Es cluster of 3 nodes 23/6

Data persistence

* One option by docker compose
* Second on one host

3 containers on one host

With –link flag name of the container

ליצר שלוש תיקיות ולהגדיר עם /es1 /es2 /es3

-v

docker compose

התקנה של פאקיג נוסף חייב להיות מותאם לורסית הדוקר

docker\_compose.yml

version:

service:

es1

iname:elasticsearch:7.1.1

port:

- 9200

- 9300

Volume:

Docker Swarn

Cluster up 5000 nodes

Docker services

Nginx 80:80

Image Nginx

Containers: 10

ברגע שהותקן הסוורן עולים 10 קונטיינרים

ודוקר יחליט איך לפזר אותם בין המכונות

חלוקת התעבורה round robin בין הקונטיינרים

Commands : Docker service

docker service create name nginx -p 80:80 nqinx

docker service ps nginx

cocker service scale nginx=30 הגדלת הקונטיינרים

Docker Orchestration

אפשר להוסיף בדיקת חיות כדי לודא שהקונטיינר באויר

Docker create –help לבדוק בפקודה את משתני הhealth

Health cmd string

Health – interval int

Health retries

Health -timeout duration

אפשרות לrollback מה לעשות במקרה של נפילה

מה קורה אם יש צורך בעדכון

docker service update –help

docker network

docker service inspect < name >

מראה את הנתונים של הקונטיינר וגם הרשת

יצירת רשת לחיבור כמה מכונות בקלות

Docker network create driver=overlay< שם הרשת > endpoint-mode=dnsrr

Docker service logs -f <service name >

פקודות :

1. docker network ls shows logical networks, it’s types and drivers

2. docker network create –driver=x mynetwork(name) creates a network using the specified driver with a chosen name

3. docker service create –network mynetwork(name) –end-point-mode=dnsrr (enables DNS discovery) –name X nginx(image)

This will allow the containers to contact each other via DNS discovery through the service

docker network create --driver overlay mydrupal

docker network ls

docker service create --name psql --network mydrupal -e POSTGRES\_PASSWORD=mypass postgres

docker service ls

docker service ps psql

docker container logs psql TAB COMPLETION

docker service create --name drupal --network mydrupal -p 80:80 drupal

docker service ls

watch docker service ls

docker service ps drupal

docker service inspect drupal

## Scaling Out with Routing Mesh

docker service create --name search --replicas 3 -p 9200:9200 elasticsearch:2

docker service ps search

תוויות –label

להגדיר תוויות לסרוויסים למשל כדי למיין את הסרוויסים איפה לרוץ ....

Git

ניהול גרסאות וקוד

Working

Commit

Repository

תיקייה שמכילה קבצים בserver

התיקיה מתנהלת בכמה גרסאות branch

Default branch – master

Plugin :

Git

Pipeline

Blueocean

Image Jenkins with git and pipeline my bloody jenkins (ohad …)

Git flow:

Feature

Hotfix

מייצרים מהמסטר קופי שנקרא dev branch

שעליו המפתחים עובדים .

ממנו יוצאים לfeature branch

ואחכ חוזרים לdev master ….

חשוב לעבוד מסודר ועל כל תיקון לפתוח branch חדש

שיעורי ביית

לפתוח חשבון בgithub

Open private repository

Open public repository Evgeni

Docker service Jenkins

Single container

קורס 26.6

Jenkins

חלק מהדוקרס ויש plugin מוכנים כמעט להכל .

Jobs :

Freestyle

DSL / Scripted pipeline

הגוב)סט של פעולות ) של הגנקינס מאזין לגיט וברגע שיש שינוי מבצע את כל התהליכים .. בדיקות הפצה וכדומה

הגנקיס צריך לשמור את התוצרים בינרים דוקרים וכדומה עדיף :

Nexus

Artifactory

Docker repository

איך בונים pipeline ?

גנקניס רצוי להתקין בתוך קונטיינרים כי מעבר בין גרסאות גנקינס הוא בעייתי .

Home directory /var/Jenkins\_home /opt/Jenkins\_home

מכיל הגדרות יוזרים ...

אמור לשמור את התיקייה הזו ואז ניתן לחזור אליה בתור גיבוי . וגם לעבוד על העתק שלה במקביל עם קונטיינר אחר .

Docker service mode

Global container on each node

Replicated define a number

Jenkins user 1000

Port 8080

Port 50000 for slave

לגנקינס יש גוי לעבודה

מה שמענין זה הplugin

הגדרת גובים

Post action – send notification by mail

להתקין את הimages הנל הם מומלצים:

Jenkis/jenkis:lts

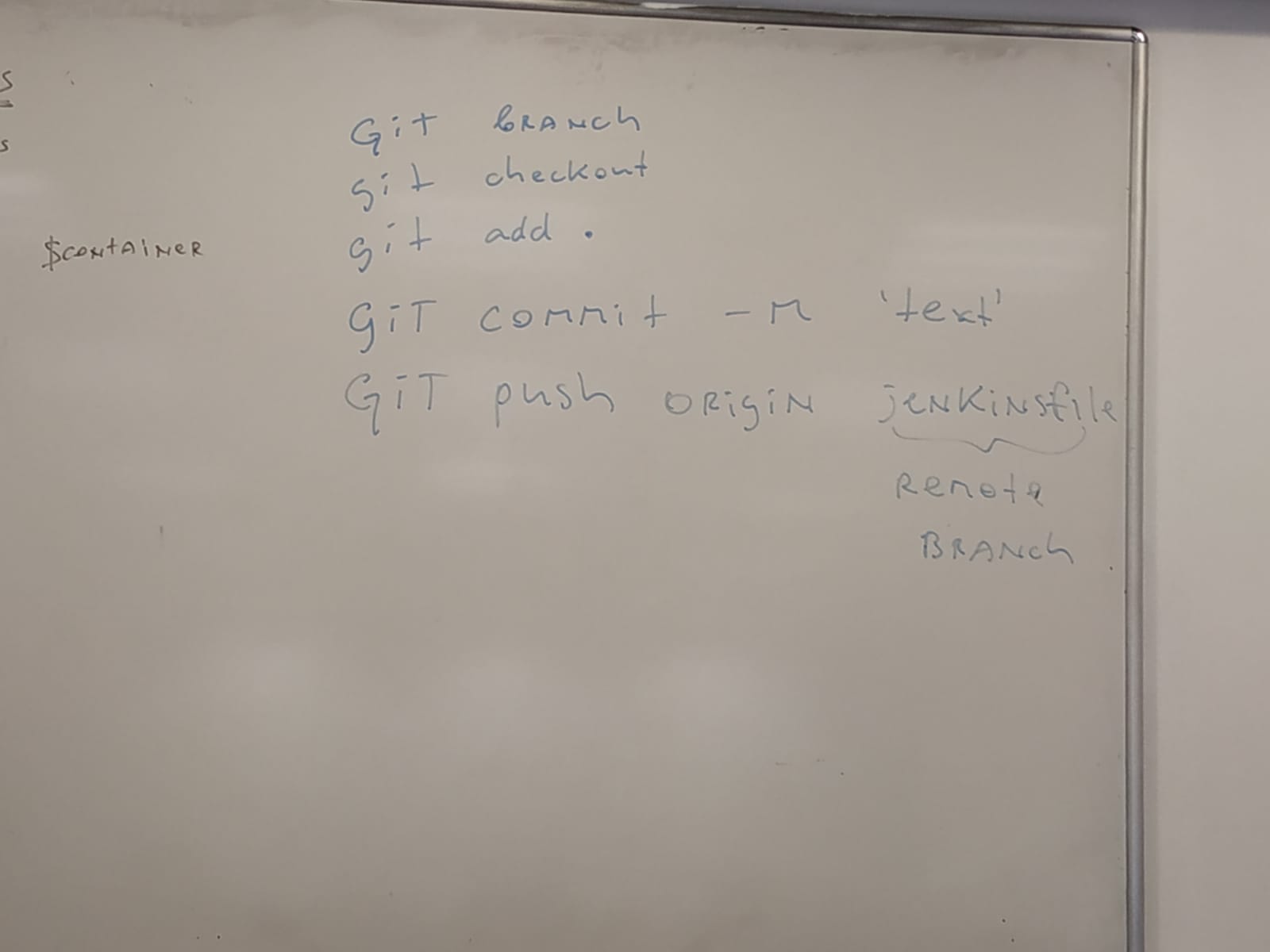
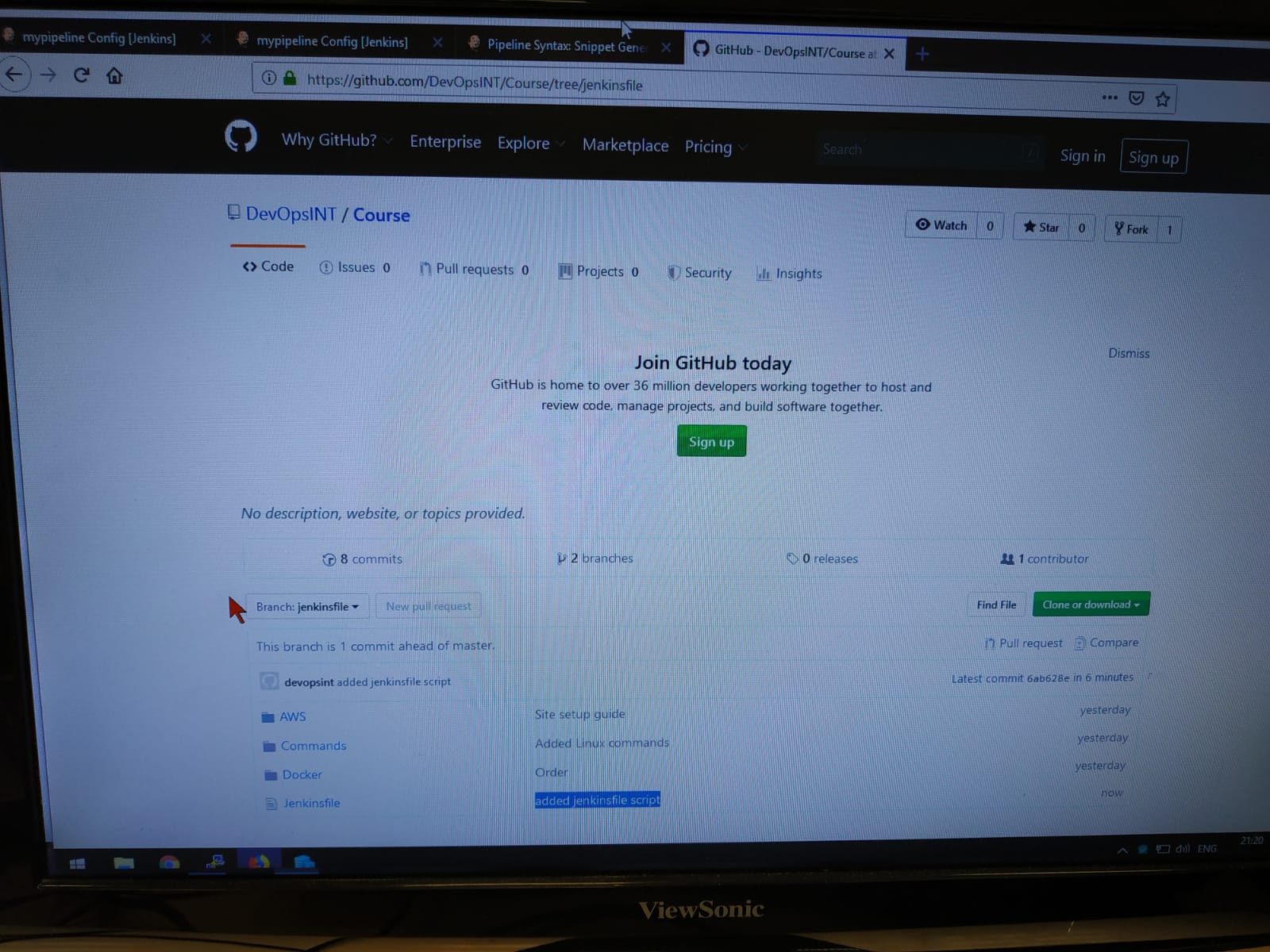
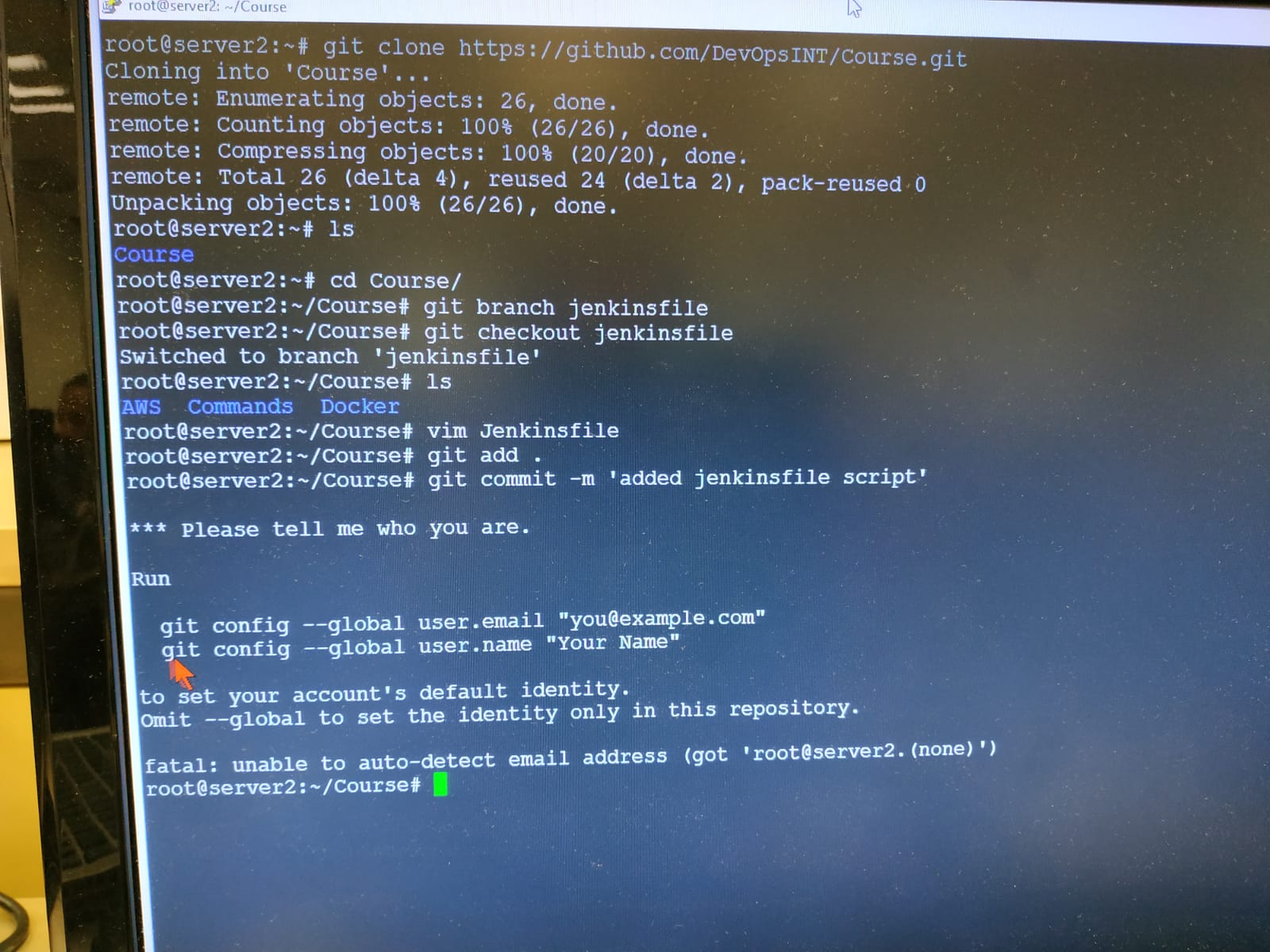
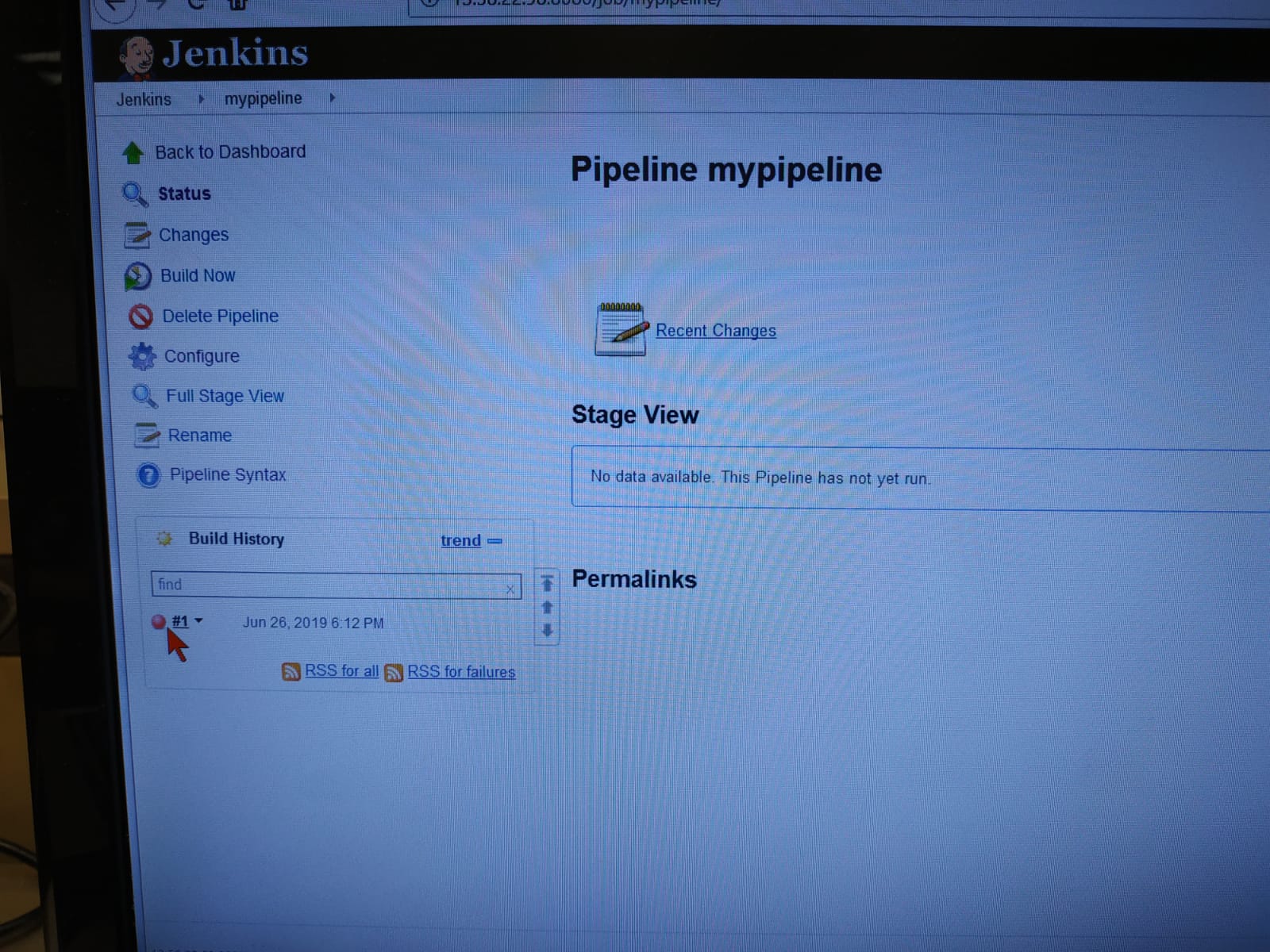
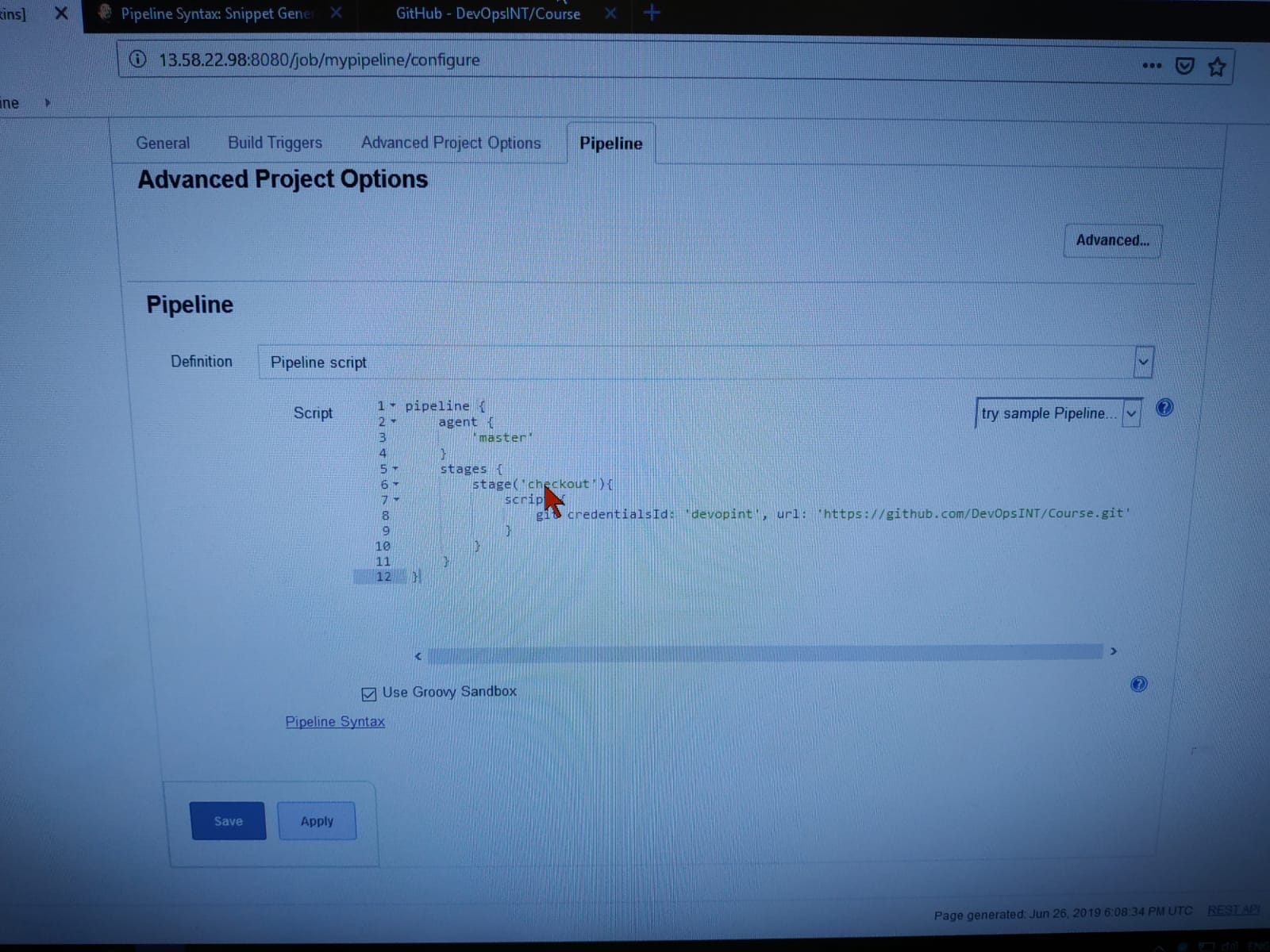
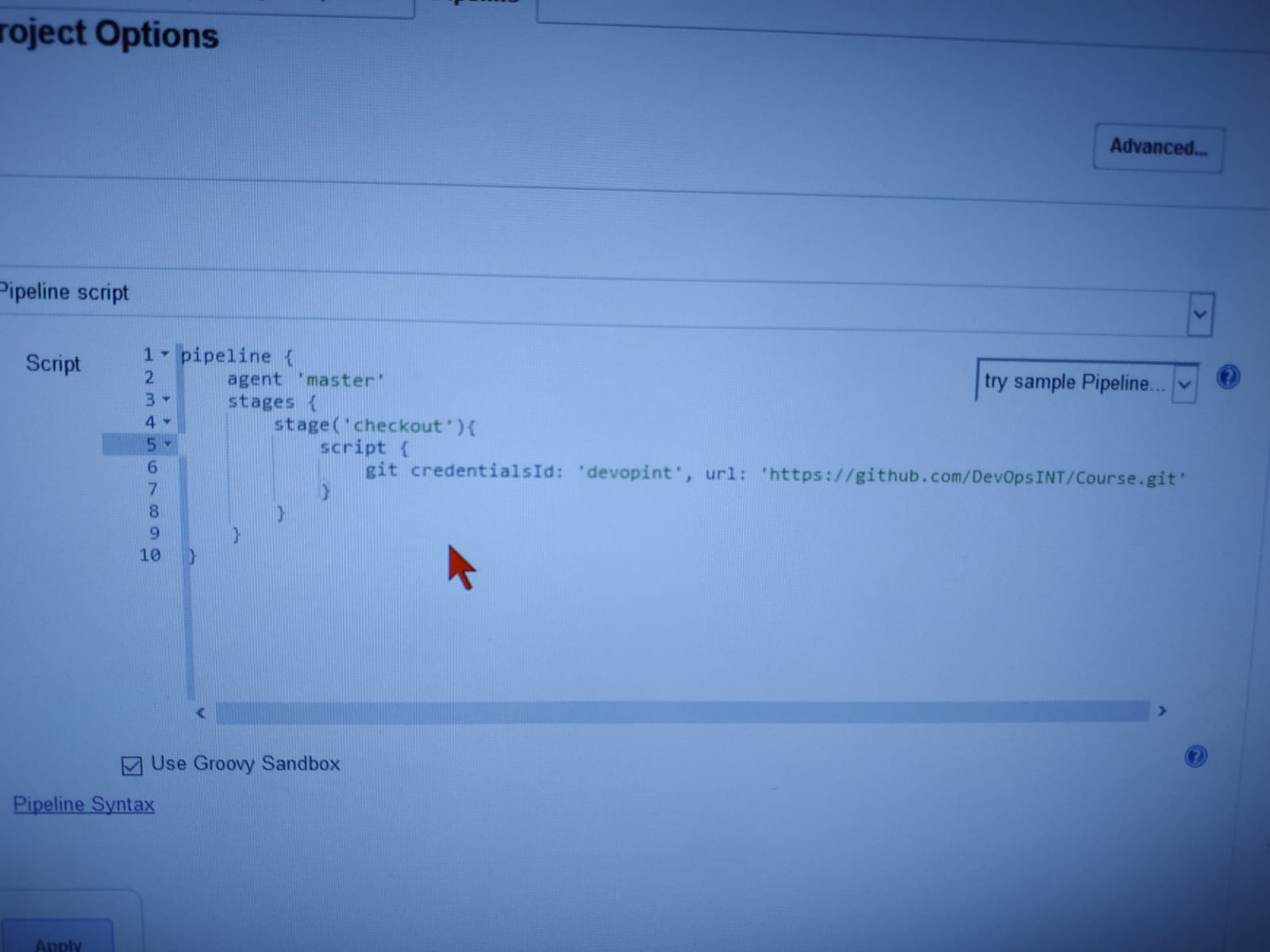
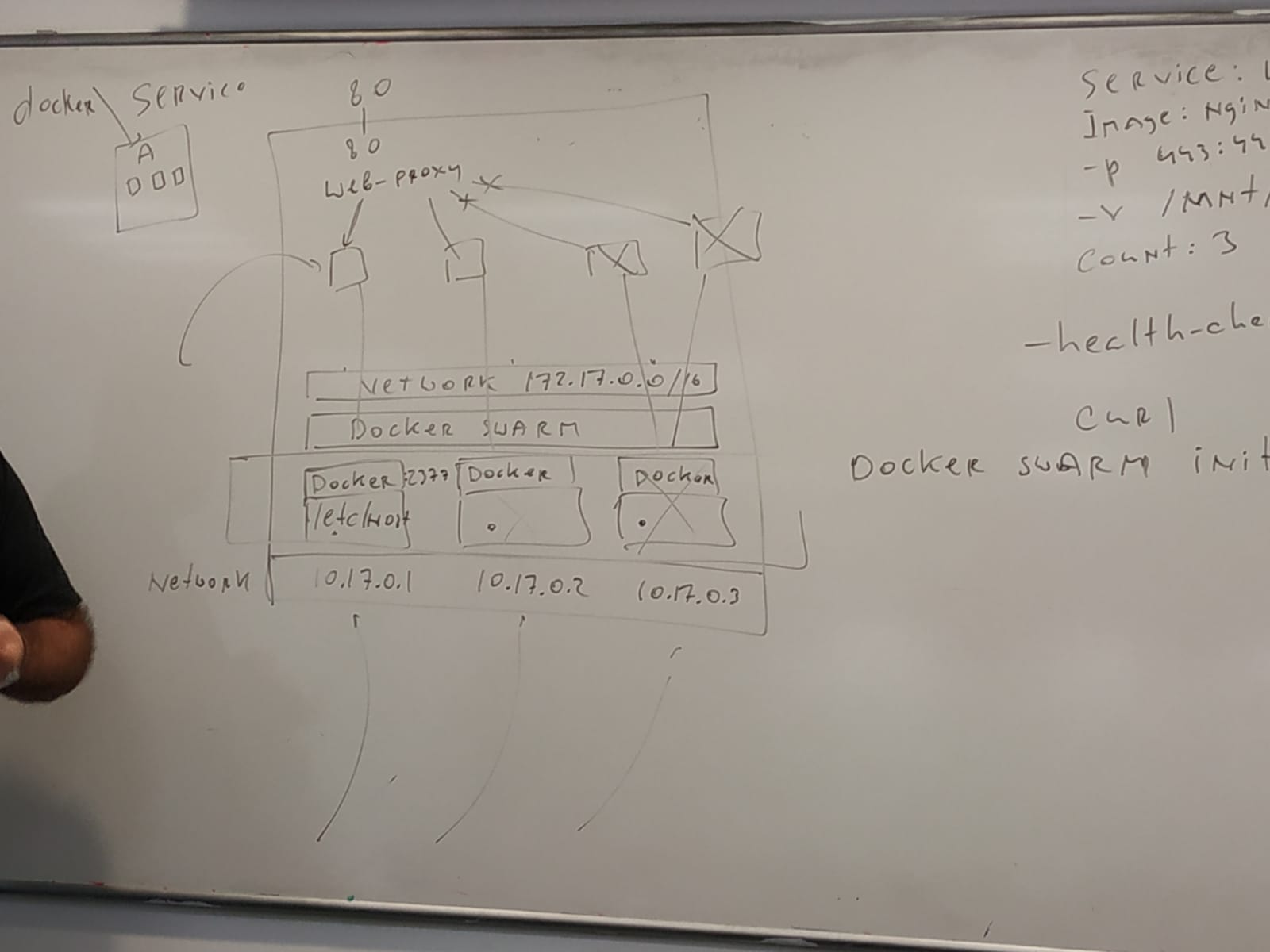
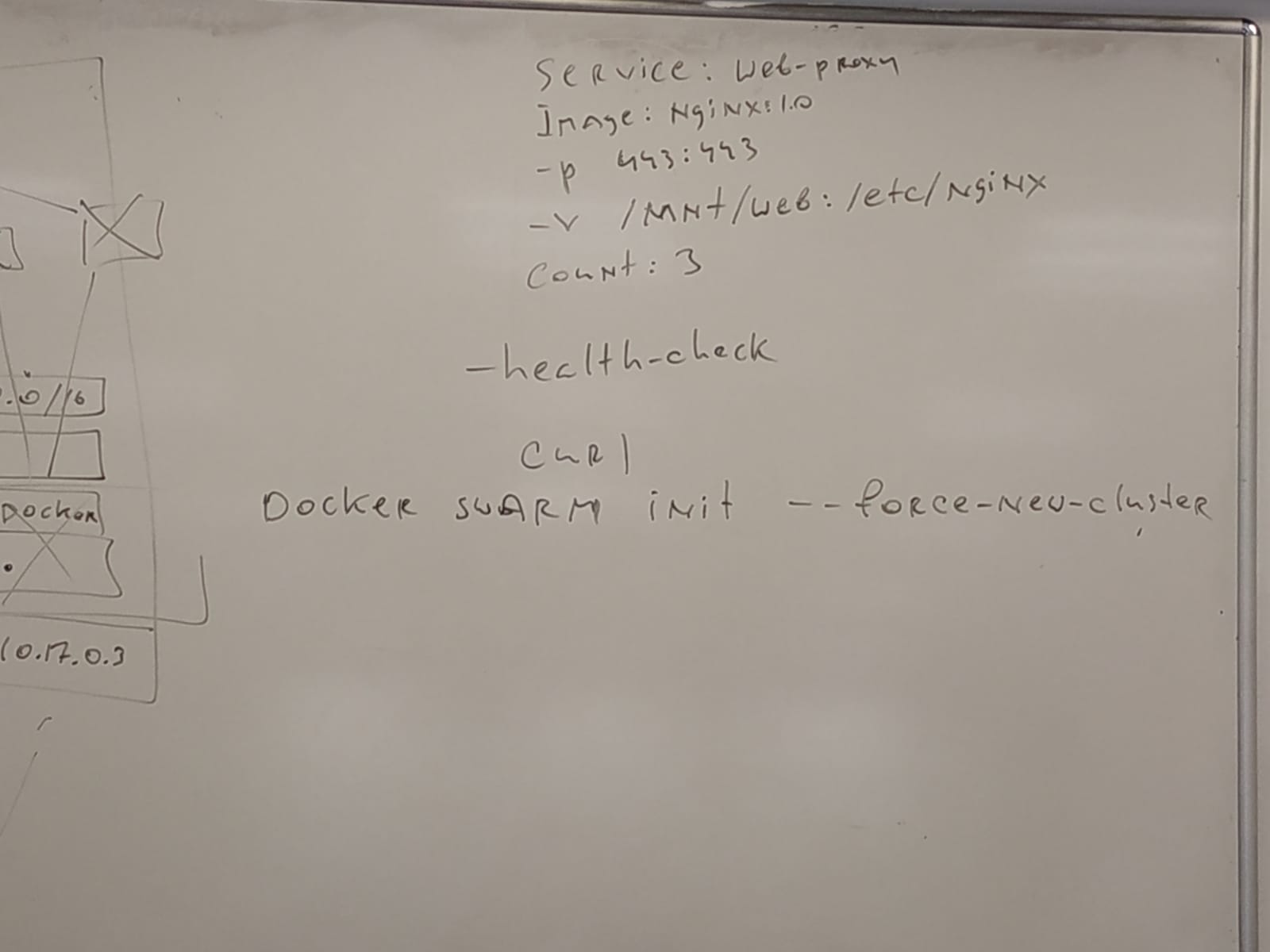
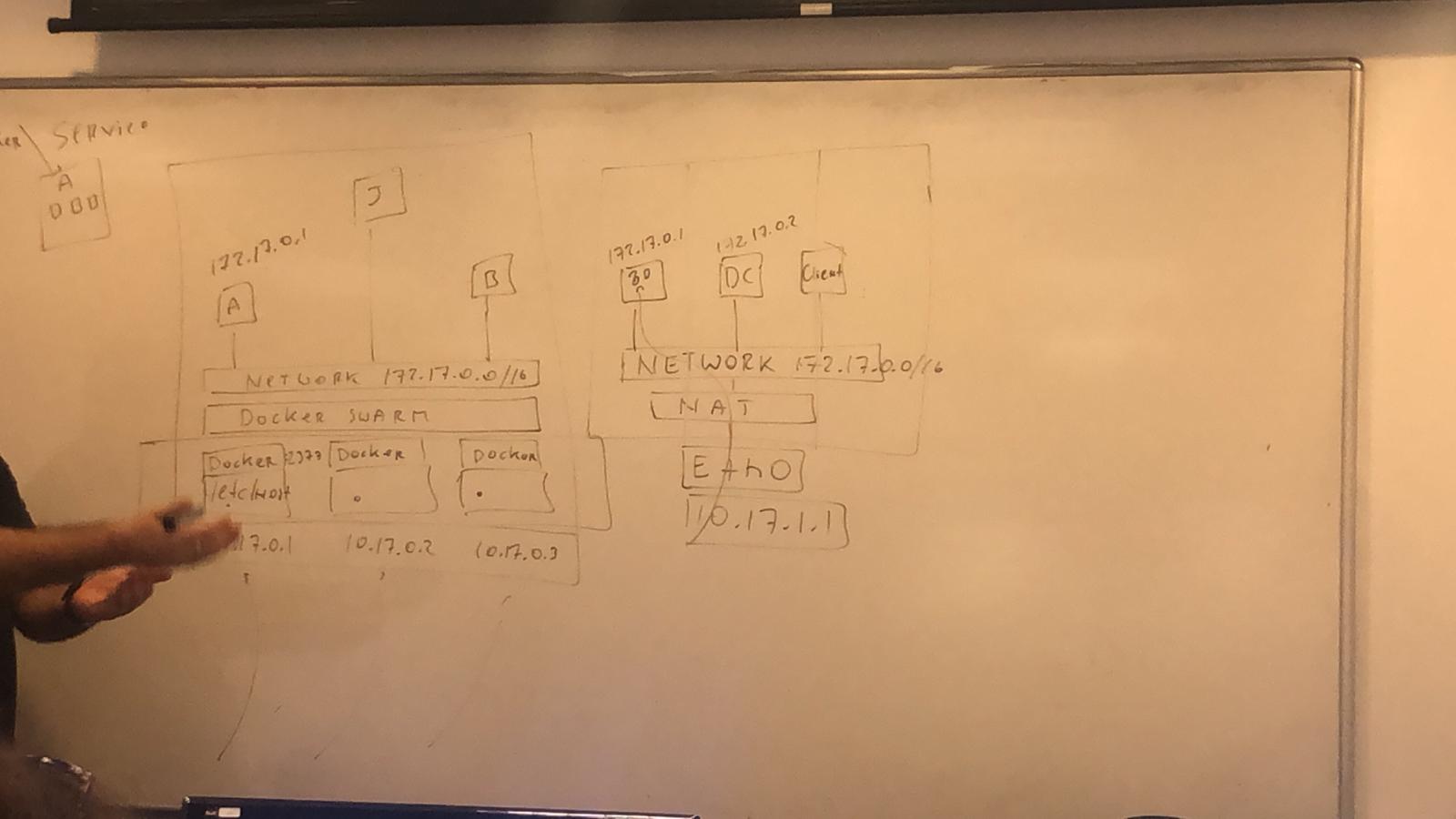
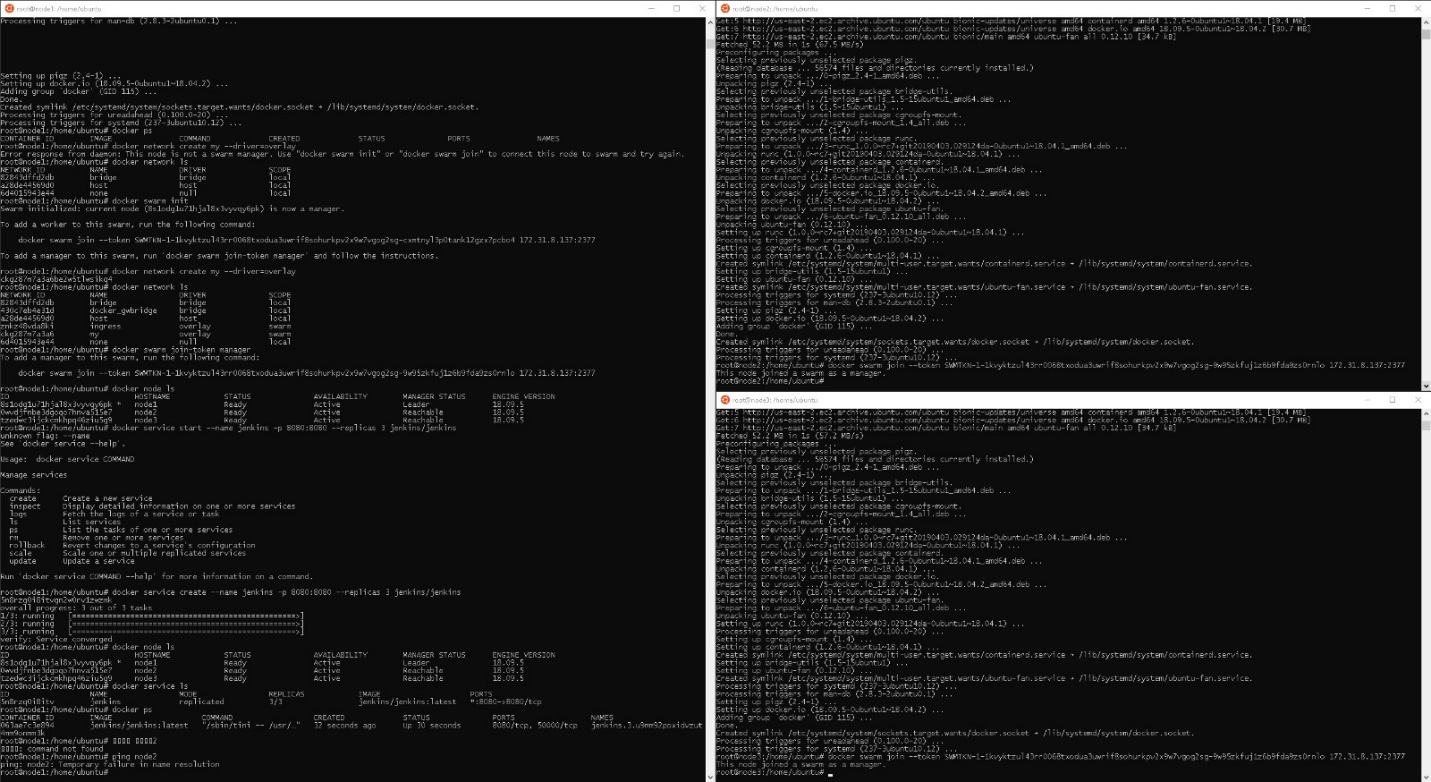
Jenkisci/jenkisci:lst

Docker inspect לראות את תאור האוביקט

netstat -tulpn בדיקת הפורט

גנקינס מביא את הקבצים בגיט ושומר אותם בתיקייה הבאה:

/var/Jenkins\_home/workspace/$job



קורס 3.7

Jenkins /var/Jenkins\_home

Credential

Free style

Pipeline

מדוע פייפליין עדיף ?

מבחינת גיבויים עדיף כי גם אותו ניתן לשמור בgit

Docker history <image>

אפשר לראות את כל הפקודות שהורצו

כדי להסתיר את הפקודות ניתן ליצר image ולאחריו לא ניתן לראות את הפקודות וגם יש חיסכון במקום .

Git

Repository

Branch : git clone http/ssh

Feature or hotfix

Tag – הגדרה של נקודת זמן שמורה שניתן לחזור אליה

Git branch 1.o

Git checkout 1.0

Git add .

Git commit -m “message “

Git push source destination

Git push origin 1.0

--- GIT commands

1. git status : check status of the files

2. git add <path> <filename> (. for all)

3. git commit -m "<message"

4. git log --author="<name>"

5. git diff - shows changes between files in the staging area and the working area

6. git diff --staged - changes between the repository and the staging

7. git pull

8. got pull origin <"GitHub url"

9. git pull origin <branch name>

10. git push origin <branch name>

11. got branch <branch name>

12. git checkout <"branch name">

13. git push -u origin <branch name>

14. from master - git merge <branch name>

15. git reset HEAD <file name> - revert the file from the staging area to the working area

16. git reset HEAD~ - reverts the last commit and pulling it back to the working area

17. git checkout -- <filename> - Discard the last changes within the file

18. git clone <link>

19. git log --oneline --graph --decorate

20. git log --since="2 days ago"

21. git show <commit id> - see log for commit

22. Create command quick shortcuts - git config --global alias.history "log --oneline --all --graph --decorate"

23. .gitignore file will exclude files in the project.

25. got branch -m "name" - rename branch

משתנים בpipeline :

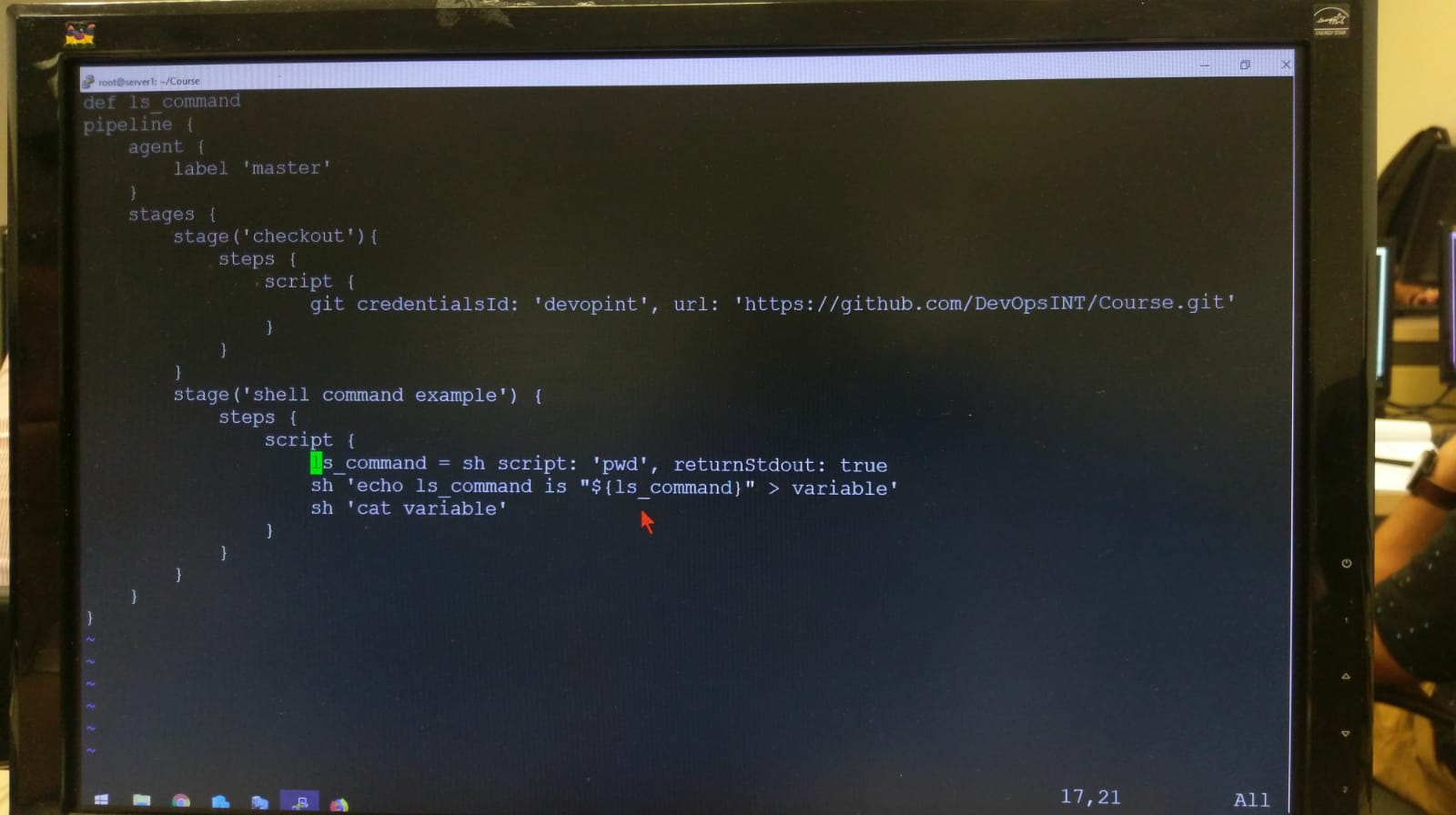
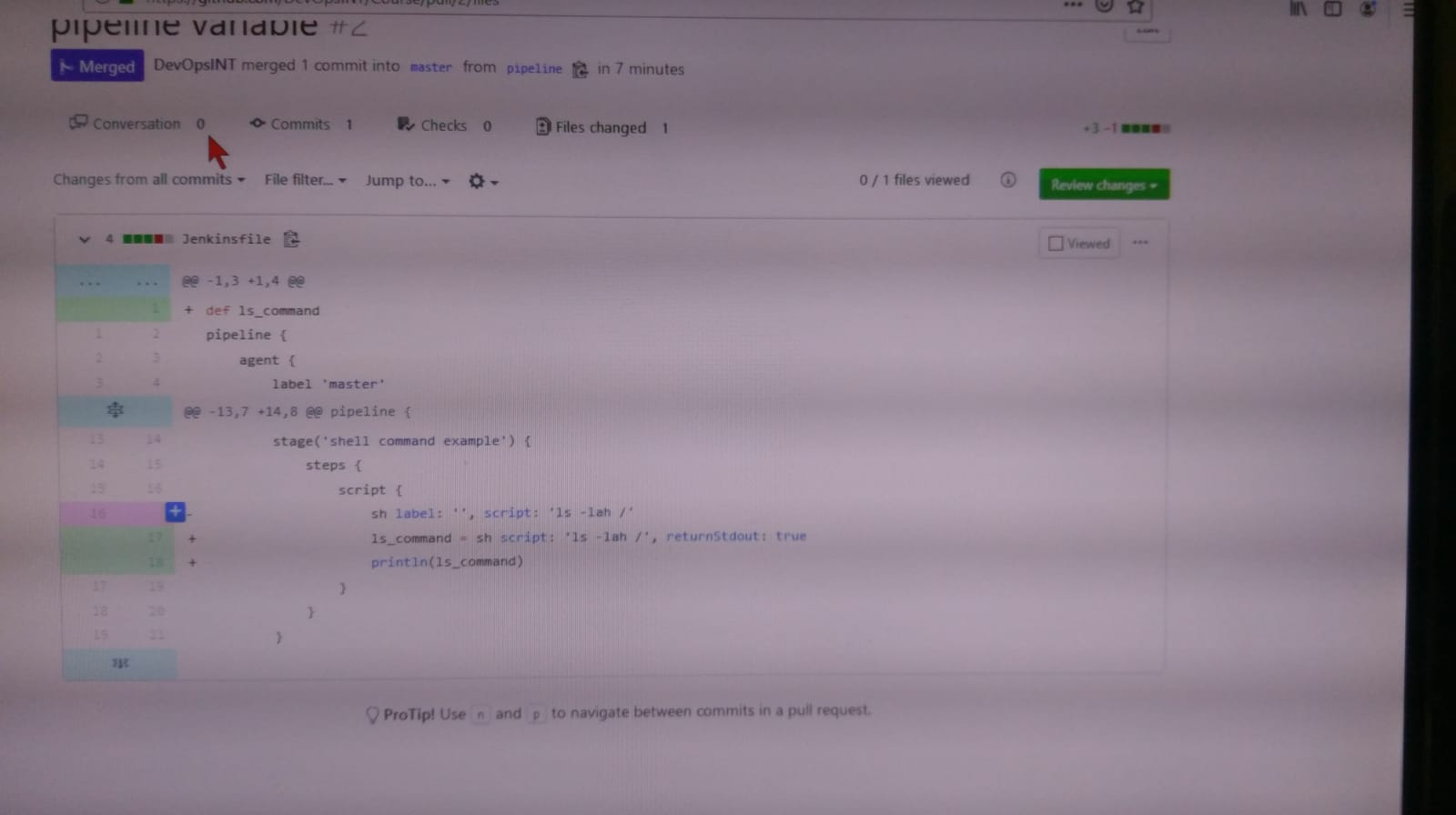
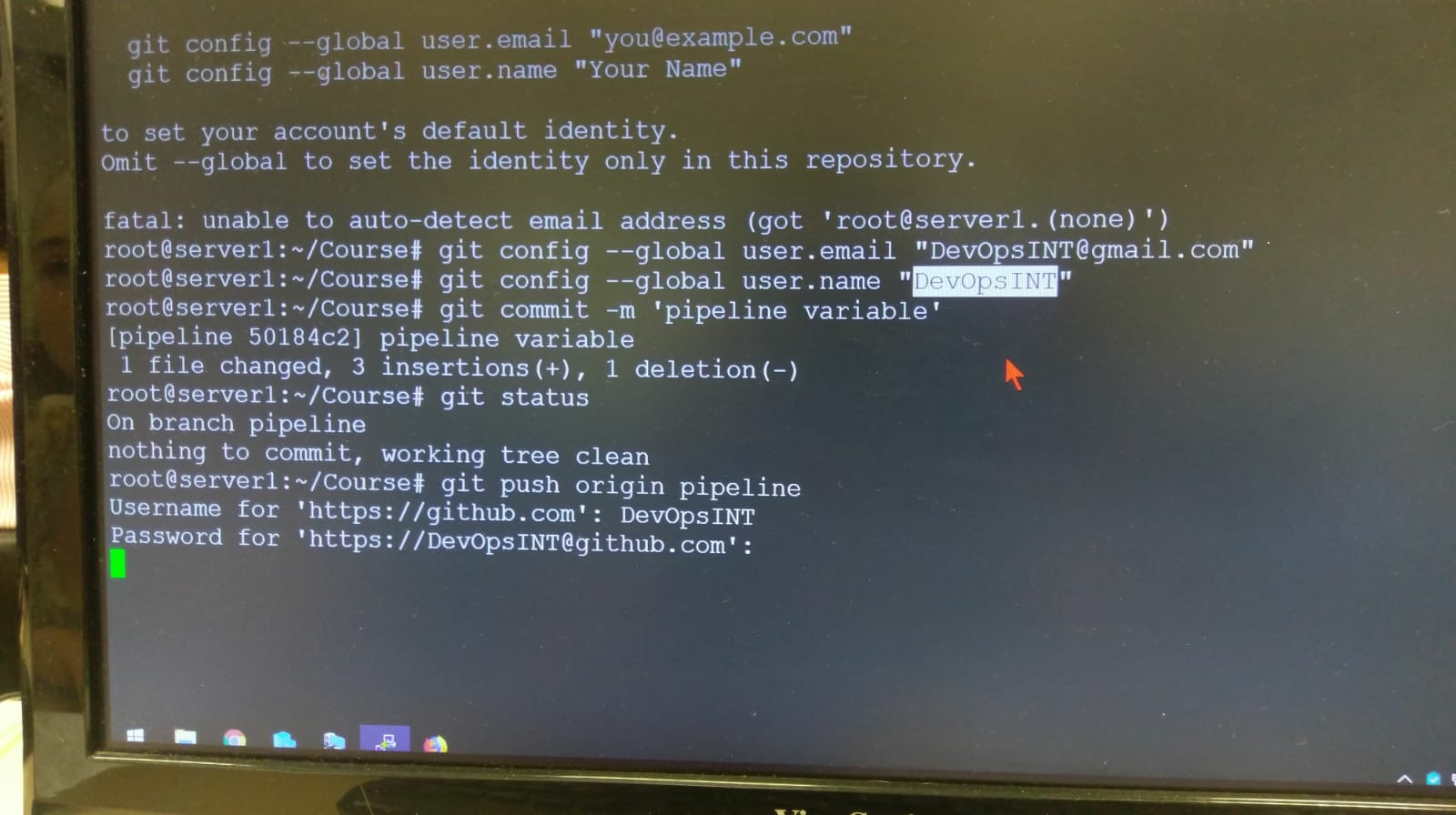
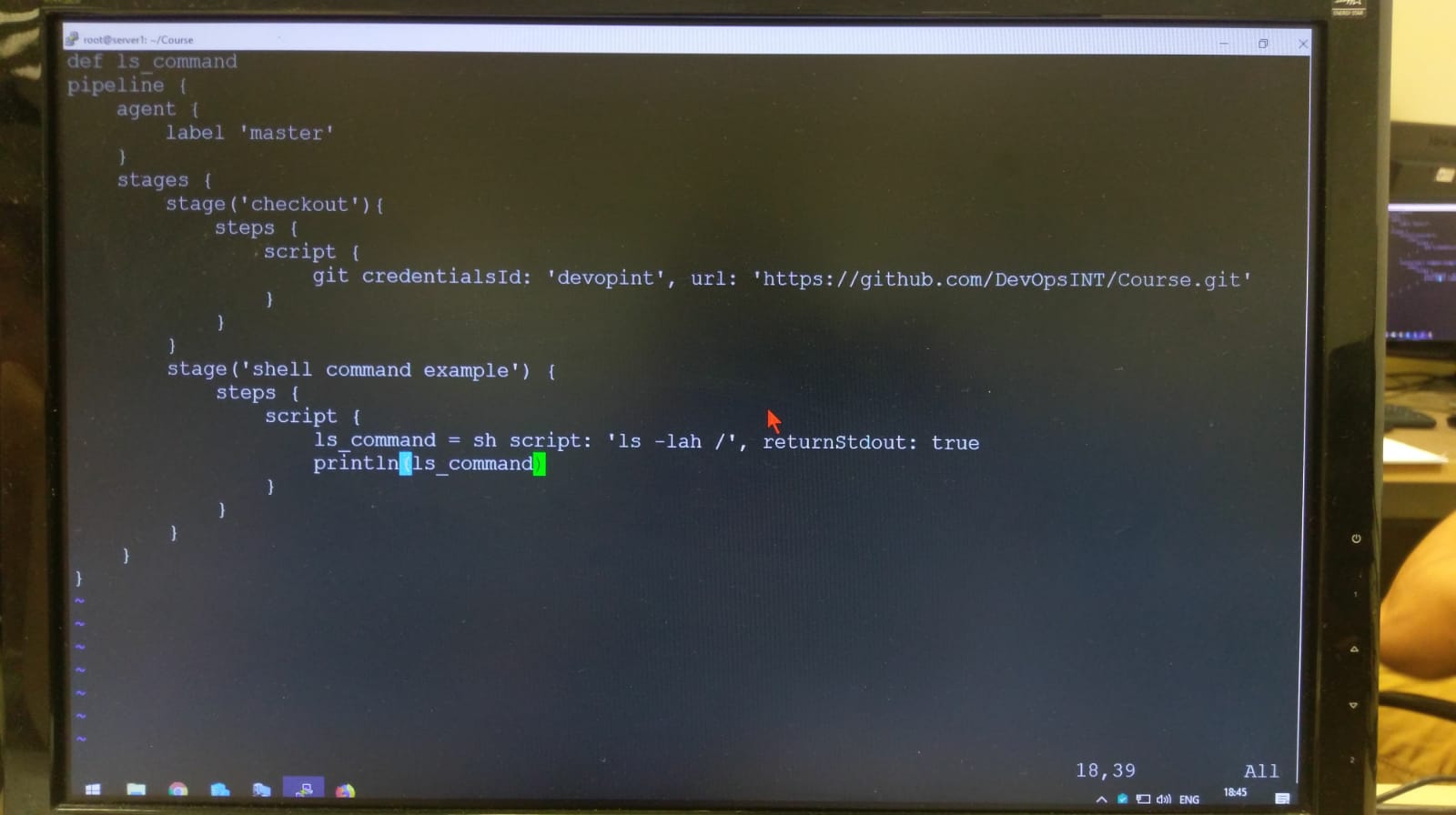
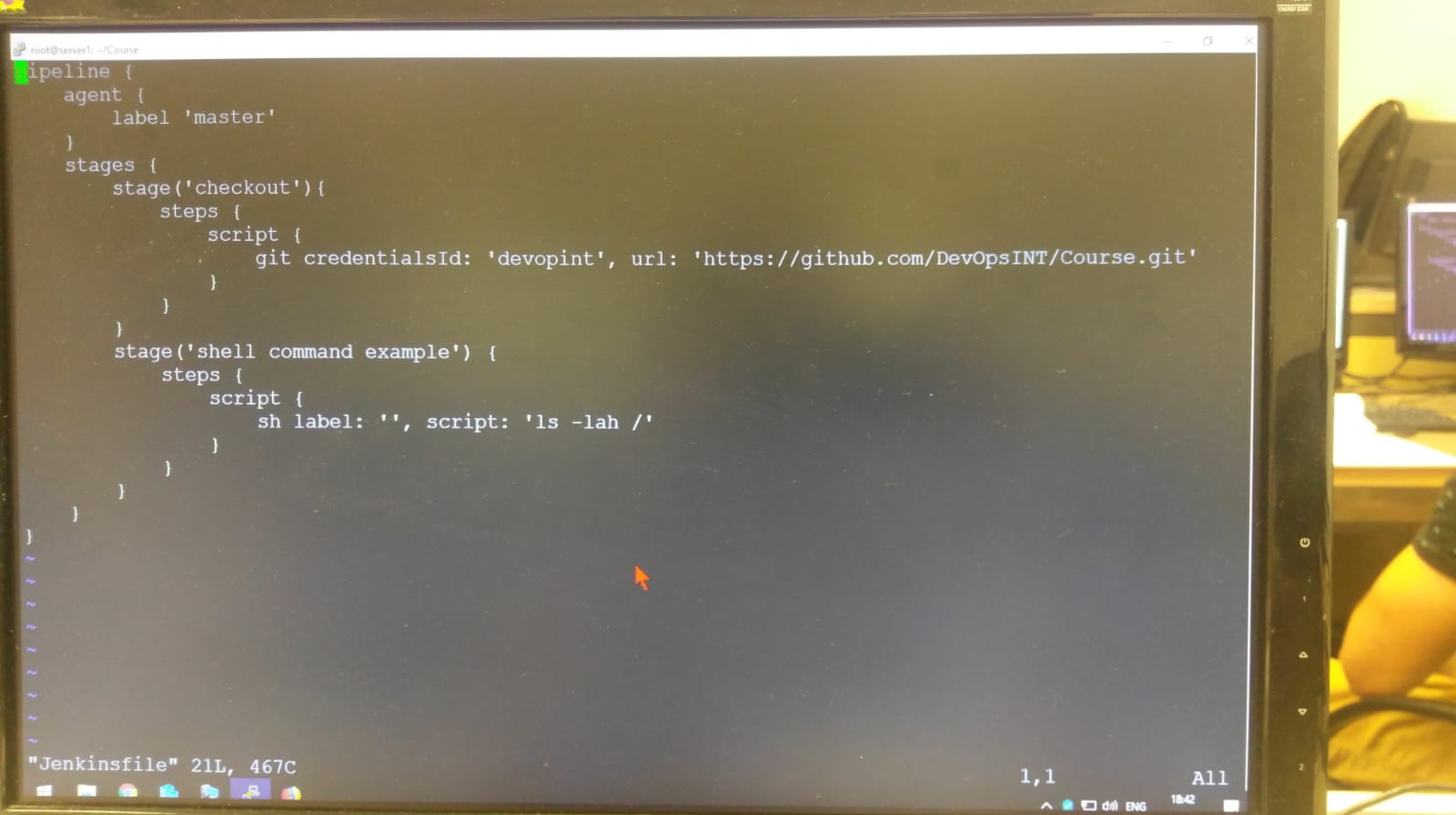
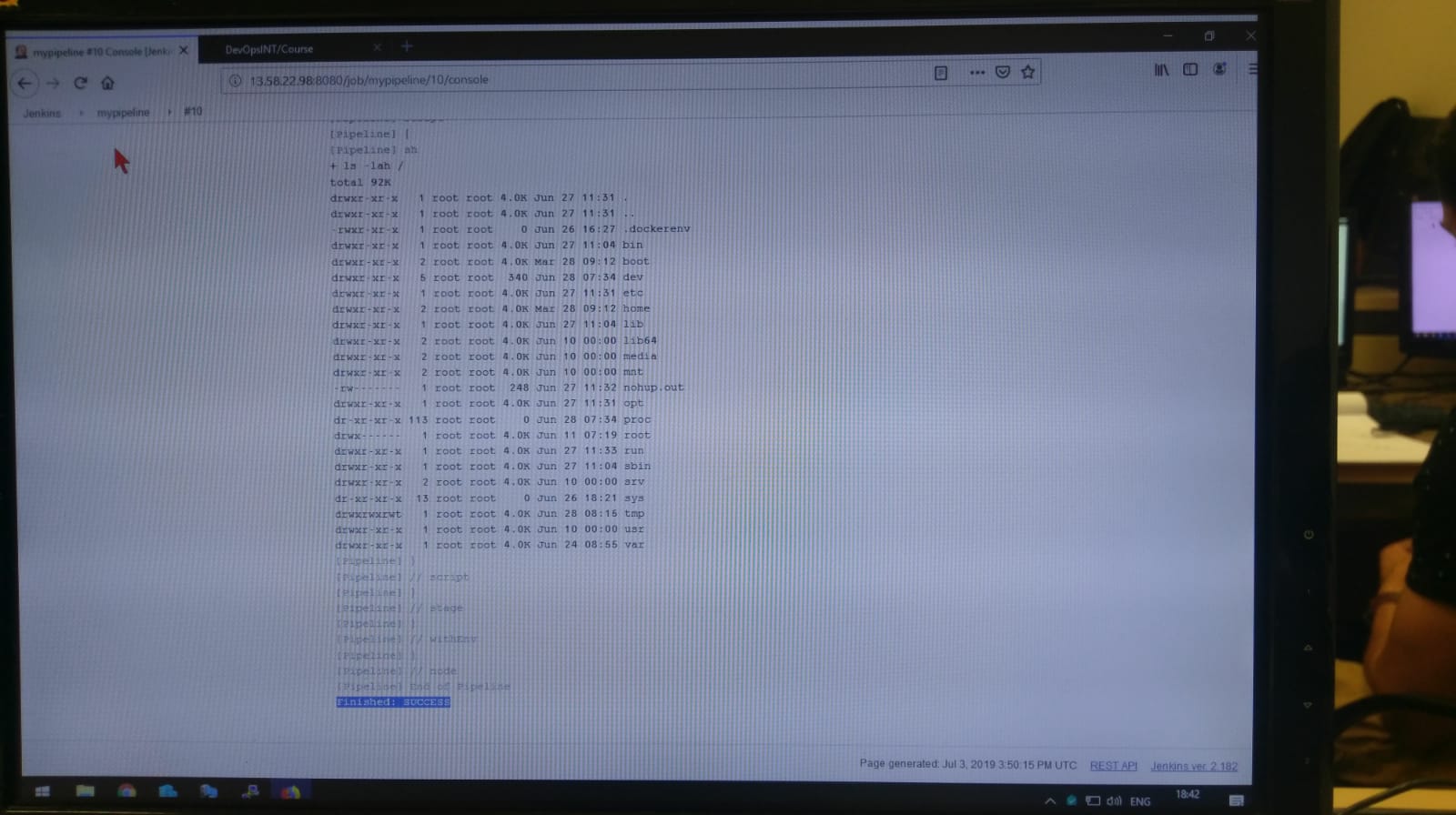
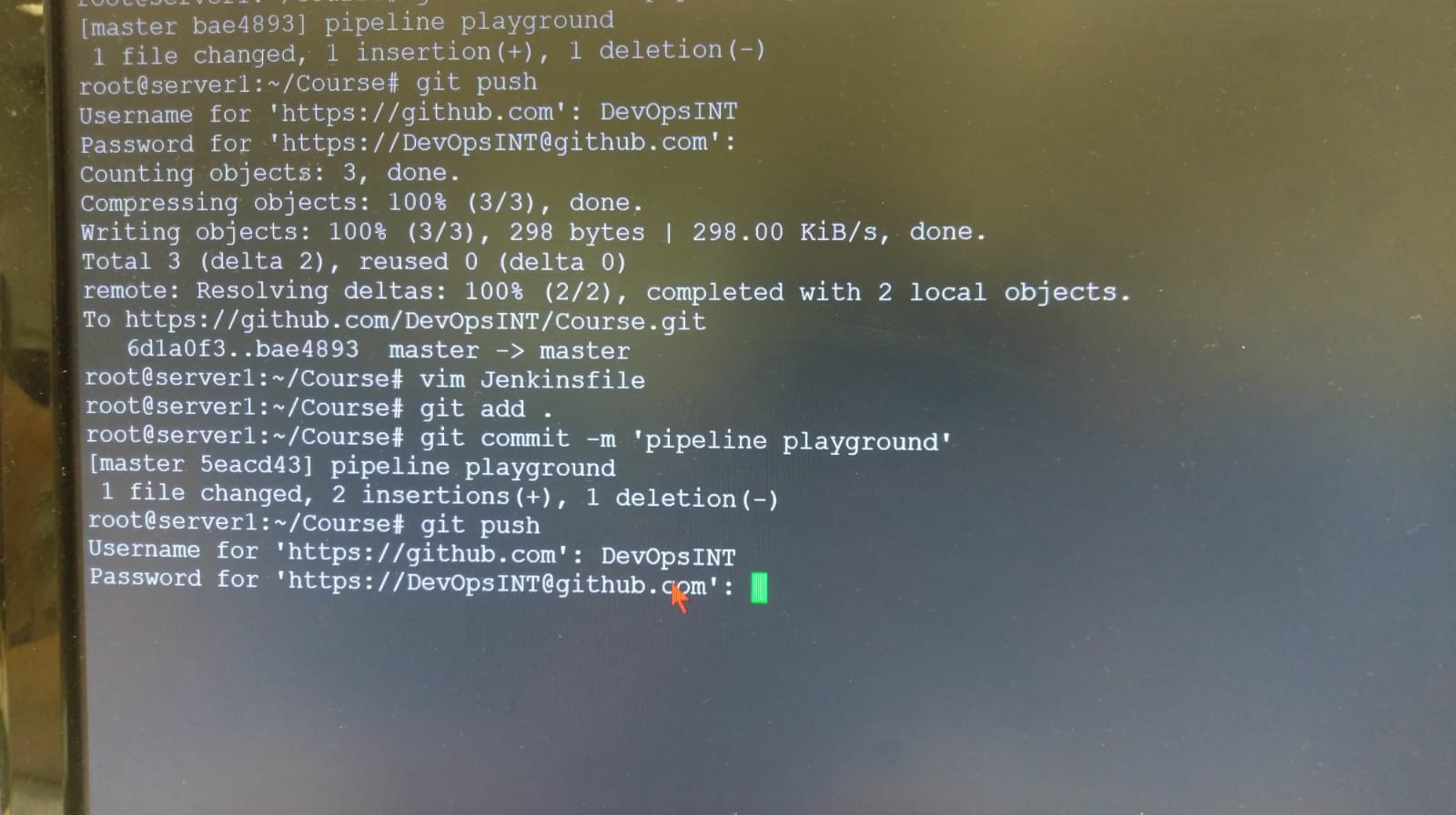
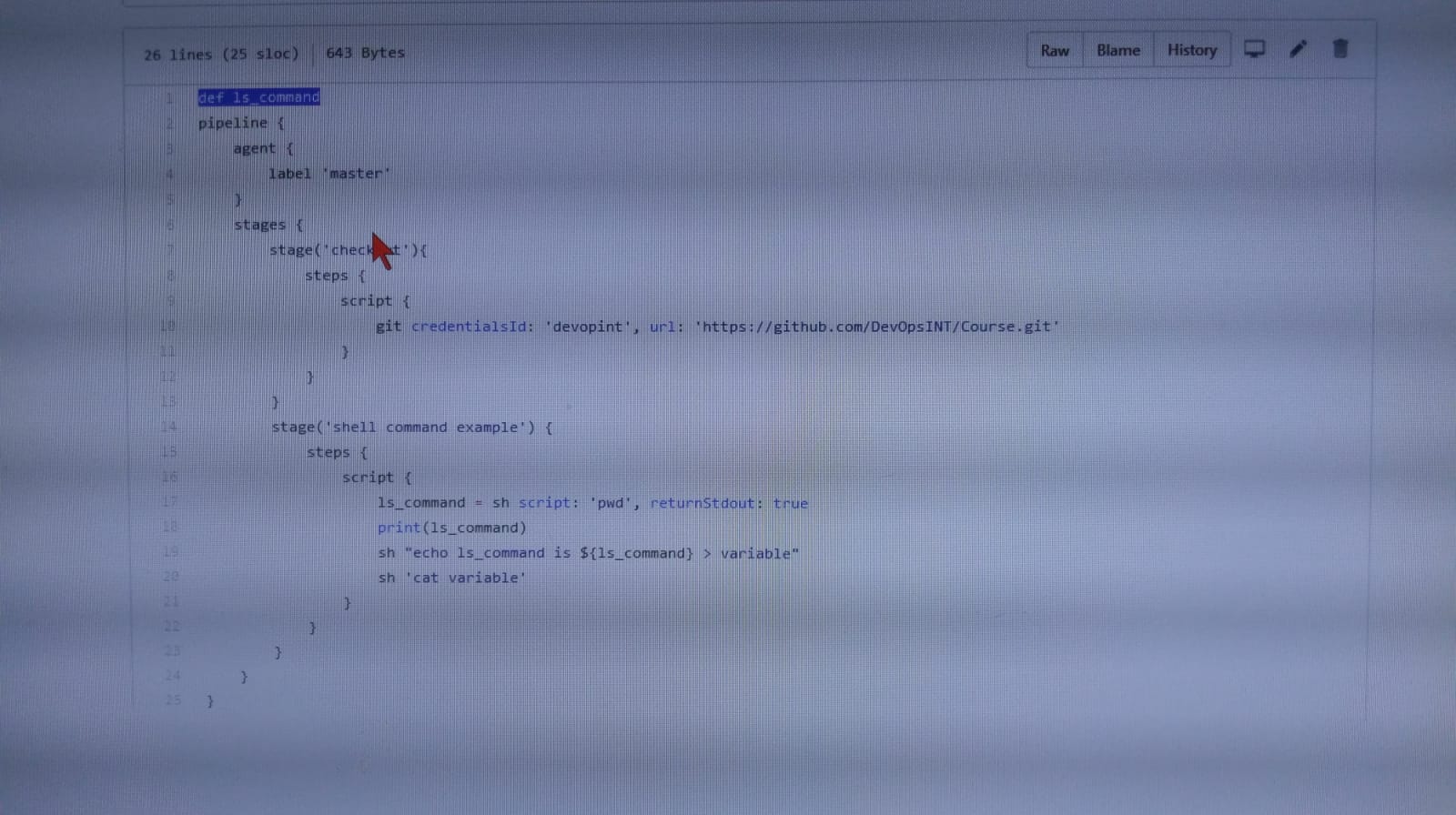
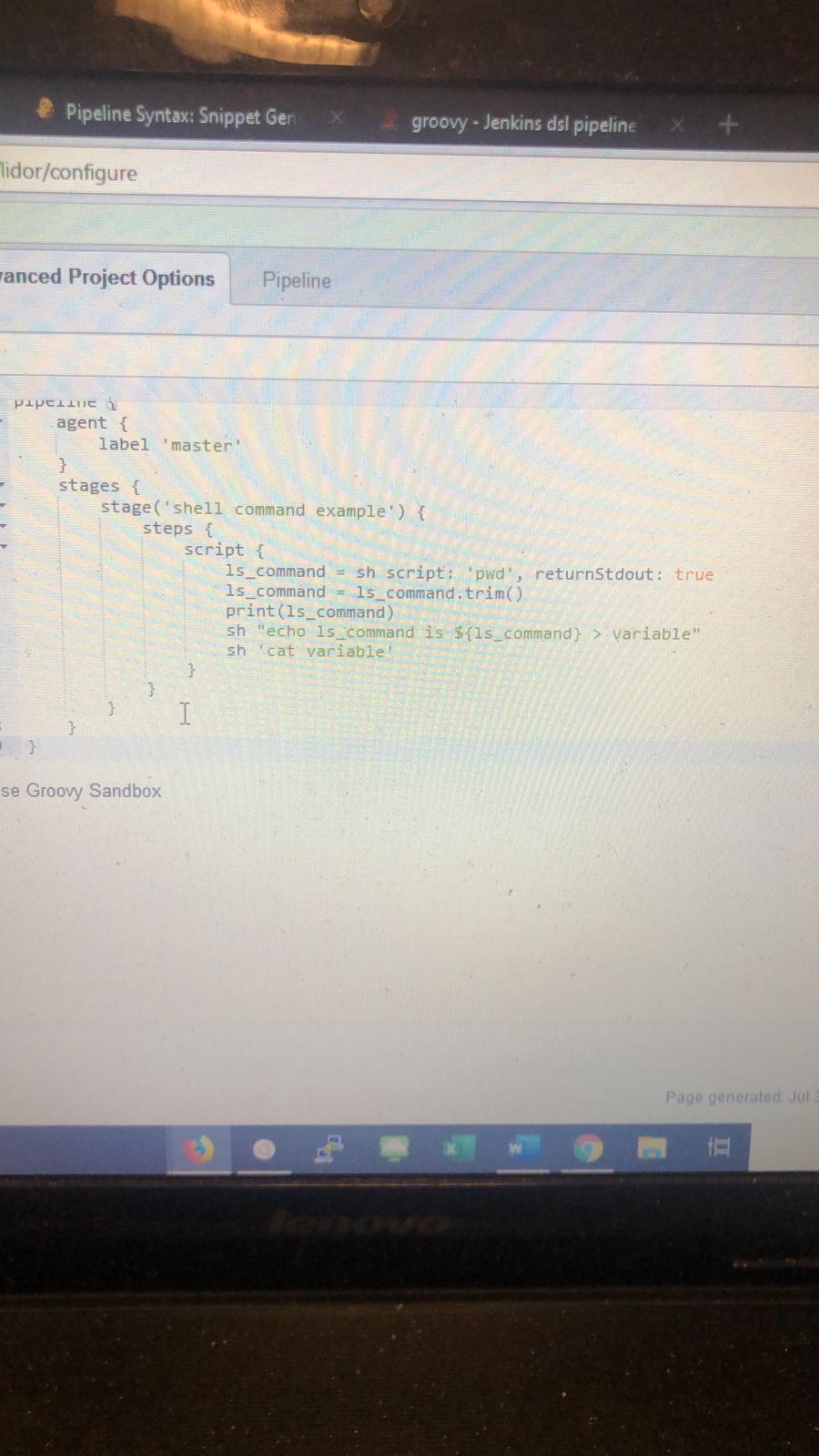
Def <name>

Groovy השפה שבה נכתבים קבצי הגנקינס דומה מאוד לjs

Jobs :

/var/Jenkins\_home/workspace/yuri

Folder for every job and version for it for each run the default 20 .



קורס 7.7

פתרון שיעורי הביית :

1. Withcredenfield
2. Sh `git push [https://${user}:${pass}@github](https://$%7buser%7d:$%7bpass%7d@github)`

<https://git-scm.com/book/en/v2/Git-Tools-Credential-Storage>

שיטה נוספת :

Git config user/pass

<https://git-scm.com/docs/git-config>

Infrastructure as a code

כל האפליקציות צריכות להיות התשתית בעבר יצרו אימגים פר חומרה וגירסה ואז הגיע הווירטואליזציה והטמפלייטים .

עדיין היה ניהול של הרבה טמפלייטים ושרתים התהליך לא נוח ומסורבל .

איך מנהלים את כל השרתים .

Declaration /Manifest

Install docker

Start docker service

Copy files

Create user

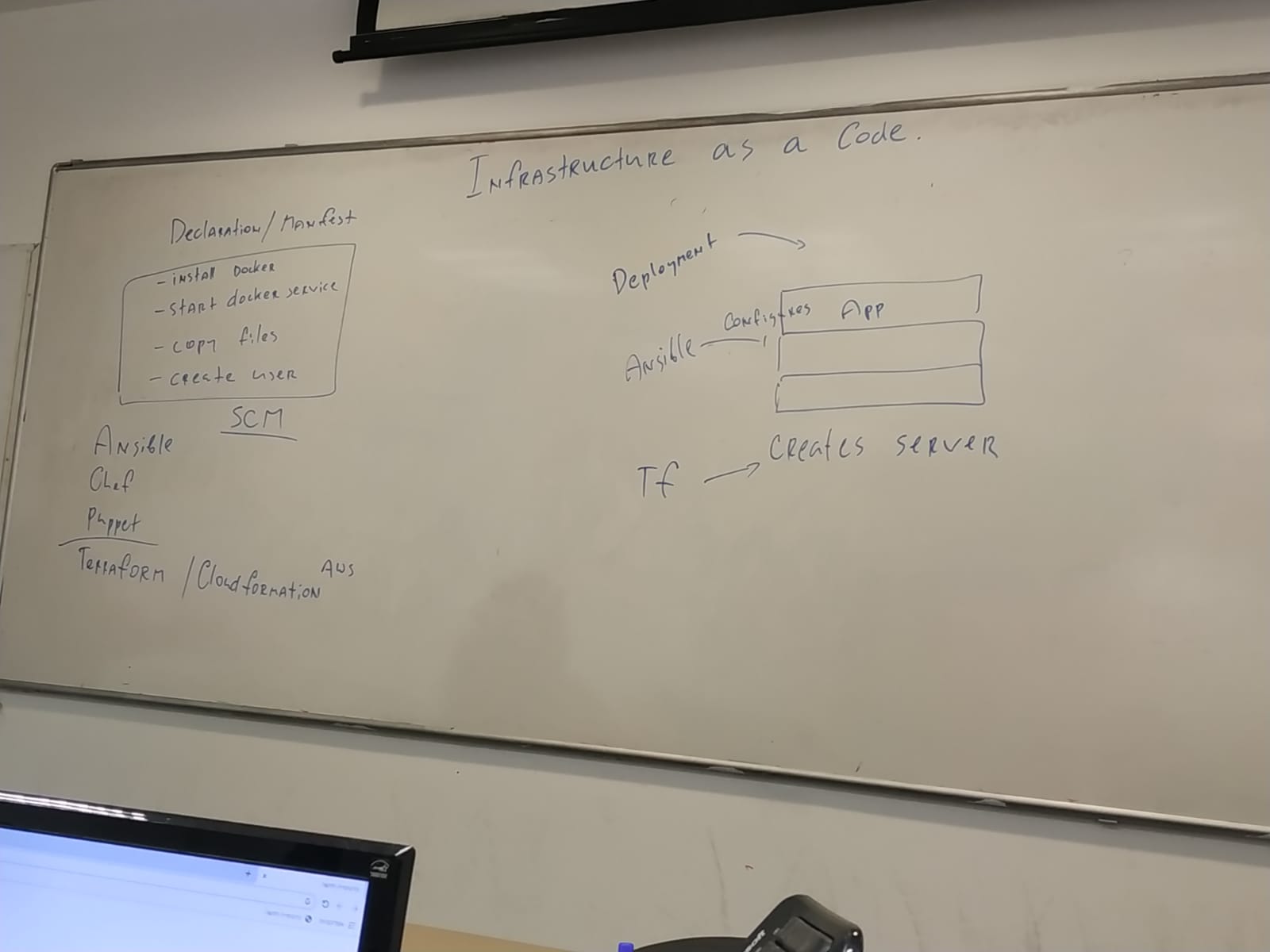
SCM

Ansible

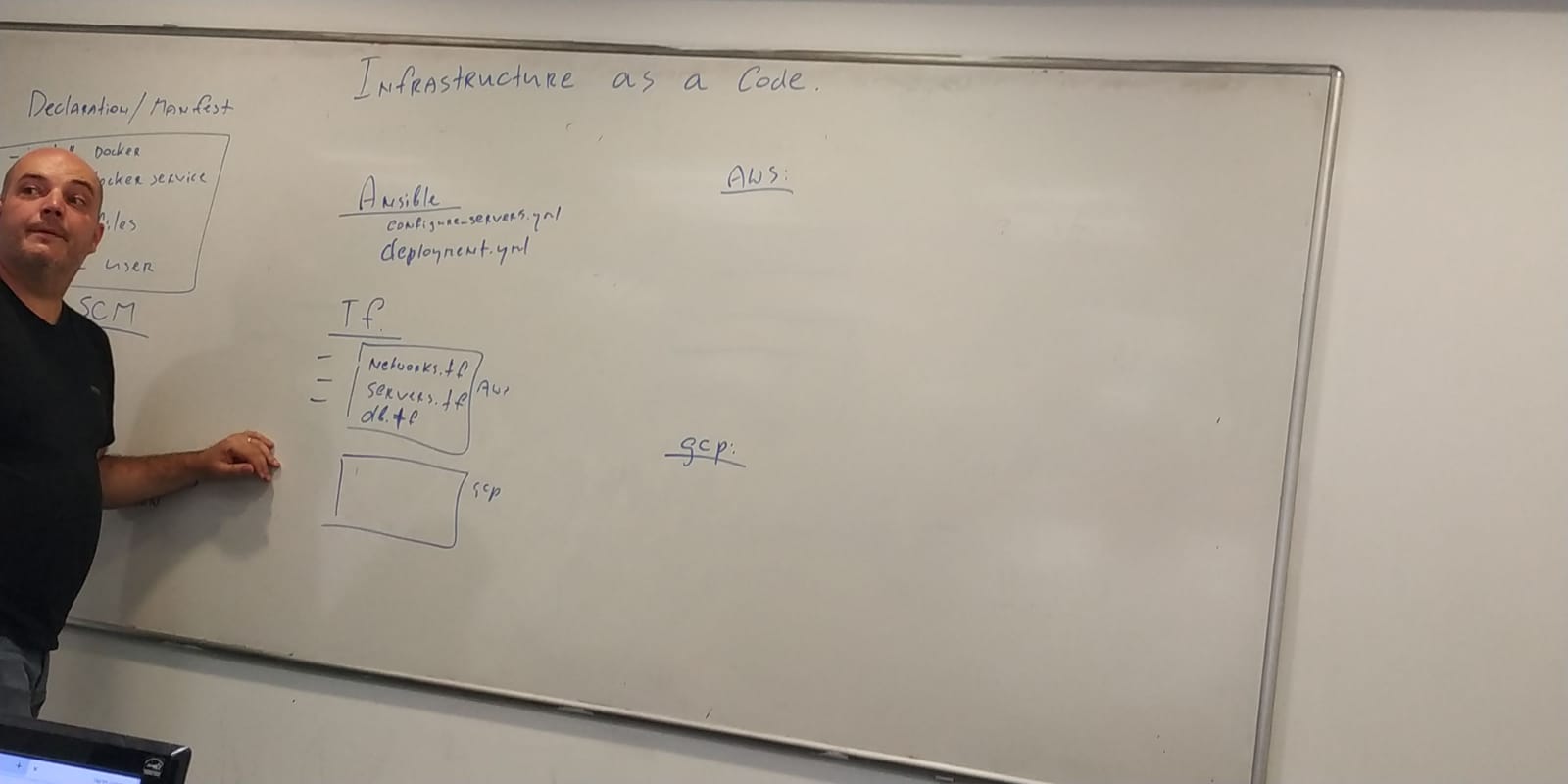
Chef

Puppet

TerraForm / CloudFormation (aws)



ברגע שיש שינוי בשכבות השינוי הוא מינורי .



היתרונות של אנסבל :

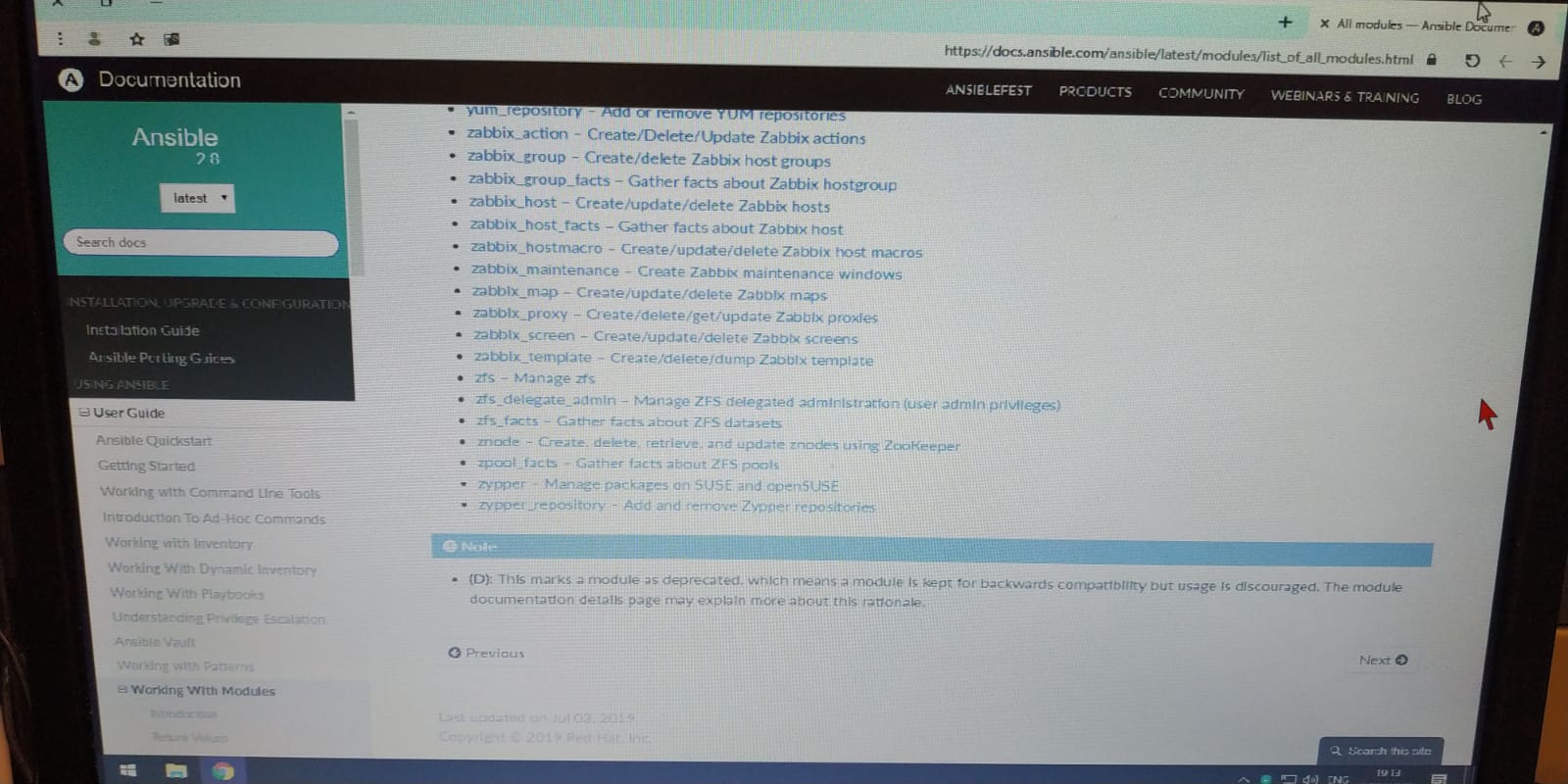
נותנים לו inventory והוא רץ עליו בקלות ויש לו תמיכה של redhat

החיסרון צריך להיות מודע לסטייג של הקליינט .

Playbook.yml : אוסף של Tasks

idempotent : נכתבים ללא זהות לא משנה מה מצב המקור ונגיע לאותה התוצאה

Roles : מורכב מvariables playbook files



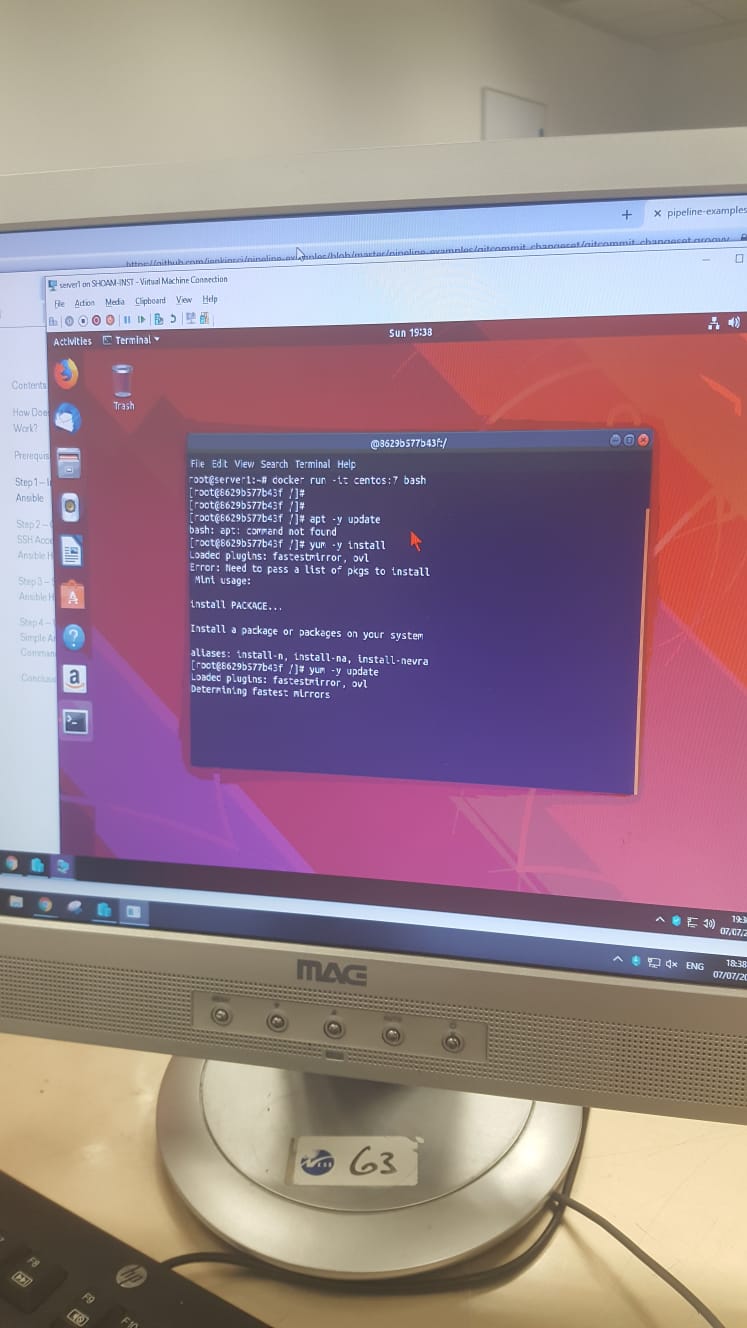


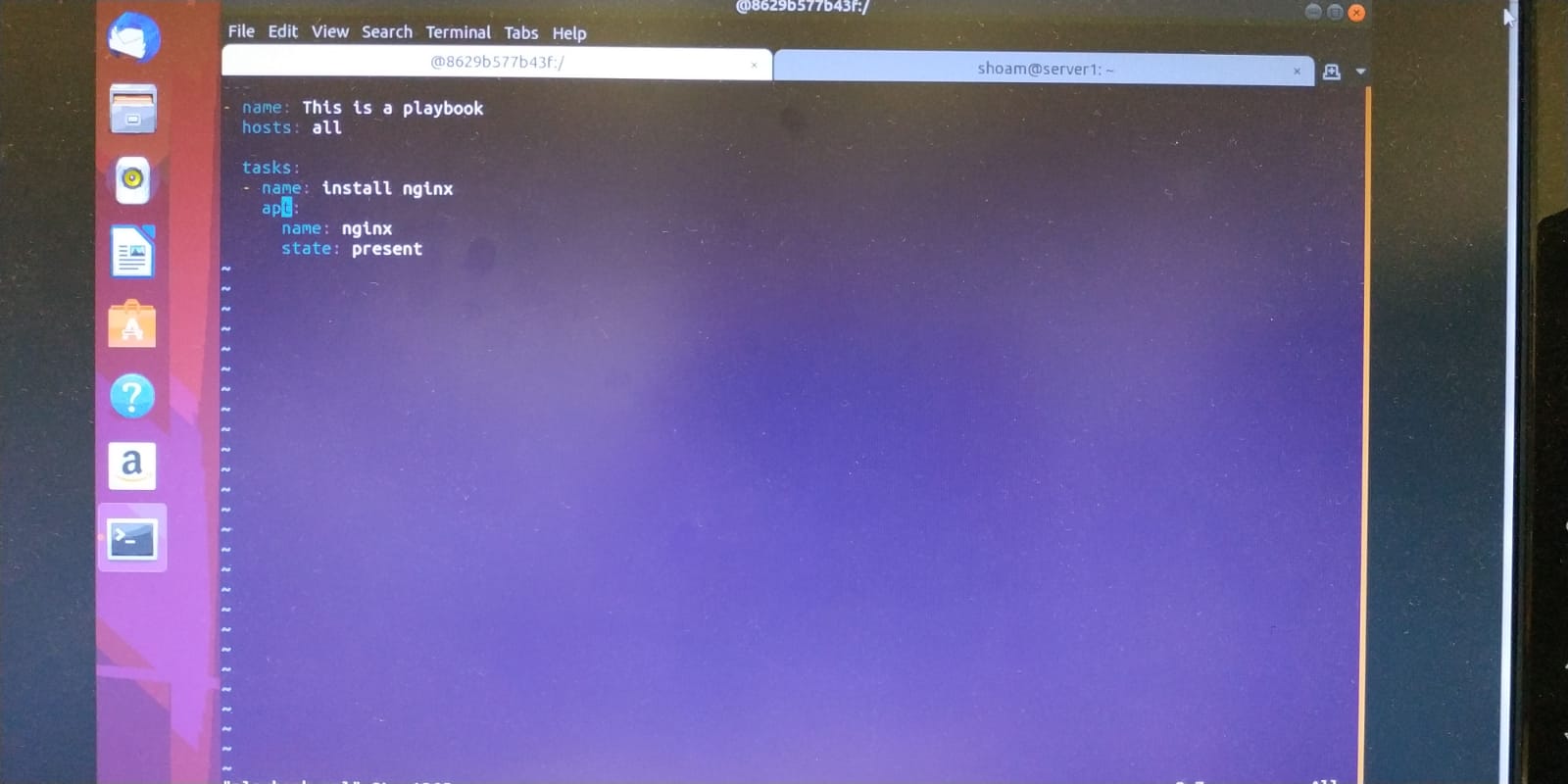
Inventory : החיסרון עבודה עם כתובות ip

Command line

Inventory file

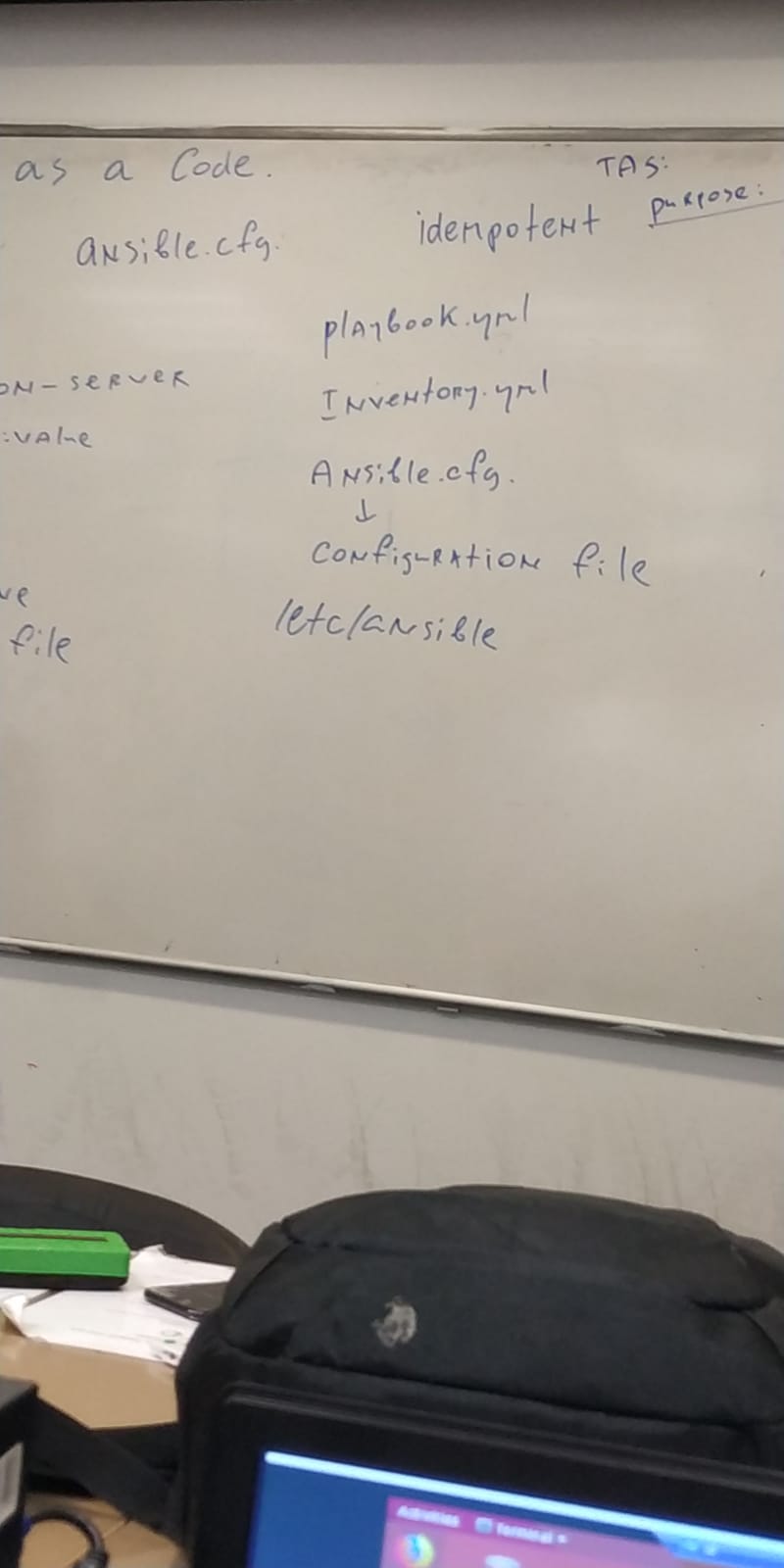
האנסבל מבוסס פייטון



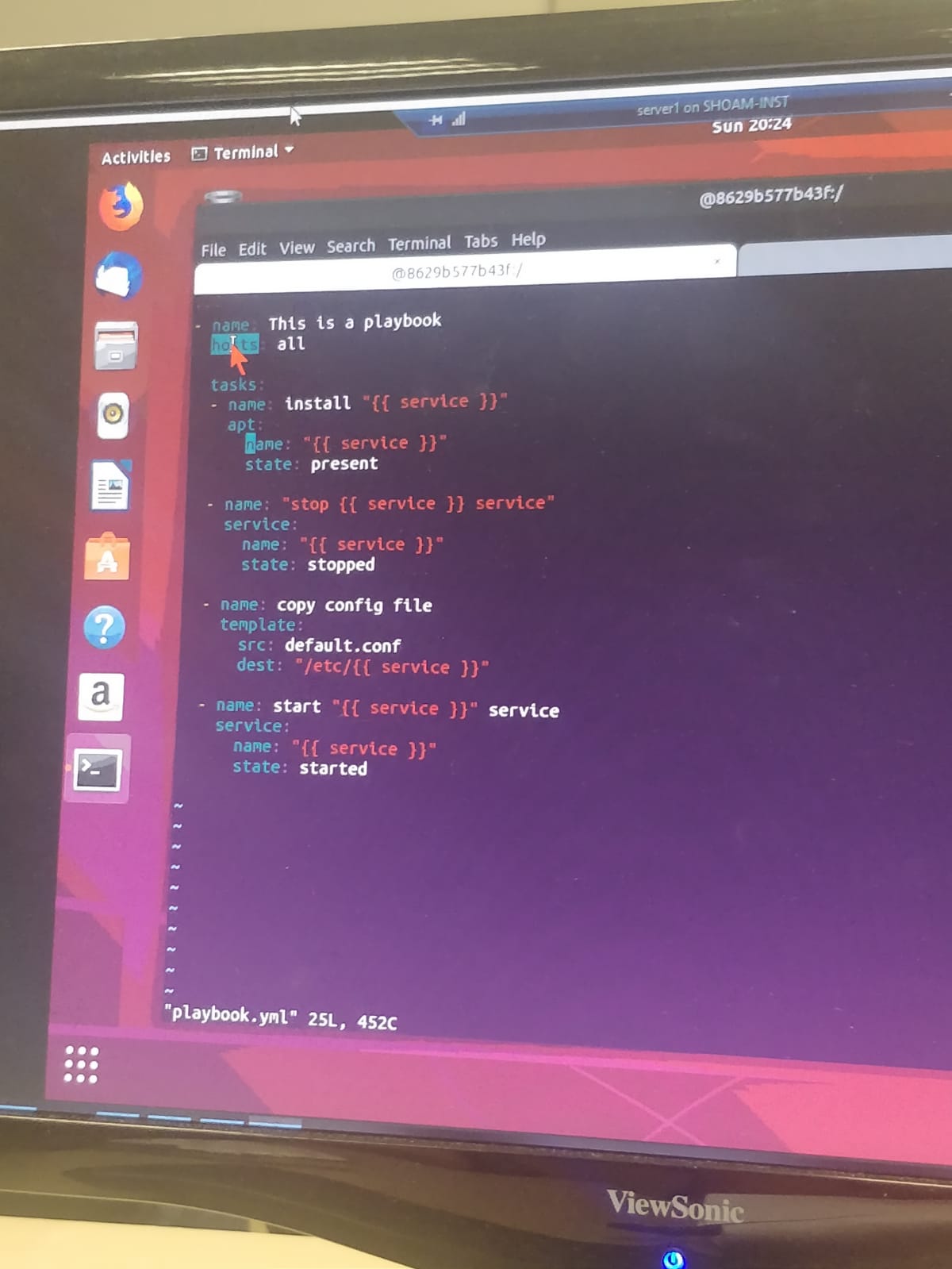
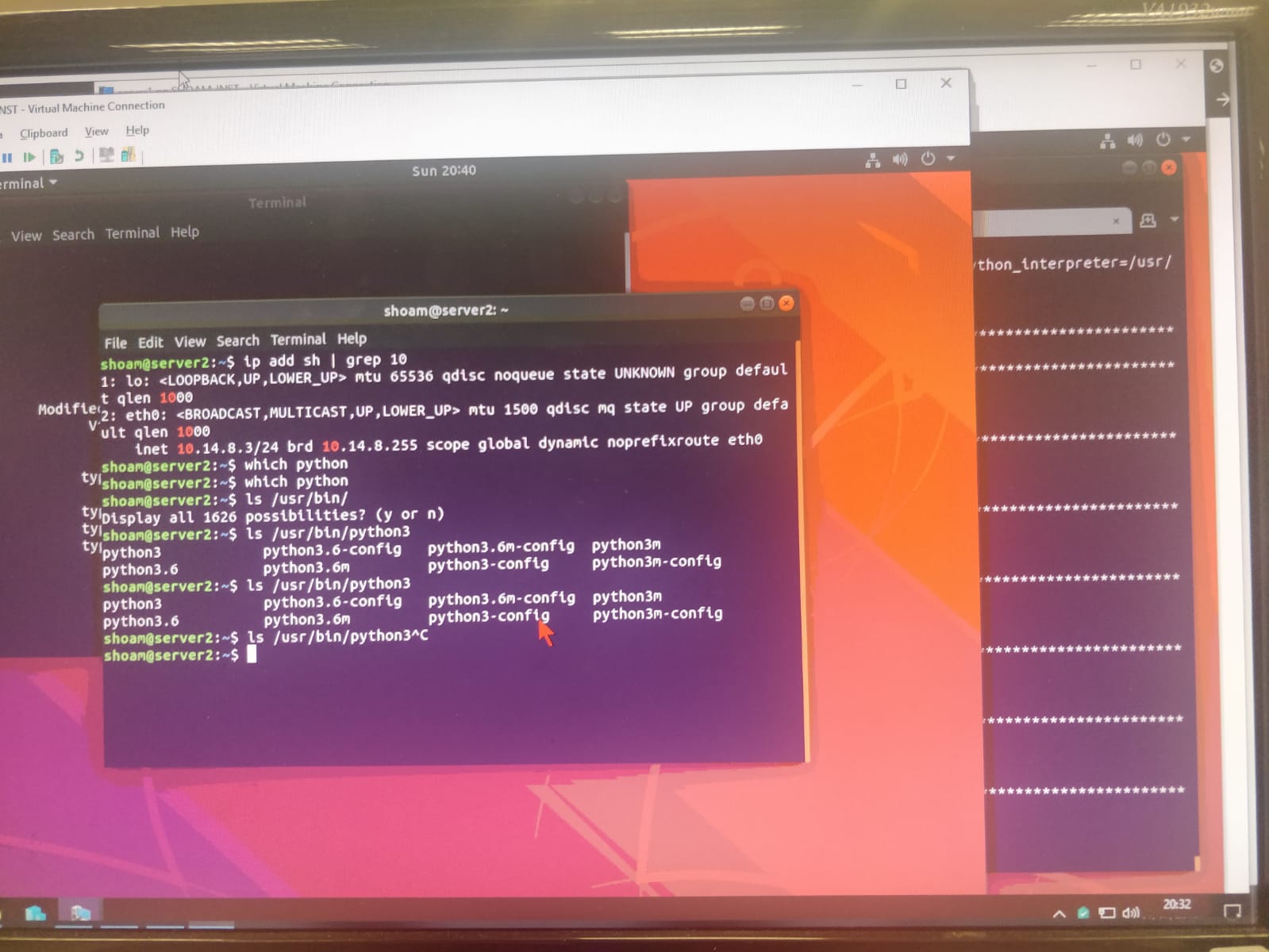
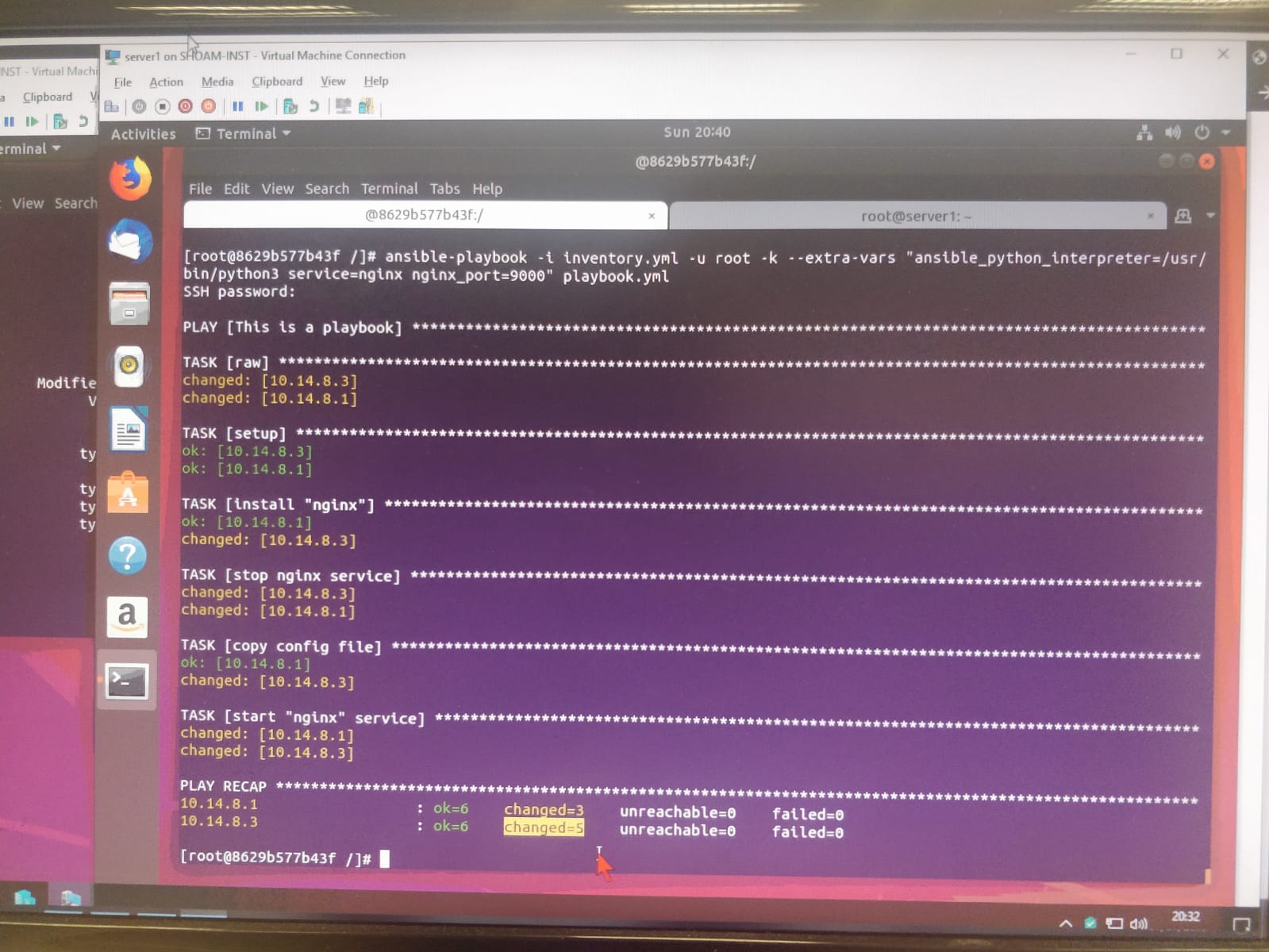
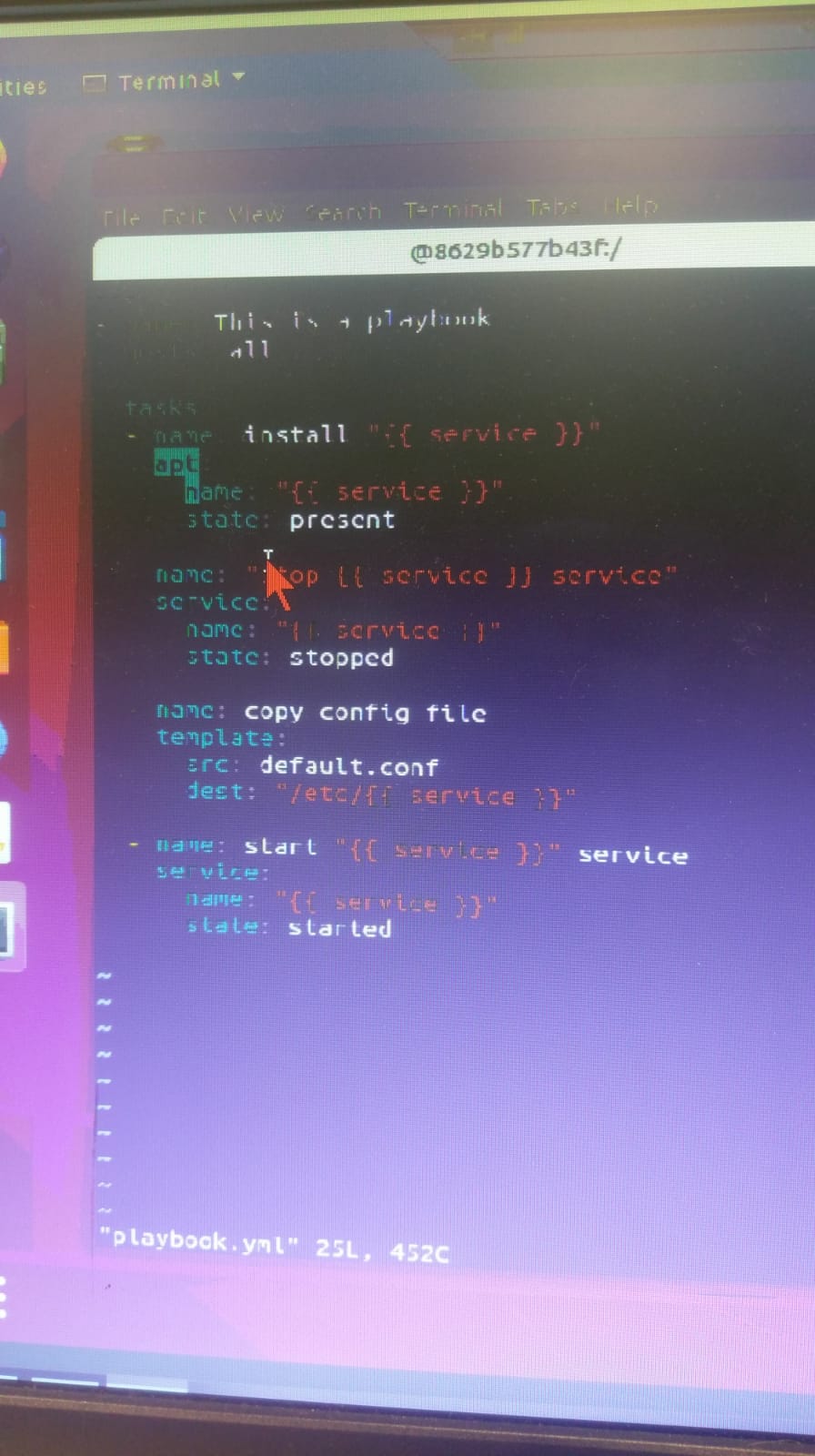
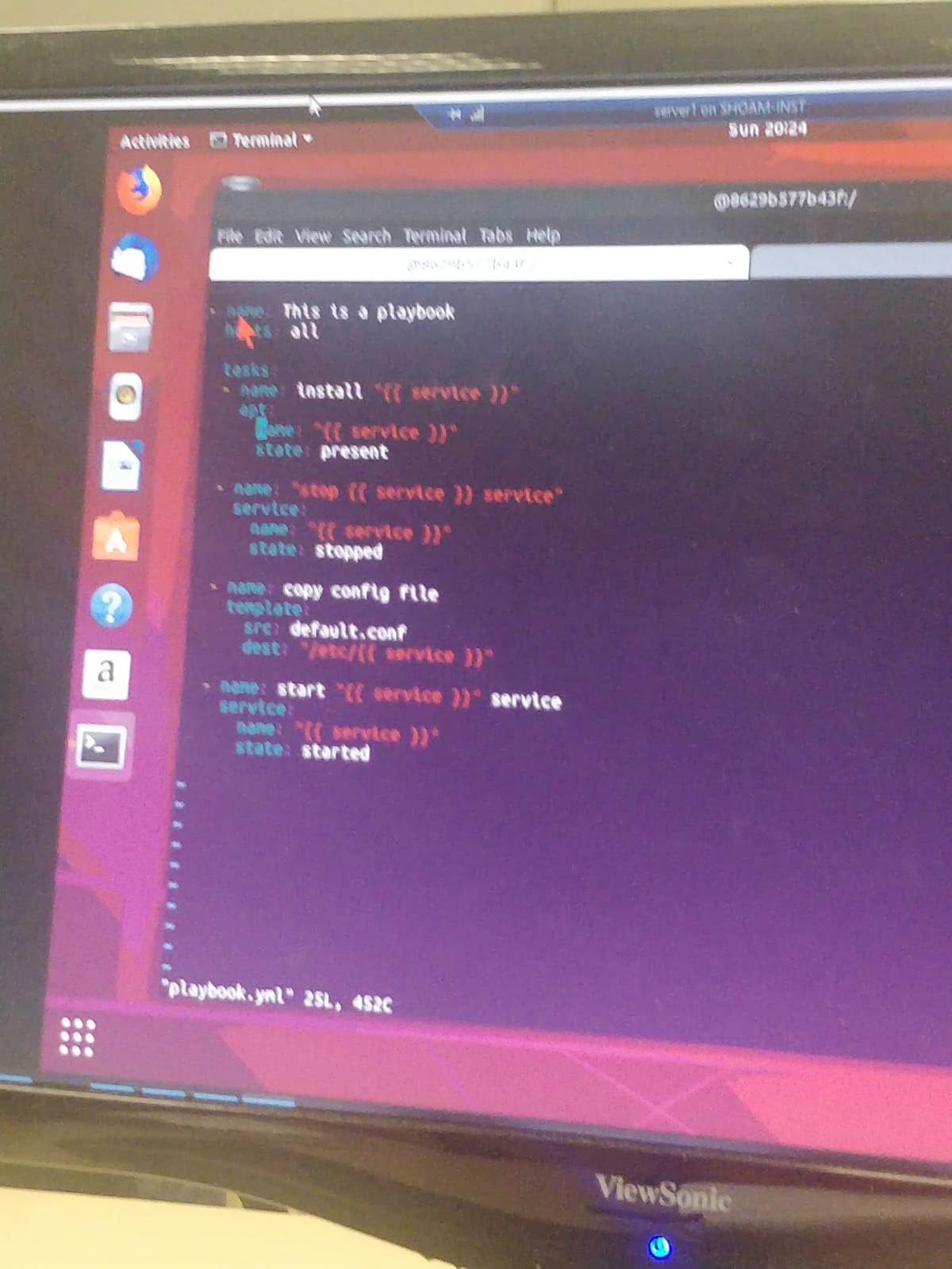


ansible.cfg /etc/ansible קובץ הקונפיגורציה :

user/pass of the remote host



בתמונות : התקנה של פייטון וnginx והחלפת הפורט של הnginx על שני מחשבים .



Jenkinsfile :

:

|  |
| --- |
| def ls\_command |
|  | pipeline { |
|  | agent { |
|  | label 'master' |
|  | } |
|  | stages { |
|  | stage('checkout'){ |
|  | steps { |
|  | script { |
|  | git credentialsId: 'devopint', url: 'https://github.com/DevOpsINT/Course.git' |
|  | } |
|  | } |
|  | } |
|  | stage('shell command example') { |
|  | steps { |
|  | script { |
|  | ls\_command = sh script: 'pwd', returnStdout: true |
|  | print(ls\_command) |
|  | sh "echo ls\_command is ${ls\_command} > variable" |
|  | sh 'cat variable' |
|  | } |
|  | } |
|  | } |
|  | stage('run ansible playbook') { |
|  | steps { |
|  | script { |
|  | sh 'ansible-playbook -i inventory.yml -u root -k --extra-vars "ansible\_python\_interpreter=/usr/bin/python3 service=nginx nginx\_port=9000" playbook.yml' |
|  | } |
|  | } |
|  | } |
|  | } |
|  | }  שיעורי הביית : |

