iskanred/iu-devsecops-course/tree/main/lab-03>

1.

Task description

Clone the project you have created in step 1.2.g.

- I cloned the project



```
~/Study/iu-devsecops-course/lab-03 git:[main]
git clone http://st15.sne.com/i.nafikov/st15-repo.git
Cloning into 'st15-repo'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

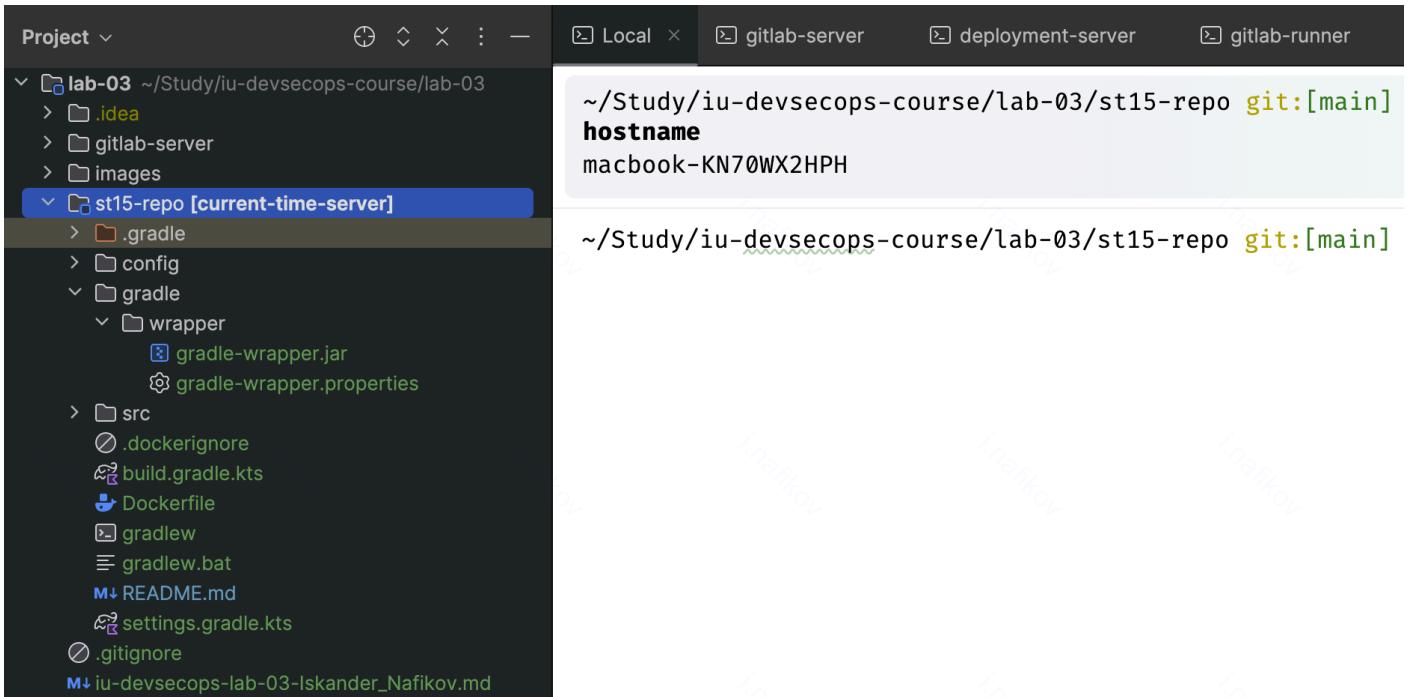
~/Study/iu-devsecops-course/lab-03 git:[main]
```

2.

Task description

Write a simple web application in any programming language. (E.g. Random text or Addition of two numbers)

- I created a simple web application using the Kotlin/Spring stack.
- This application provides information about current time in Europe/Moscow timezone also shows the number of getting this information (number of visits).
- Below is the file structure of this application



The screenshot shows a terminal window and a file explorer side-by-side.

The terminal window displays:

```
~/Study/iu-devsecops-course/lab-03/st15-repo git:[main]
hostname
macbook-KN70WX2HPH
```

The file explorer shows the project structure:

- Project dropdown
- File operations: +, ▲, X, :, -
- Local connection icon
- gitlab-server connection icon
- deployment-server connection icon
- gitlab-runner connection icon
- Project tree:
 - lab-03 (~/Study/iu-devsecops-course/lab-03)
 - .idea
 - gitlab-server
 - images
 - st15-repo [current-time-server] (selected)
 - .gradle
 - config
 - gradle
 - wrapper
 - gradle-wrapper.jar
 - gradle-wrapper.properties
 - src
 - .dockerignore
 - build.gradle.kts
 - Dockerfile
 - gradlew
 - gradlew.bat
 - README.md
 - settings.gradle.kts
 - .gitignore
 - iu-devsecops-lab-03-Iskander_Nafikov.md

- You can check the description of the application inside the [lab-03/st15-repo/README.md](#) in my GitHub [repository](#) for the course (as well as all the source code of the application).

- What is more, I wrote several unit tests for the application and added [detekt](#) static code analyser as a linter

3.

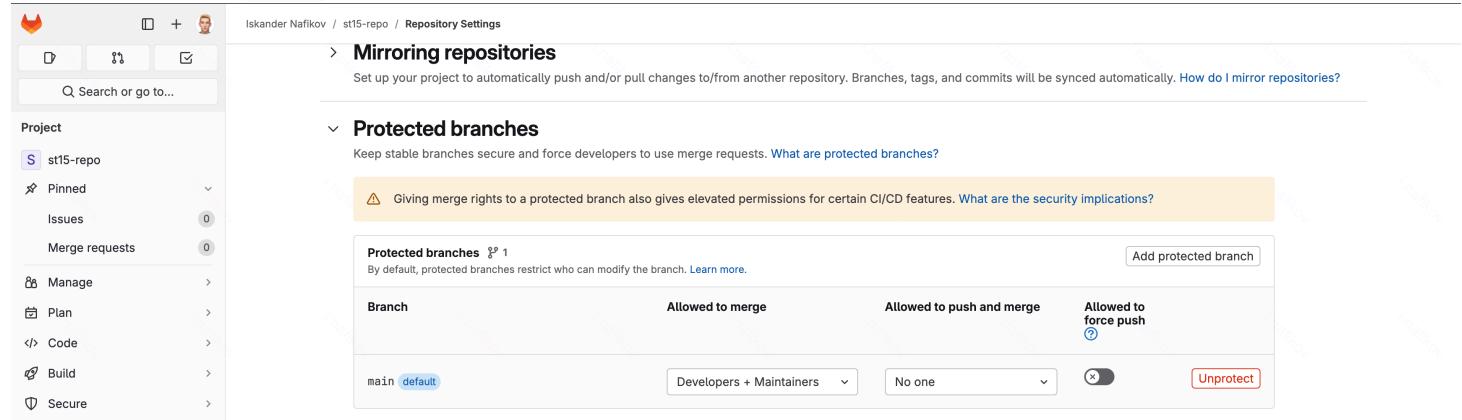
Task description

Create CI/CD pipeline (.gitlab-ci.yml)

1. CI stages of the pipeline should:
 - i. Build the application
 - ii. Run test (to check the application works ok)
 - iii. Build docker image (Note: you need Dockerfile)
 - iv. Push to your docker hub account.
2. CD stages of the pipeline should:
 - i. Pull the docker image and deploy it on the deployment server

Prerequisites

- I made my branch protected from direct pushes (even force) to accept changes only from Merge Requests to the main branch



The screenshot shows the 'Protected branches' section in the GitLab repository settings. It lists the 'main' branch with the following permissions:

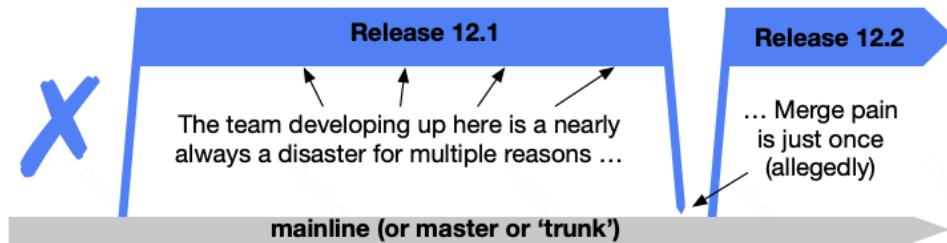
| Branch | Allowed to merge | Allowed to push and merge | Allowed to force push |
|----------------|--------------------------|---------------------------|-----------------------|
| main (default) | Developers + Maintainers | No one | (checkbox) |

There is also a 'Unprotect' button at the bottom right of the table.

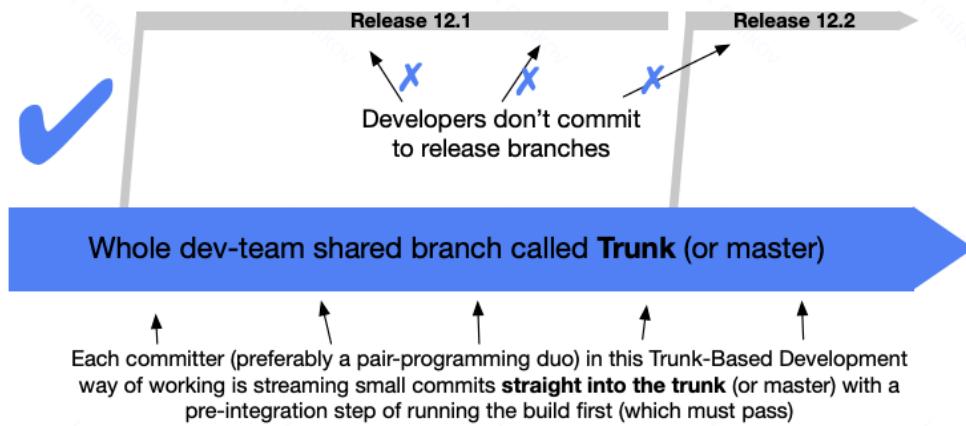
- I did it to configure CI part only triggered by MRs in order to keep runner's resources until a developer make a decision that he or she is ready for merging. So now, developers can push whatever they want but all the checks will start only when they become ready!

- This matches the idea of "Branch by abstraction" from the [Trunk Based Development](#) principles

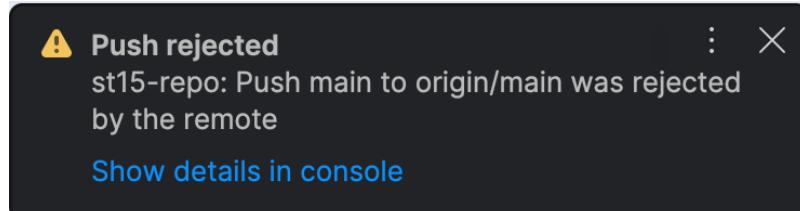
Shared branches off mainline/main/trunk are bad at any release cadence:



Trunk-Based Development For Smaller Teams:



- The configured politics really works:



Overview of .gitlab-ci.yml

- Below is the content of `.gitlab-ci.yml` file that I was created for CI/CD of this application:

`stages:`

- Build
- Check
- Push
- Deploy

`variables:`

```
IMAGE_NAME: current-time-server
FULL_IMAGE_NAME: $IMAGE_NAME:1.0.0
```

`cache: &build_cache`

```
# Take downloaded dependencies from cache
- key: gradle-dependencies-$CI_COMMIT_REF_SLUG
  paths:
```

```
- .gradle/
  policy: pull
# Take application executables
- key: gradle-build-$CI_COMMIT_REF_SLUG
  paths:
    - build/
  policy: pull

build application:
  stage: Build
  before_script: &export_gradle_user_home
    # Say gradle to save dependencies in this directory
    - export GRADLE_USER_HOME=`pwd`/.gradle
  script:
    - ./gradlew clean bootJar --build-cache --no-daemon
  cache:
    # Cache downloaded dependencies for current branch only
    - key: gradle-dependencies-$CI_COMMIT_REF_SLUG
      paths:
        - .gradle/
      policy: pull-push
    # Cache application executables for current branch only
    - key: gradle-build-$CI_COMMIT_REF_SLUG
      paths:
        - build/
      policy: push
  rules:
    - if: $CI_MERGE_REQUEST_ID

run lint:
  stage: Check
  before_script: *export_gradle_user_home
  script:
    - ./gradlew detekt --no-daemon
  cache: *build_cache
  artifacts:
    name: detekt-report
    paths:
      - build/reports/detekt
  needs:
    - "build application"
  rules:
    - if: $CI_MERGE_REQUEST_ID

run tests:
  stage: Check
  before_script: *export_gradle_user_home
  script:
    - ./gradlew test --no-daemon
  cache: *build_cache
```

```

artifacts:
  name: junit-report
  paths:
    - build/reports/tests
needs:
  - "build application"
rules:
  - if: $CI_MERGE_REQUEST_ID

build and push docker image:
  stage: Push
  before_script: &docker_login
  - docker login --username $DOCKER_USER --password $DOCKER_PASSWORD
script:
  - docker build --tag $DOCKER_USER/$FULL_IMAGE_NAME .
  - docker push $DOCKER_USER/$FULL_IMAGE_NAME
cache: *build_cache
needs:
  - "run lint"
  - "run tests"
rules:
  - if: $CI_MERGE_REQUEST_ID

deploy application:
  stage: Deploy
  before_script: *docker_login
  script:
    - >
      ansible-playbook playbooks/deploy_app.yaml --extra-vars
"docker_image=$DOCKER_USER/$FULL_IMAGE_NAME"
  # Run manually and only from "main" branch
  when: manual
  rules:
    - if: $CI_COMMIT_REF_NAME == $CI_DEFAULT_BRANCH

```

CI overview

- The CI pipeline is **triggered only for MRs** and it consists of 3 stages:
 1. **Build** : On this stage the application is built (compiled) by [Gradle](#) with all the necessary dependencies and the JAR file is created. This stage consists of 1 job:
 - **build application** : Gradle builds the the application and job caches all the necessary files (compiled executables and dependencies).
 2. **Check** : On this stage the application is checked for correctness. This stage consists of 2 jobs:
 - **run lint** : Job pulls cache from the previous stage and then Gradle runs checks of the project for any code issues. After success the job provides code analysing report as an artifact. This job needs `build application` job to succeed.

- **run tests** : Job pulls cache from the previous stage and then Gradle runs unit tests. After success the job also provides tests report as an artifact. This job needs `build` application job to succeed.
3. **Push** : On this stage the Docker image is built and pushed to [DockerHub](#). This stage consists of 1 job:

- **build and push docker image** : Job pulls cache from the `Build` stage, logins to Docker registry, builds an image and push it to DockerHub. This job needs `run lint` and `run tests` jobs to succeed.

CD overview

- The CD pipeline is available only for the `main` branch and can be triggered only manually. It consists of 1 stage:

1. **Deploy** : On this stage the Docker image is run on the `deployment-server`. This stage consists of 1 job:

- **deploy application** : Job logins to Docker registry and runs specific Ansible playbook to pull the Docker image from and run it on the `deployment-server`.
 - You can check Ansible configurations in my GitHub repo for the course: [link](#). I'll show the single role that does the job:

```

- name: Create application directory
  file:
    path: "{{ app_dir }}"
    state: directory

- name: Create visits directory
  file:
    path: "{{ app_dir }}/visits_dir"
    state: directory
    # User inside the container should have access to `visits`
    mode: o+rw

- name: Pull Docker image
  community.docker.docker_image:
    name: "{{ docker_image }}"
    state: present
    source: pull
    force_source: true

- name: Run Docker container
  community.docker.docker_container:
    name: current-time-server
    image: "{{ docker_image }}"
    state: started
    published_ports:
      - "8080:8080"
  
```

volumes:

- "{{ app_dir }}/visits_dir/:/current-time-server/visits_dir"

Applying changes

- Firstly, I had to merge my changes to the `main` branch to enable pipelines that are described in `.gitlab-ci.yaml`.
- I committed the changes to the `feature` branch and made my first MR (`feature` → `main`) in this repo.

Feature

Open Iskander Nafikov requested to merge `feature` into `main` 5 minutes ago

Overview 0 Commits 1 Pipelines 0 Changes 23

0 0

8v Revoke approval Approved by you

Ready to merge!

Delete source branch Squash commits Edit commit message

1 commit and 1 merge commit will be added to main (squashes 1 commit).

Merge

Activity

All activity

- Iskander Nafikov assigned to @i.nafikov 4 minutes ago
- Iskander Nafikov added 1 commit 2 minutes ago
 - [68bcf958 - feature](#)
 - [Compare with previous version](#)
- ✓ Iskander Nafikov approved this merge request 2 minutes ago

Preview

Write a comment or drag your files here...

Switch to rich text editing

- I merged my feature branch to the main branch:

The screenshot shows a GitLab repository interface for 'st15-repo'. The 'Code' tab is selected, displaying a list of files and their last update times. Below the code area, there's a 'README.md' section and a 'Current Time Server' section.

Commits:

- Merge branch 'feature' into 'main' (1c7affe3) - Author: Iskander Nafikov, 11 hours ago
- Feature (100aa6a1) - Author: Iskander Nafikov, 11 hours ago
- Initial commit (189943df) - Author: Iskander Nafikov, 1 day ago

Results

Let's check of how CI/CD works.

- First, I created a new branch `from_scratch_dockerfile` and committed a change with adding another one Dockerfile for building an image from scratch with no need to run Gradle locally. Also I

changed README.md correspondingly.

The screenshot shows two Dockerfile snippets in a code editor. The left snippet is for a build step and the right snippet is for a scratch build. Both snippets include comments, environment setup, user creation, file copying, and final configuration like port exposure and entrypoint.

```
1 # This image is built from already compiled executables by Gradle
2 # It works faster for subsequent runs but needs for `./gradlew bootJar` to be exec
3
4 # Application executable image
5 > FROM eclipse-temurin:17-jre
6
7 # Set working directory
8 WORKDIR /current-time-server
9
10 # Add user 'user' and give them permissions for `/current-time-server` dir
11 RUN addgroup --system user 66 \
12     adduser --system --group user 66 \
13     chown user /current-time-server
14 # Switch to user 'user'
15 USER user
16
17 COPY build/libs/current-time-server-1.0.0.jar .
18
19 EXPOSE 8080
20 ENTRYPOINT [ "java", "-jar", "/current-time-server/current-time-server-1.0.0.jar" ]
21
```



```
1 # This image is built from scratch meaning here the source code is compiled to JAR by €✓
2
3 # Application builder image
4 > FROM eclipse-temurin:17-jdk AS APP_BUILDER
5
6 # Set working directory
7 WORKDIR /current-time-server
8
9 # Copy project files excepting those in .dockerignore
COPY .
10
11 # Build application
RUN ./gradlew bootJar --no-daemon
12
13
14
15 # Application executable image
16 FROM eclipse-temurin:17-jre
17
18 # Set working directory
19 WORKDIR /current-time-server
20
21 # Add user 'user' and give them permissions for `/current-time-server` dir
22 RUN addgroup --system user 66 \
23     adduser --system --group user 66 \
24     chown user /current-time-server
25 # Switch to user 'user'
26 USER user
27
28
29 COPY --from=APP_BUILDER current-time-server/build/libs/current-time-server-1.0.0.jar .
30
31 EXPOSE 8080
32 ENTRYPOINT [ "java", "-jar", "/current-time-server/current-time-server-1.0.0.jar" ]
33
```

- After pushing a new branch GitLab suggested me to create an MR from it

The screenshot shows a merge request for the 'from_scratch_dockerfile' branch into the 'main' branch. It includes a success message, a 'Create merge request' button, and standard GitLab navigation links.

You pushed to [from_scratch_dockerfile](#) at [Iskander Nafikov / st15-repo](#) just now

Create merge request

S st15-repo

main / +

History Find file Edit Code Project information

- I created an MR from `from_scratch_dockerfile` to `main`

The screenshot shows a detailed view of a merge request pipeline. It includes sections for overview, merge request pipeline status, merge blocked issues, and activity feed. A sidebar on the left provides navigation for the project.

feat: add from_scratch.Dockerfile

Iskander Nafikov requested to merge [from_scratch_dockerfile](#) into [main](#) just now

Overview 0 Commits 0 Pipelines 0 Changes 2

Merge request pipeline #17 running

Merge blocked: 1 check failed

Activity

Iskander Nafikov assigned to @lnafikov just now

Preview Write a comment or drag your files here...

Switch to rich text editing

Make this an internal note

Comment Close merge request

Build stage

- After creating the MR the pipeline was triggered and the first job `build_application` had been started

The screenshot shows the GitLab interface for a project named 'st15-repo'. The left sidebar is open, showing various sections like Project, Pipelines, Jobs, Pipeline editor, etc. The 'Jobs' section is currently selected. A single job, 'build application', is listed under the 'Running' status. The log output for this job is displayed in the main area, showing the command history and progress. The log starts with 'Running with gitlab-runner 17.8.3 (690ce25c)' and ends with 'Job succeeded'. The pipeline summary on the right indicates an elapsed time of 43 seconds, a queue time of 1 minute 43 seconds, and a timeout of 30m (from runner). The commit hash is b698bad3, and the pipeline ID is #40.

```
1 Running with gitlab-runner 17.8.3 (690ce25c)
2 on gitlab-runner t3_xHKQNM, system ID: s_f87c9f11ca63
3 Preparing the "shell" executor
4 Using Shell (bash) executor...
5 Preparing environment
6 Running on gitlab-runner...
7 Getting source from Git repository
8 Fetching changes with git depth set to 28...
9 Reinitialized existing Git repository in /home/gitlab-runner/builds/t3_xHKQNM/0/i.nafikov/st15-repo/.git/
10 Checking out b698bad3 as detached HEAD [ref= refs/merge-requests/5/head]...
11 Removing .gradle/
12 Skipping Git submodules setup
13 Restoring cache
14 Checking cache for gradle-dependencies-from-scratch-dockerfile-non_protected...
15 Runtime platform arch=amd64 os=linux pid=54177 revision=690ce25c version=17.8.3
16 No URL provided, cache will not be downloaded from shared cache server. Instead a local version of cache will be extracted.
17 WARNING: Cache file does not exist
18 Failed to extract cache
19 Not downloading cache gradle-build-from-scratch-dockerfile-non_protected due to policy
20 Executing "step_script" stage of the job script
21 $ export GRADLE_USER_HOME=`pwd`/.gradle
22 $ ./gradlew clean bootJar --build-cache --no-daemon
23 Downloading https://services.gradle.org/distributions/gradle-8.2.1-bin.zip
...
28 Executing "step_script" stage of the job script
29 $ export GRADLE_USER_HOME=`pwd`/.gradle
30 $ ./gradlew clean bootJar --build-cache --no-daemon
31 Downloading https://services.gradle.org/distributions/gradle-8.2.1-bin.zip
32 Welcome to Gradle 8.2.1!
33 Here are the highlights of this release:
34 - Kotlin DSL: new reference documentation, assignment syntax by default
35 - Kotlin DSL is now the default with Gradle init
36 - Improved suggestions to resolve errors in console output
37 - Reduced sync memory consumption
38 For more details see https://docs.gradle.org/8.2.1/release-notes.html
39 To honour the JVM settings for this build a single-use Daemon process will be forked. For more on this, please refer to https://docs.gradle.org/8.2.1/userguide/gradle\_daemon.html#sec:disabling\_the\_daemon in the Gradle documentation.
40 Daemon will be stopped at the end of the build
41 > Task :clean UP-TO-DATE
42 > Task :compileKotlin
43 > Task :processResources
44 > Task :compileJava NO-SOURCE
45 > Task :classes
46 > Task :resolveMainClassName
47 > Task :bootJar
48 BUILD SUCCESSFUL in 11m 44s
49 5 actionable tasks: 4 executed, 1 up-to-date
50 Saving cache for successful job
51 Creating cache gradle-dependencies-from-scratch-dockerfile-non_protected...
52 Runtime platform arch=amd64 os=linux pid=54637 revision=690ce25c version=17.8.3
53 .gradle/: found 3349 matching artifact files and directories
54 No URL provided, cache will not be uploaded to shared cache server. Cache will be stored only locally.
55 Created cache
56 Creating cache gradle-build-from-scratch-dockerfile-non_protected...
57 Runtime platform arch=amd64 os=linux pid=54644 revision=690ce25c version=17.8.3
58 build/: found 119 matching artifact files and directories
59 No URL provided, cache will not be uploaded to shared cache server. Cache will be stored only locally.
60 Created cache
61 Cleaning up project directory and file based variables
62 Job succeeded
```

- The process of downloading Gradle's and application's dependencies took long time. That's why I used caching. However on the picture we can see a warning `WARNING: Cache file does not exist` which is not surprising since it was a first run of this job.

This screenshot shows the same 'build application' job after it has completed successfully. The log output is identical to the previous one, but the pipeline summary on the right indicates a duration of 12 minutes 45 seconds, a finished time of 1 minute ago, and a queue time of 1 minute 43 seconds. The commit hash is b698bad3, and the pipeline ID is #40. The 'Related jobs' section shows three other 'build application' jobs, all of which have succeeded.

```
12 Skipping Git submodules setup
13 Restoring cache
14 Checking cache for gradle-dependencies-from-scratch-dockerfile-non_protected...
15 Runtime platform arch=amd64 os=linux pid=54177 revision=690ce25c version=17.8.3
16 No URL provided, cache will not be downloaded from shared cache server. Instead a local version of cache will be extracted.
17 WARNING: Cache file does not exist
18 Failed to extract cache
19 Not downloading cache gradle-build-from-scratch-dockerfile-non_protected due to policy
20 Executing "step_script" stage of the job script
21 $ export GRADLE_USER_HOME=`pwd`/.gradle
22 $ ./gradlew clean bootJar --build-cache --no-daemon
23 Downloading https://services.gradle.org/distributions/gradle-8.2.1-bin.zip
24 .....10%.....20%.....30%.....40%.....50%.....60%.....70%.....80%.....90%.....100%
25 Welcome to Gradle 8.2.1!
26 Here are the highlights of this release:
27 - Kotlin DSL: new reference documentation, assignment syntax by default
28 - Kotlin DSL is now the default with Gradle init
29 - Improved suggestions to resolve errors in console output
30 - Reduced sync memory consumption
31 For more details see https://docs.gradle.org/8.2.1/release-notes.html
32 To honour the JVM settings for this build a single-use Daemon process will be forked. For more on this, please refer to https://docs.gradle.org/8.2.1/userguide/gradle\_daemon.html#sec:disabling\_the\_daemon in the Gradle documentation.
33 Daemon will be stopped at the end of the build
34 > Task :clean UP-TO-DATE
35 > Task :compileKotlin
36 > Task :processResources
37 > Task :compileJava NO-SOURCE
38 > Task :classes
39 > Task :resolveMainClassName
40 > Task :bootJar
41 BUILD SUCCESSFUL in 11m 44s
42 5 actionable tasks: 4 executed, 1 up-to-date
43 Saving cache for successful job
44 Creating cache gradle-dependencies-from-scratch-dockerfile-non_protected...
45 Runtime platform arch=amd64 os=linux pid=54637 revision=690ce25c version=17.8.3
46 .gradle/: found 3349 matching artifact files and directories
47 No URL provided, cache will not be uploaded to shared cache server. Cache will be stored only locally.
48 Created cache
49 Creating cache gradle-build-from-scratch-dockerfile-non_protected...
50 Runtime platform arch=amd64 os=linux pid=54644 revision=690ce25c version=17.8.3
51 build/: found 119 matching artifact files and directories
52 No URL provided, cache will not be uploaded to shared cache server. Cache will be stored only locally.
53 Created cache
54 Cleaning up project directory and file based variables
55 Job succeeded
```

- Nevertheless, we see that cache was saved for the successful job after its finish. Therefore, I decided to rerun it and check if caching worked

- And it we can that it actually worked. Elapsed time of the job became much less (10 times) and we can notice there was no warnings about using cache but only successfully extracted cache message.

secure st15.sne.com/l/nafikov/st15-repo/-/jobs/41

Iskander Nafikov / st15-repo / Jobs / #41

build application

Passed Started just now by Iskander Nafikov

Search visible log output Log Copy Replay Up Down Left Right

```

1 Running with gitlab-runner 17.8.3 (690ce25c)
2 on gitlab-runner t3_xHKqNN system ID: 5_f87c9f1ca63
3 Preparing the "shell" executor
4 Using Shell (bash) executor...
5 Preparing environment
6 Running on gitlab-runner...
7 Getting source from Git repository
8 Fetching changes with git depth set to 20...
9 Initialized existing Git repository in /home/gitLab-runner/builds/t3_xHKqNN/0/1.nafikov/st15-repo/.git/
10 Checking out b698bad3 as detached HEAD (ref is refs/merge-requests/3/head)...
11 Removing .gradle/
12 Removing build/
13 Skipping Git submodules setup
14 Restoring cache
15 Checking cache for gradle-dependencies-from-scratch-dockerfile-non_protected...
16 Runtime platform arch=amd64 os=linux pid=55306 revision=690ce25c version=17.8.3
17 No URL provided, cache will not be downloaded from shared cache server. Instead a local version of cache will be extracted.
18 Successfully extracted cache
19 Not downloading cache gradle-build-from-scratch-dockerfile-non_protected due to policy
20 Executing "step_script" stage of the job script
21 $ export GRADLE_USER_HOME=~/.gradle
22 $ ./gradlew clean bootJar --build-cache --no-daemon
23 To honour the JVM settings for this build a single-use Daemon process will be forked. For more on this, please refer to https://docs.gradle.org/8.2.1/userguide/gradle\_daemon.html#sec:disabling\_the\_daemon in the Gradle documentation.
24 Daemon will be stopped at the end of the build
25 > Task :clean UP-TO-DATE
26 > Task :compileKotlin FROM-CACHE
27 > Task :compileJava NO-SOURCE
28 > Task :processResources
29 > Task :classes
30 > Task :resolveMainClassName
31 > Task :bootJar
32 BUILD SUCCESSFUL in 13s
33 5 actionable tasks: 3 executed, 1 from cache, 1 up-to-date
34 Saving cache for successful job
35 Creating cache gradle-dependencies-from-scratch-dockerfile-non_protected...
36 Runtime platform arch=amd64 os=linux pid=55465 revision=690ce25c version=17.8.3
37 .gradle/: found 3351 matching artifact files and directories
38 No URL provided, cache will not be uploaded to shared cache server. Cache will be stored only locally.
39 Created cache
40 Creating cache gradle-build-from-scratch-dockerfile-non_protected...
41 Runtime platform arch=amd64 os=linux pid=55496 revision=690ce25c version=17.8.3

```

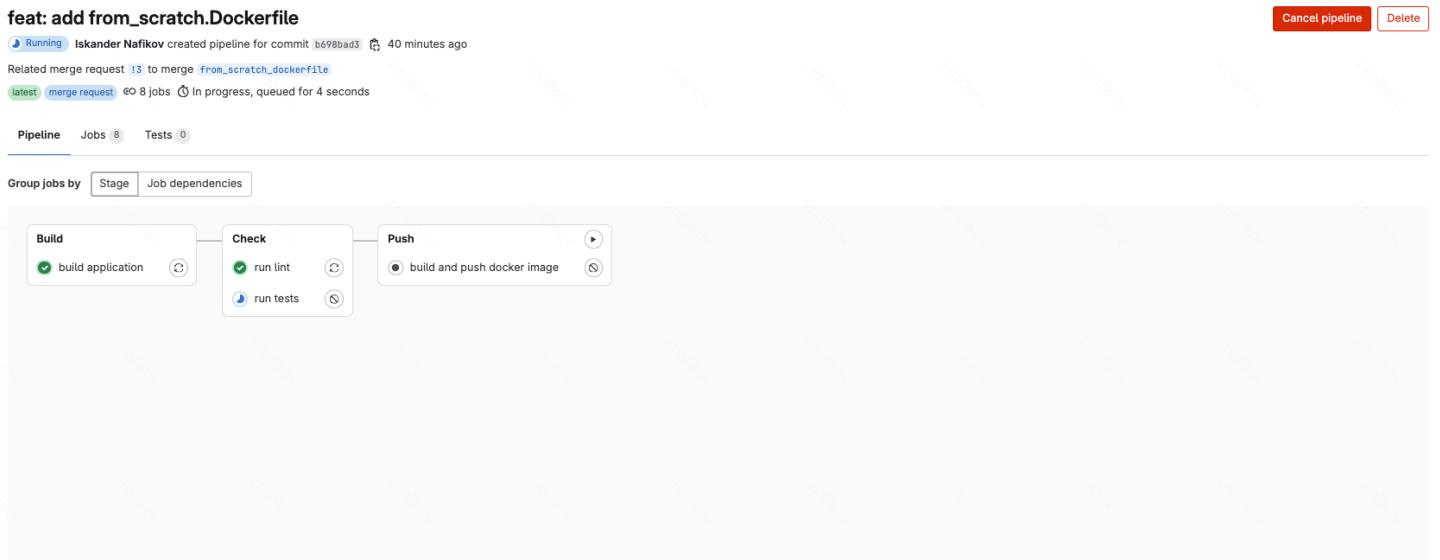
Duration: 1 minute 41 seconds
 Finished: just now
 Queued: 2 minutes 41 seconds
 Timeout: 30m (from runner) ?
 Runner: #2 (xHKqNN3T)
 Commit b698bad3 File in I3
 feat: add from_scratch.Dockerfile
 Pipeline #17 Canceled for I3 with from_scratch_dockerfile File
 Build

Related jobs

- Passed build application
- Running build application
- Queued build application
- Passed build application

Check stage

- After completing the Build stage the Check stage started



- The run lint job had been successfully and quickly finished with using necessary caching

Not Secure st15.sne.com/i.nafikov/st15-repo/-/jobs/36

Iskander Nafikov / st15-repo / Jobs / #36

run lint

Passed Started 9 minutes ago by Iskander Nafikov

Project st15-repo

Pinned Manage Plan Code Build Pipelines Jobs Pipeline editor Pipeline schedules Artifacts Secure Deploy Operate Monitor Analyze Settings Help Admin

Duration: 1 minute 30 seconds
Finished: 7 minutes ago
Queued: 2 seconds
Timeout: 30m (from runner) Runner: #2 (xHqgNN3T)

Job artifacts Job artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.

Keep Download Browse

Commit b698bad3 in I3
feat: add from_scratch.Dockerfile

Pipeline #17 Running for I3 with from_scratch_dockerfile

Check

Related jobs

run tests

→ run lint

run tests

```

1 Running with gitlab-runner 17.8.3 (690ce25c)
2 on gitlab-runner t3_xHqgNN, system ID: s_f87c9f11ca63
3 Preparing the "shell" executor
4 Using Shell (bash) executor...
5 Preparing environment
6 Running on gitlab-runner...
7 Getting source from Git repository
8 Fetching changes with git depth set to 20...
9 Reinitialized existing Git repository in /home/gitlab-runner/builds/t3_xHqgNN/0/i.nafikov/st15-repo/.git/
10 Checking out b698bad3 as detached HEAD (ref is refs/merge-requests/3/head)...
11 Removing .gradle/
12 Removing build/
13 Skipping Git submodules setup
14 Restoring cache
15 Checking cache for gradle-dependencies-from-scratch-dockerfile-non_protected...
16 Runtime platform arch=amd64 os=linux pid=54741 revision=690ce25c version=17.8.3
17 No URL provided, cache will not be downloaded from shared cache server. Instead a local version of cache will be extracted.
18 Successfully extracted cache
19 Checking cache for gradle-build-from-scratch-dockerfile-non_protected...
20 Runtime platform arch=amd64 os=linux pid=54776 revision=690ce25c version=17.8.3
21 No URL provided, cache will not be downloaded from shared cache server. Instead a local version of cache will be extracted.
22 Successfully extracted cache
23 Executing "step_script" stage of the job script
24 $ export GRADLE_USER_HOME=~/gradle
25 $ ./gradlew detekt --no-daemon
26 To honour the JVM settings for this build a single-use Daemon process will be forked. For more on this, please refer to https://docs.gradle.org/8.2.1/userguide/gradle_daemon.html#sec:dishabiling_the_daemon in the Gradle documentation.
27 Daemon will be stopped at the end of the build
28 > Task :detekt
29 BUILD SUCCESSFUL in 48s
30 1 actionable task: 1 executed
31 Saving cache for successful job
32 Not uploading cache gradle-dependencies-from-scratch-dockerfile-non_protected due to policy
33 Not uploading cache gradle-build-from-scratch-dockerfile-non_protected due to policy
34 Uploading artifacts for successful job
35 Uploading artifacts...
36 Runtime platform arch=amd64 os=linux pid=54949 revision=690ce25c version=17.8.3
37 build/reports/detekt: found 6 matching artifact files and directories
38 Uploading artifacts as "archive" to coordinator... 201 Created id:36 responseStatus:201 Created token:glcbt-64
39 Cleaning up project directory and file based variables
40 Job succeeded

```

- After successful job we can download or browse Detekt report as an artifact

Duration: 1 minute 30 seconds

Finished: 44 minutes ago

Queued: 2 seconds

Timeout: 30m (from runner) [?](#)

Runner: #2 (xHKqNN3T)

Job artifacts [?](#)

These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.

[Keep](#)

[Download](#)

[Browse](#)

Commit b698bad3 [检视](#) in !3

feat: add from_scratch.Dockerfile

Pipeline #17 Passed for !3 with from_scratch_dockerfile [检视](#)

[Check](#)



Related jobs

→ run lint

run tests

run tests



st15.sne.com/i.nafikov/st15-repo/-/jobs/36/artifacts/browse/build/reports/detekt/

Iskander Nafikov / st15-repo / Jobs / #36 / Artifacts

Job #36 in pipeline #17 for b698bad3 from refs/merge-requests/3/head by Iskander Nafikov 1 hour ago

Artifacts / build / reports / detekt [Download artifacts archive](#)

| Name | Size |
|--------------|----------|
| detekt.html | 10.3 KIB |
| detekt.md | 580 B |
| detekt.sarif | 82.7 KIB |
| detekt.txt | 0 B |
| detekt.xml | 79 B |

- The same was happened to run tests job

Not Secure st15.sne.com/i.nafikov/st15-repo/-/jobs/42

Iskander Nafikov / st15-repo / Jobs / #42

Project st15-repo Pinned Manage Plan Code Build Pipelines Jobs Pipeline editor Pipeline schedules Artifacts Secure Deploy Operate Monitor Analyze Settings Help Admin

```

9 Reinitialized existing Git repository in /home/gitlab-runner/builds/t3_xHKqNN/0/i.nafikov/st15-repo/.git/
10 Checking out b698bad3 as detached HEAD [ref is refs/merge-requests/5/head]...
11 Removing .gradle/
12 Removing build/
13 Skipping Git submodules setup
14 Restoring cache
15 Checking cache for gradle-dependencies-from-scratch-dockerfile-non_protected...
16 Runtime platform archamd64 os=linux pid=55677 revision=490ce25c version=17.8.3
17 No URL provided, cache will not be downloaded from shared cache server. Instead a local version of cache will be extracted.
18 Successfully extracted cache
19 Checking cache for gradle-build-from-scratch-dockerfile-non_protected...
20 Runtime platform archamd64 os=linux pid=55499 revision=490ce25c version=17.8.3
21 No URL provided, cache will not be downloaded from shared cache server. Instead a local version of cache will be extracted.
22 Successfully extracted cache
23 Executing "step_script" stage of the job script
24 $ export GRADLE_USER_HOME=`pwd`/.gradle
25 $ ./gradlew test --no-daemon
26 To honour the JVM settings for this build a single-use Daemon process will be forked. For more on this, please refer to https://docs.gradle.org/8.2.1/userguide/gradle_daemon.html#sec:id_enabling_the_daemon in the Gradle documentation.
27 Daemon will be stopped at the end of the build
28 > Task :compileKotlin UP-TO-DATE
29 > Task :compileJava NO-SOURCE
30 > Task :processResources UP-TO-DATE
31 > Task :classes UP-TO-DATE
32 > Task :processTestResources NO-SOURCE
33 > Task :compileTestKotlin
34 > Task :compileTestJava NO-SOURCE
35 > Task :testClasses UP-TO-DATE
36 > Task :test
37 OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
38 OpenJDK 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended
39 Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.
40 You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.
41 For more on this, please refer to https://docs.gradle.org/8.2.1/userguide/command_line_interface.html#sec:command_line_warnings in the Gradle documentation.
42 BUILD SUCCESSFUL in 5s
43 4 actionable tasks: 2 executed, 2 up-to-date
44 Saving cache for successful job
45 Not uploading cache gradle-dependencies-from-scratch-dockerfile-non_protected due to policy
46 Not uploading cache gradle-build-from-scratch-dockerfile-non_protected due to policy
47 Uploading artifacts for successful job
48 Uploading artifacts...
49 Runtime platform archamd64 os=linux pid=55976 revision=490ce25c version=17.8.3
50 build/reports/tests: found 14 matching artifact files and directories
51 Uploading artifacts as 'archive' to coordinator... 201 Created id=42 responseStatus=201 Created token=globt-64
52 Cleaning up project directory and file based variables
53 Job succeeded

```

Duration: 1 minute 31 seconds
Finished: just now
Queued: 1 second
Timeout: 30m (from runner) [?](#)
Runner: #2 (xHKqNN3T)

Job artifacts [?](#)
These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.

Keep Download Browse

Commits b698bad3 in 13
feat: add from_scratch.Dockerfile

Pipeline #17 [Running](#) for 13 with from_scratch_dockerfile [?](#)

Check

Related jobs

run lint
→ run tests
run tests

- And here we can also check the artifacts of test running report

st15.sne.com/i.nafikov/st15-repo/-/jobs/42/artifacts/browse/build/reports/tests/test/

Iskander Nafikov / st15-repo / Jobs / #42 / Artifacts

passed Job #42 in pipeline #17 for b698bad3 from refs/merge-requests/3/head by Iskander Nafikov 39 minutes ago

Artifacts / build / reports / tests / test

Download artifacts archive

| Name | Size |
|------------|----------|
| .. | |
| classes | |
| css | |
| js | |
| packages | |
| index.html | 2.88 KiB |

Push stage

- The final Push stage had been started with the build and push docker image job and it also succeed

```

39 #1 [internal] load build definition from Dockerfile
40 #1 transferring dockerfile: 308 0.6s
41 #1 transferring dockerfile: 669B 0.0s done
42 #1 DONE 0.1s
43 #2 [internal] load metadata for docker.io/library/eclipse-temurin:17-jre
44 #2 ...
45 #3 [auth] library/eclipse-temurin:pull token for registry-1.docker.io
46 #3 DONE 0.0s
47 #2 [internal] load metadata for docker.io/library/eclipse-temurin:17-jre
48 #2 DONE 2.8s
49 #4 [internal] load .dockignore
50 #4 transferring context: 928 done
51 #4 DONE 0.0s
52 #5 [1/4] FROM docker.io/library/eclipse-temurin:17-jre@sha256:346c5f54f77fe750741ed254e2e85af989870f56e65e837e9d0d83bdc40910
53 #5 DONE 0.0s
54 #6 [internal] load build context
55 #6 transferring context: 27.07MB 0.2s done
56 #6 DONE 0.2s
57 #7 [2/4] WORKDIR /current-time-server
58 #7 CACHED
59 #8 [3/4] RUN addgroup --system user && adduser -system -group user && chown user /current-time-server
60 #8 CACHED
61 #9 [4/4] COPY build/libs/current-time-server-1.0.0.jar .
62 #9 DONE 6.8s
63 #10 exporting to Image
64 #10 exporting layers
65 #10 exporting layers 0.2s done
66 #10 writing image sha25a-2194cb9a45fb1f4e48767a7d654bfcb844692b7626a4d0eaa62abf4392e3d56b 0.0s done
67 #10 naming to docker.io/iskanred/current-time-server:1.0.0 0.0s done
68 #10 DONE 0.3s
69 $ docker push ${DOCKER_USER}/${FULL_IMAGE_NAME}
70 The push refers to repository [docker.io/iskanred/current-time-server]
71 02c5405384b9: Preparing
72 fd4ec8781d55: Preparing
73 1a83d6d76789: Preparing
74 SebTecl2c0fb: Preparing
75 Sck9f67857a: Preparing
76 93fb089434f0: Preparing
77 de6284a56c1: Preparing
78 407c01ed0d34: Preparing
79 93fb089434f0: Waiting
80 de6284a56c1: Waiting
81 407c01ed0d34: Waiting
82 1a83d6d76789: Layer already exists
83 fd4ec8781d55: layer already exists
84 Sck9f67857a: layer already exists
85 SebTecl2c0fb: layer already exists
86 de6284a56c1: layer already exists
87 93fb089434f0: layer already exists
88 407c01ed0d34: layer already exists
89 02c5405384b9: Pushed
90 90 0.0: digest: sha256:b61c99fb5b4922e84298f634bef7f59227ad43b49b5900991f79390be5af2 size: 1995
91 Saving cache for successful job
92 Not uploading cache because build-from-scratch-dockerfile-non_protected due to policy
93 Not uploading cache because build-from-scratch-dockerfile-non_protected due to policy
94 Cleaning up project directory and file based variables
95 Job succeeded

```

Duration: 2 minutes 2 seconds
 Finished: 1 minute ago
 Queued: 2 seconds
 Timeout: 30m (from runner)

Commit 1d98bae in 13
 feat: add_from_scratch.Dockerfile

Pipeline #17 Passed for i3 with fro_m_scratch_dockerfile

Push

Related jobs
 → build and push docker image

- And we can see that the Docker image of the application was actually pushed

The screenshot shows the Docker Hub repository page for the 'current-time-server' repository under the 'iskanred' organization. The repository has a size of 135.5 MB and was last pushed 3 minutes ago. It contains one tag, '1.0.0'. The 'General' tab is selected, showing basic repository details like description and category fields, which are currently empty. The 'Tags' section lists the single tag. The 'Automated builds' section indicates that manual pushes are possible and automated builds are available with Pro, Team, and Business subscriptions. The 'Repository overview' section is incomplete and awaiting content.

- You can check the image by this link: <https://hub.docker.com/repository/docker/iskanred/current-time-server/general>

- Finally, the pipeline succeeded and we can go to merging the changes

st15.sne.com/j/nafikov/st15-repo/-/pipelines/17

Iskander Nafikov / st15-repo / Pipelines / #17

feat: add from_scratch.Dockerfile

Passed Iskander Nafikov created pipeline for commit b698bad3 1 hour ago, finished 36 minutes ago

Related merge request [I3](#) to merge [from_scratch_dockerfile](#)

latest merge request [go](#) 8 jobs ⏱ 6 minutes 46 seconds, queued for 4 seconds

Pipeline Jobs 8 Tests 0

Group jobs by Stage Job dependencies

```
graph LR; Build[Build] --> Check[Check]; Check --> Push[Push];
```

Build

- build application

Check

- run lint
- run tests

Push

- build and push docker image

Deploy stage

- The changes were merged and now we can go the CD process

1 Merge requests / !3

feat: add from_scratch.Dockerfile

Merged Iskander Nafikov requested to merge [from_scratch_dockerfile](#) into [main](#) 3 hours ago

- We can notice that now a new pipeline is available but the job can be run manually only

The screenshot shows the CircleCI web interface for a project named 'stt15-repo'. On the left, a sidebar menu is open with various options like Project, Pipelines, Jobs, Pipeline editor, etc. The 'Pipelines' option is selected. In the main content area, there is a section titled 'feat: add from_scratch.Dockerfile' which was created by 'Iskander Nafikov' for the 'main' branch. It shows one job named 'latest' with a status of 'Skipped'. Below this, there are tabs for 'Pipeline', 'Jobs 1', and 'Tests 0'. Under the 'Pipeline' tab, there is a 'Deploy' step with a single job named 'deploy application'.

- For now there is no application deployed on the deployment-server

```

~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/ssh/id_ed25519_deployment_server ubuntu@158.160.18.234
ubuntu@deployment-server:~$ ls
ubuntu@deployment-server:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ubuntu@deployment-server:~$ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu@deployment-server:~$ 
ubuntu@deployment-server:~$ curl 127.0.0.1:8080
curl: (7) Failed to connect to 127.0.0.1 port 8080 after 0 ms: Couldn't connect to server
ubuntu@deployment-server:~$ 

```

- So let's run the deploy job

Iskander Nafikov / st15-repo / Pipelines / #24

feat: add from_scratch.Dockerfile

Running

Iskander Nafikov created pipeline for commit 4294029f 2 minutes ago

For main

latest 1 job In progress, queued for 145 seconds

Pipeline Jobs 1 Tests 0

Deploy

deploy application

- The job is done successfully

The screenshot shows the GitLab Pipeline interface. On the left, there's a sidebar with project navigation (Project, Manage, Plan, Code, Build, Pipelines, Jobs, Pipeline editor, Pipeline schedules, Artifacts, Secure, Deploy, Operate, Monitor, Analyze, Settings). The 'Jobs' section is selected. In the center, the log output for Pipeline #24 is displayed, showing the deployment process. The log includes commands like 'Running on gitlab-runner...', 'Getting source from Git repository', 'Fetching changes with git depth set to 28...', 'Reinitialized existing Git repository in /home/gitlab-runner/builds/t3_xHKqNN/0/l.nafikov/st15-repo/.git/...', 'Checking out 4294029f as detached HEAD (ref is main)...', and 'deploy application'. The log ends with 'Job succeeded'. On the right side of the log, there's a summary: Duration: 17 seconds, Finished: just now, Queued: 3 seconds, Timeout: 30m (from runner), Runner: #2 (xHKqNN3T), Commit: 4294029f, Pipeline: #24 Passed for main, and Related jobs: deploy application.

```

6  Running on gitlab-runner...
7  Getting source from Git repository
8  Fetching changes with git depth set to 28...
9  Reinitialized existing Git repository in /home/gitlab-runner/builds/t3_xHKqNN/0/l.nafikov/st15-repo/.git/...
10 Checking out 4294029f as detached HEAD (ref is main)...
11 Removing .gradle/
12 Removing build/
13 Skipping Git submodules setup
14 Restoring cache
15 Checking cache for gradle-dependencies-main-protected...
16 Runtime platform arch=amd64 os=linux pid=64754 revision=690ce25c version=17.8.3
17 No URL provided, cache will not be downloaded from shared cache server. Instead a local version of cache will be extracted.
18 WARNING: Cache file does not exist
19 Failed to extract cache
20 Checking cache for gradle-build-main-protected...
21 Runtime platform arch=amd64 os=linux pid=64765 revision=690ce25c version=17.8.3
22 No URL provided, cache will not be downloaded from shared cache server. Instead a local version of cache will be extracted.
23 WARNING: Cache file does not exist
24 Failed to extract cache
25 Executing "step_script" stage of the job script
26 $ docker login --username $DOCKER_USER --password $DOCKER_PASSWORD
27 WARNING! Using --password with the CLI is insecure. Use --password-stdin.
28 WARNING! Your password will be stored unencrypted in /home/gitlab-runner/.docker/config.json.
29 Configure a credential helper to remove this warning. See
30 https://docs.docker.com/engine/reference/commandline/login/#credential-stores
31 Login Succeeded
32 $ cd ansible
33 $ ansible-playbook playbooks/deploy_app.yaml --extra-vars "docker_image=$DOCKER_USER/$FULL_IMAGE_NAME"
34 PLAY [Deploy Application] ****
35 TASK [Gathering Facts] ****
36 ok: [158.160.18.234]
37 TASK [current_time_server : Create application directory] ****
38 changed: [158.160.18.234]
39 TASK [current_time_server : Create visits directory] ****
40 changed: [158.160.18.234]
41 TASK [current_time_server : Pull Docker image] ****
42 changed: [158.160.18.234]
43 TASK [current_time_server : Run Docker container] ****
44 changed: [158.160.18.234]
45 PLAY RECAP ****
46 158.160.18.234 : ok=5    changed=4  unreachable=0   failed=0   skipped=0   rescued=0   ignored=0
47 Saving cache for successful job
48 Not uploading cache gradle-dependencies-main-protected due to policy
49 Not uploading cache gradle-build-main-protected due to policy
50 Cleaning up project directory and file based variables
51 Job succeeded

```

- Let's check if the application has been deployed

The screenshot shows a terminal window with the following command and output:

```

~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/._ssh/id_ed25519_deployment_server ubuntu@158.160.18.234
ubuntu@deployment-server:~$ ls
current-time-server
ubuntu@deployment-server:~$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                 NAMES
73c02cba40e9        iskanred/current-time-server:1.0.0   "java -jar /current-..."   About a minute ago   Up About a minute   0.0.0.0:8080→8080/tcp
current-time-server
ubuntu@deployment-server:~$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED             SIZE
iskanred/current-time-server  1.0.0    79872de9e43e  4 minutes ago   290MB
ubuntu@deployment-server:~$ curl 127.0.0.1:8080 -w '\n'
23:40:52
ubuntu@deployment-server:~$

```

- It has

4.

Task description

Validate that the deployment is successful by accessing the web app via the browser on deployment server side.

- I accessed my web application via the browser accessing `deployment_server` by its public IP



- As we see the visits are counted correctly

A screenshot of a terminal window titled 'deployment-server'. The window displays a series of curl commands being run against the local host (127.0.0.1:8080) to count visits. The output shows the visit count increasing from 0 to 7 over time. The terminal also shows some git-related commands at the top.

```
~/Study/iu-devsecops-course/lab-03 git:[main]
ssh -i ~/.ssh/id_ed25519_deployment_server ubuntu@158.160.18.234
ubuntu@deployment-server:~$ curl 127.0.0.1:8080 -w '\n'
23:42:13
ubuntu@deployment-server:~$ curl 127.0.0.1:8080/visits -w '\n'
5
ubuntu@deployment-server:~$ curl 127.0.0.1:8080 -w '\n'
23:42:24
ubuntu@deployment-server:~$ curl 127.0.0.1:8080/visits -w '\n'
6
ubuntu@deployment-server:~$ curl 127.0.0.1:8080 -w '\n'
23:42:27
ubuntu@deployment-server:~$ curl 127.0.0.1:8080/visits -w '\n'
7
ubuntu@deployment-server:~$
```

Task 3: Polish the CI/CD

5.

Task description

Update the CD stages to be able to deploy the web application using Ansible.

Already done in the Task 2

6

Task description

Update the pipeline to support multi-branch (e.g. master and develop) and jobs should be triggered based on the specific target branch.

I had already created a mechanism of triggering by MR for any new branch while Deploy stage works only on main branch. This was done to satisfy "Branch by abstraction" principle.

7

Task description

Update keywords such as cache, artifact, needs, and dependencies to have more control of pipeline execution.

Already done in the Task 2