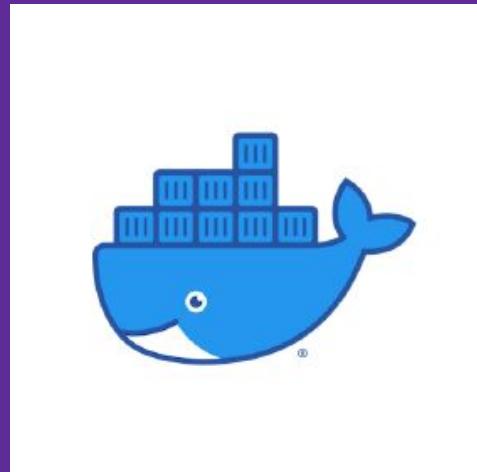
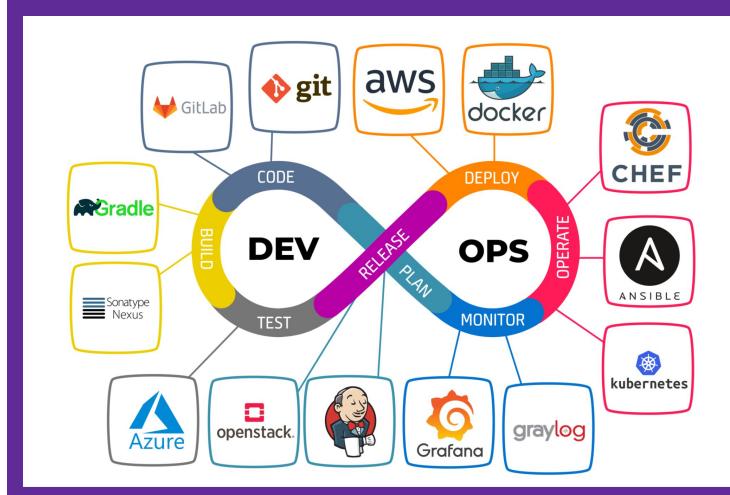


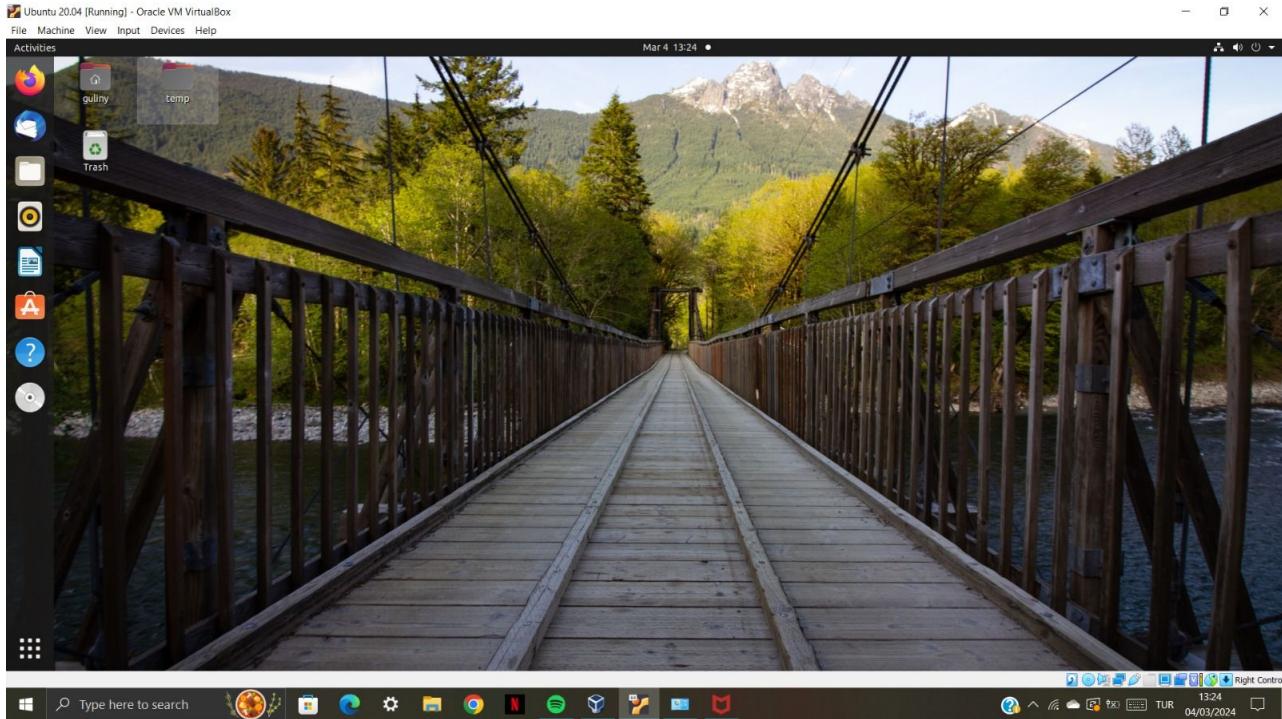
Mid Level DevOps Bootcamp

Bitirme Projesi

Gülin Yılmaz

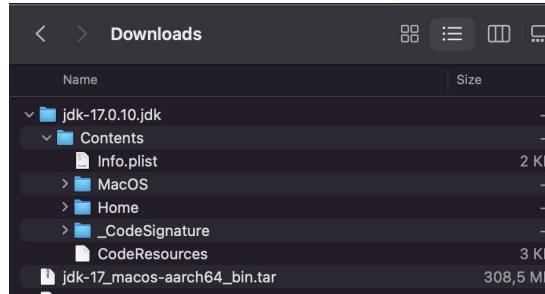
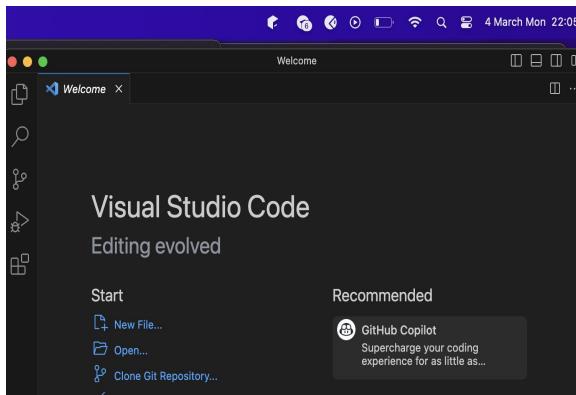


1. Adım: Sanal Makine Kurulumu

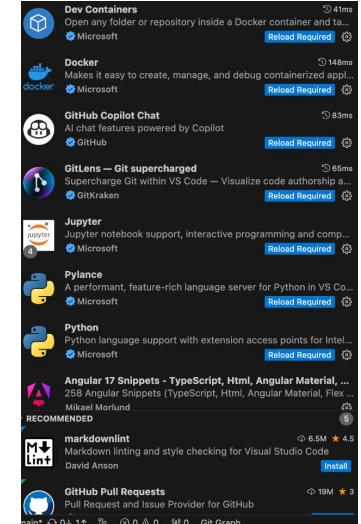
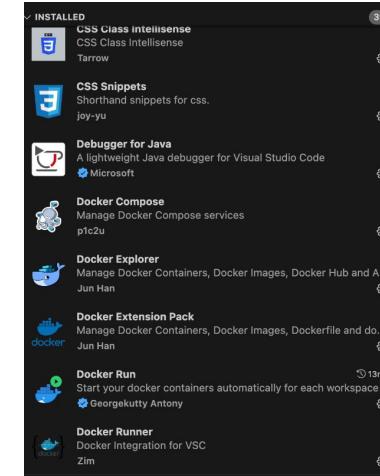


2. Adım: Kurulumlar

```
Last login: Mon Mar  4 21:13:20 on ttys001
[gulinyilmaz@Gulins-Laptop ~ % git version
git version 2.39.2 (Apple Git-143)
gulinyilmaz@Gulins-Laptop ~ % ]
```



```
[gulinyilmaz@Gulins-Laptop Desktop % mvn -v
Apache Maven 3.9.6 (bc0240f3c744dd6b6ec2920b3cd08dcc295161ae)
Maven home: /opt/homebrew/Cellar/maven/3.9.6/libexec
Java version: 21.0.2, vendor: Homebrew, runtime: /opt/homebrew/Cellar/openjdk/21.0.2/libexec/
openjdk.jdk/Contents/Home
Default locale: en_TR, platform encoding: UTF-8
OS name: "mac os x", version: "13.0", arch: "aarch64", family: "mac"
gulinyilmaz@Gulins-Laptop Desktop % ]
```



3. Adım Port Ayarları

```
gulinyilmaz@Gulins-Laptop Desktop % sudo lsof -i -P -n | grep LISTEN

Password:
ControlCe 466 gulinyilmaz 5u IPv4 0x426c0f8f57601b73 0t0 TCP *:7000 (LISTEN)
ControlCe 466 gulinyilmaz 6u IPv6 0x426c0f8f5cb5b85b 0t0 TCP *:7000 (LISTEN)
ControlCe 466 gulinyilmaz 7u IPv4 0x426c0f8f57601063 0t0 TCP *:5000 (LISTEN)
ControlCe 466 gulinyilmaz 8u IPv6 0x426c0f8f5cb5bfdb 0t0 TCP *:5000 (LISTEN)
rapportd 481 gulinyilmaz 3u IPv4 0x426c0f8f575e7193 0t0 TCP *:56795 (LISTEN)
)
rapportd 481 gulinyilmaz 4u IPv6 0x426c0f8f5cb5e55b 0t0 TCP *:56795 (LISTEN)
)
jetbrains 559 gulinyilmaz 59u IPv6 0x426c0f8f5cb5c75b 0t0 TCP 127.0.0.1:52829
(LISTEN)
Postman 571 gulinyilmaz 44u IPv4 0x426c0f8f57602683 0t0 TCP 127.0.0.1:10533
(LISTEN)
Spotify 574 gulinyilmaz 79u IPv4 0x426c0f8f56a2b193 0t0 TCP *:52783 (LISTEN)
)
Spotify 574 gulinyilmaz 94u IPv4 0x426c0f8f57603ca3 0t0 TCP *:57621 (LISTEN)
)
com.docker 667 gulinyilmaz 2985u IPv6 0x426c0f8f5cb6125b 0t0 TCP *:9000 (LISTEN)
GitHub 1642 gulinyilmaz 28u IPv4 0x426c0f8f575fddd3 0t0 TCP 127.0.0.1:49434
(LISTEN)
cupsd 16980 root 5u IPv6 0x426c0f8f5cb619db 0t0 TCP [:1]:631 (LISTEN)
EN)
cupsd 16980 root 6u IPv4 0x426c0f8f575dddd3 0t0 TCP 127.0.0.1:631 (LISTEN)
gulinyilmaz@Gulins-Laptop Desktop %
```

Not: netstat macos'ta çalışmıyor.

4. Adım: GitHub Repository

A screenshot of a GitHub repository page for 'devops_project'. The repository has 1 commit from 'gulinyilmaz' labeled 'Initial commit' at 553ff0f · now. The README file contains the text 'Mid Level Devops Bootcamp Project'. The repository has 0 stars, 1 watching, and 0 forks. It also lists 0 packages published.

A screenshot of a Mac desktop showing GitHub Desktop and Finder. GitHub Desktop shows an updated version available for download. Finder shows the local repository directory 'devops_project' containing a single file 'README.md'.

Repository linki: https://github.com/gulinyilmaz/devops_project



5. Adım

- Git nedir?

Yazılım geliştirme süreçlerinde kullanılan bir versiyon kontrol sistemidir. Git sayesinde oluşturma olduğumuz projenin adım adım versiyonlarını takip edebilir, istediğimizde eski versiyonları kullanabiliriz veya yenilerini bu sisteme ekleyebiliriz. Ekip projelerinde de grup arkadaşlarımızın yazdığı kodlara ve aynı şekilde onların kullandığı versiyonlara erişmemiz mümkün olur.

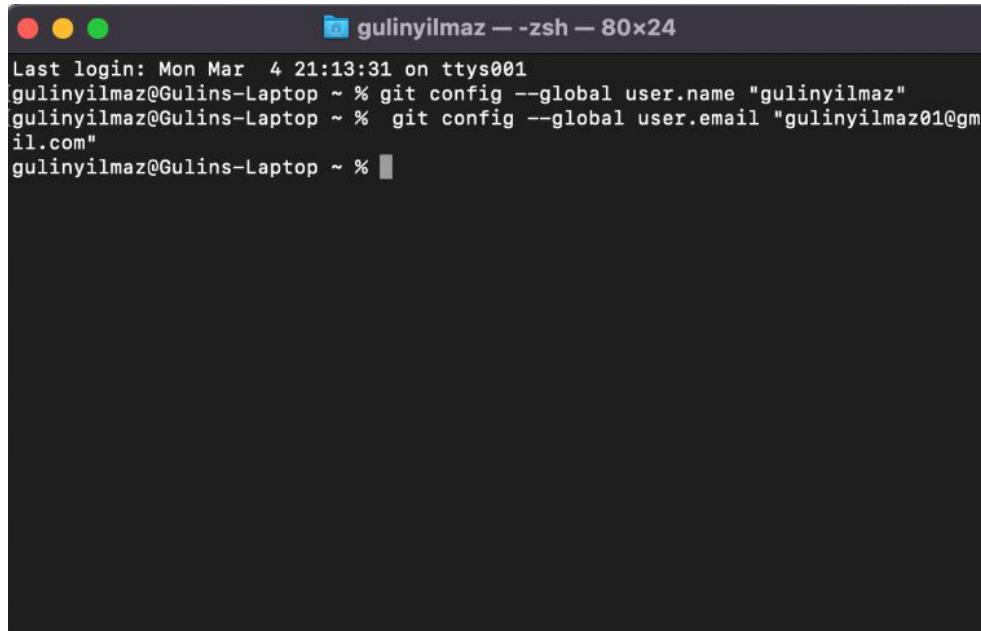
- VCS açılımı nedir?

VCS, Version Control System yani versiyon kontrol sistemidir.

- İyi bir commit özellikleri nelerdir?

Yaptığımız işi anlatan öz, açık ve anlaşılır başlık ve metinler içeren, imla ve conventionlara uyan bir commit iyi bir committir.

git ayarlarını yapalım



```
Last login: Mon Mar  4 21:13:31 on ttys001
gulinyilmaz@Gulins-Laptop ~ % git config --global user.name "gulinyilmaz"
gulinyilmaz@Gulins-Laptop ~ % git config --global user.email "gulinyilmaz01@gmail.com"
gulinyilmaz@Gulins-Laptop ~ %
```

6. Adım

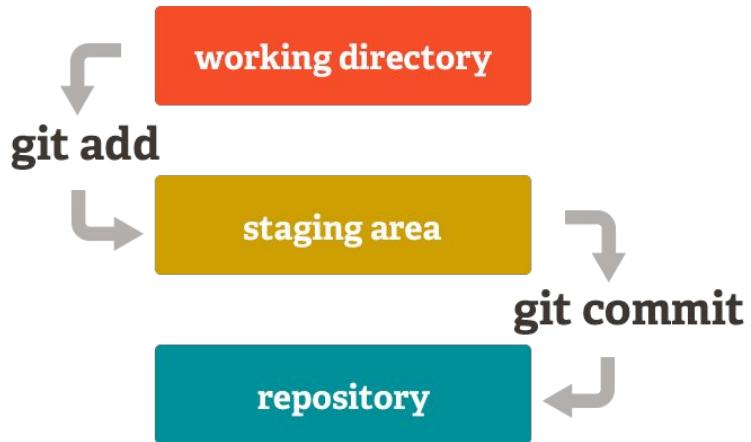
The screenshot shows a dark-themed instance of the Visual Studio Code (VS Code) code editor. In the top right corner, there is a tab labeled "jenkins.txt — devops_project". The left sidebar, titled "EXPLORER", shows a project structure with a folder named "DEVOPS_PROJECT" containing a "devops" folder. Inside "devops", there is a file named "jenkins.txt" which has been modified ("U"). Other files visible in the sidebar include "jenkins.txt" (marked as modified), "README.md", and ".gitignore". The main workspace area is currently empty. At the bottom, the "TERMINAL" tab is active, displaying a terminal session with the following history:

```
gulinyilmaz@Gulins-Laptop devops % cd jenkins.txt
cd: no such file or directory: jenkins.txt
gulinyilmaz@Gulins-Laptop devops % cd devops
cd: not a directory: jenkins.txt
gulinyilmaz@Gulins-Laptop devops % ls
jenkins.txt
gulinyilmaz@Gulins-Laptop devops % cat >> jenkins.txt
DevOps Ogreniyorum
gulinyilmaz@Gulins-Laptop devops %
```

7. Adım

- Git staged area ve unstaged area nedir?

Unstaged area, projemizdeki değişiklikleri içerir. Bu değişiklikler git tarafından tekil ediliyor veya edilmiyor olabilir fakat kaydedilmemişlerdir. Staged area ise bir sonraki commit için hazır hale getirilen dosyaları içerir ve değişiklikler git add komutuyla bu alana eklenir.



8. Adım

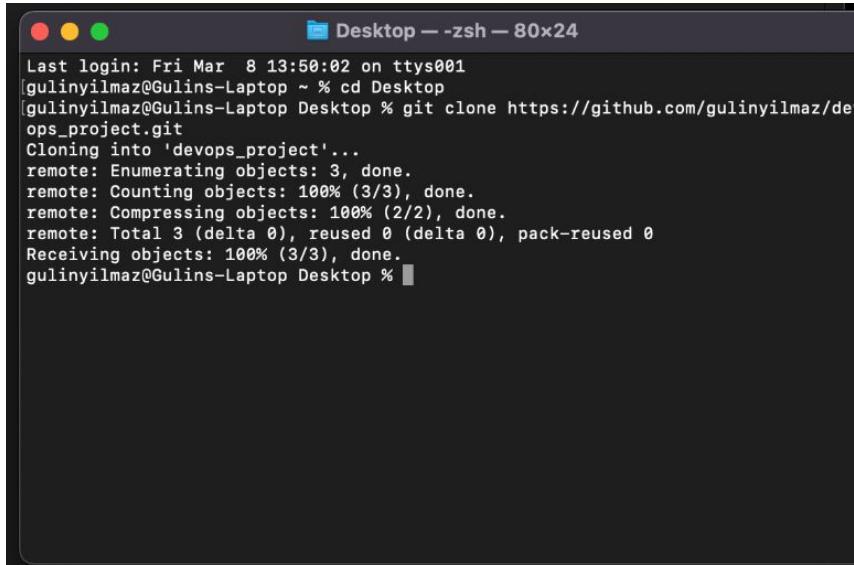
The screenshot shows a terminal session within the VS Code interface. The terminal window is titled "jenkins.txt — devops_project". The content of the terminal is as follows:

```
gulinyilmaz@Gulins-Laptop devops_project % git add .
gulinyilmaz@Gulins-Laptop devops_project % git commit -m "first commit"
[main 4e4f5b9] first commit
 1 file changed, 1 insertion(+)
  create mode 100644 devops/jenkins.txt
gulinyilmaz@Gulins-Laptop devops_project % git branch -M "main"
gulinyilmaz@Gulins-Laptop devops_project % git remot add origin https://github.com/gulinyilmaz/devops_project.git
git: 'remot' is not a git command. See 'git --help'.

The most similar command is
remote
gulinyilmaz@Gulins-Laptop devops_project % git remote add origin https://github.com/gulinyilmaz/devops_project.git
error: remote origin already exists.
gulinyilmaz@Gulins-Laptop devops_project % git remote -v
origin https://github.com/gulinyilmaz/devops_project.git (fetch)
origin https://github.com/gulinyilmaz/devops_project.git (push)
gulinyilmaz@Gulins-Laptop devops_project % git push -u origin master
error: src refspec master does not match any
error: failed to push some refs to 'https://github.com/gulinyilmaz/devops_project.git'
gulinyilmaz@Gulins-Laptop devops_project % git push -u origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 342 bytes | 342.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/gulinyilmaz/devops_project.git
 553ff0f..4e4f5b9  main --> main
branch 'main' set up to track 'origin/main'.
gulinyilmaz@Gulins-Laptop devops_project %
```

The terminal interface includes standard VS Code navigation and status bars at the bottom.

9. Adım: git clone

A screenshot of a terminal window titled "Desktop -- -zsh -- 80x24". The window shows the user's command-line session. The user has logged in on Friday, March 8 at 13:50:02. They navigated to their Desktop directory and ran a "git clone" command to download a repository from GitHub. The repository is named "devops_project" and its URL is "https://github.com/gulinyilmaz/devops_project.git". The terminal displays the progress of the cloning process, including object enumeration, counting, compressing, and receiving objects. Once completed, the user exits the terminal.

```
Last login: Fri Mar  8 13:50:02 on ttys001
gulinyilmaz@Gulins-Laptop ~ % cd Desktop
gulinyilmaz@Gulins-Laptop Desktop % git clone https://github.com/gulinyilmaz/devops_project.git
Cloning into 'devops_project'...
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
Receiving objects: 100% (3/3), done.
gulinyilmaz@Gulins-Laptop Desktop %
```

10. Adım: —amend

The screenshot shows a GitHub repository page for 'devops_project'. The repository is public and owned by 'gulinyilmaz'. The main branch is 'main'. There are four commits listed:

- Merge branch 'main' of https://github.com/gulinyilmaz/devops_project into main · a5713a5 · now · 4 Commits
- devops · edited commit · 7 minutes ago
- README.md · Initial commit · 4 days ago
- README ·

The 'README' file is currently selected. The repository description is "Mid Level Devops Bootcamp Project".

```
Please supply the message using either -m or -r option.
● gulinyilmaz@Gulins-Laptop devops_project % git commit --amend -m "edited commit"
[main fcfbd73] edited commit
Date: Fri Mar 8 14:05:39 2024 +0300
1 file changed, 1 insertion(+)
create mode 100644 devops/jenkins.txt
○ gulinyilmaz@Gulins-Laptop devops_project %
```

11. Adım: git log, git status

- Log komutu, Git reposundaki commit geçmişini gösterir. Yapılan commitleri, bunların özet bilgilerini (hash, yazar, tarih, ve commit mesajı gibi) ve commitler arasındaki ilişkileri görüntüleyebiliriz.
- Status komutu ise durum gösterir, yani projedeki değişiklikleri ve bu değişikliklerin durumunu gösterir. Hangi dosyaların değiştirildiği, hangilerinin stagelenmiş olduğu ve hangilerinin henüz stagelenmediği gibi bilgilere bu komut sayesinde ulaşabiliriz.

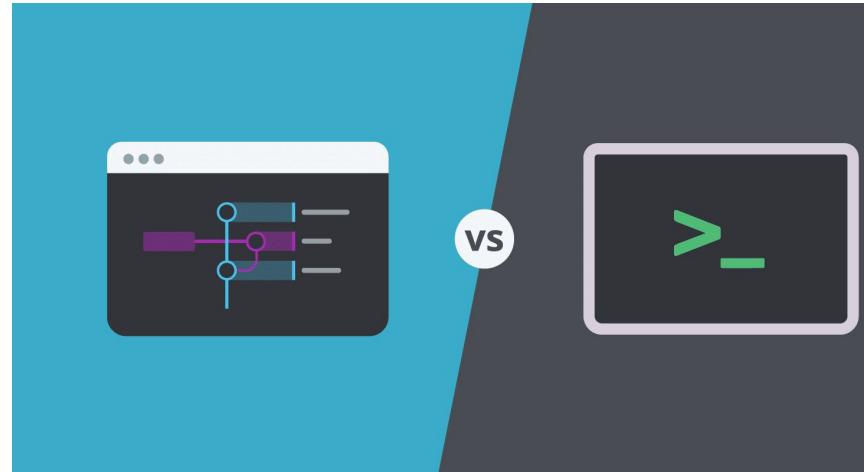


12. Adım: backend branch, fast forward merge

```
● gulinyilmaz@Gulins-Laptop devops_project % git checkout -b backend
Switched to a new branch 'backend'
● gulinyilmaz@Gulins-Laptop devops_project % mkdir backend_directory
● gulinyilmaz@Gulins-Laptop devops_project % touch backend_directory/newfile.txt
● gulinyilmaz@Gulins-Laptop devops_project % touch backend_directory/newerfile.txt
● gulinyilmaz@Gulins-Laptop devops_project % git add .
● gulinyilmaz@Gulins-Laptop devops_project % git commit -m "backend directory added"
[backend d34c866] backend directory added
 2 files changed, 0 insertions(+), 0 deletions(-)
   create mode 100644 backend_directory/newerfile.txt
   create mode 100644 backend_directory/newfile.txt
● gulinyilmaz@Gulins-Laptop devops_project % git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
● gulinyilmaz@Gulins-Laptop devops_project % git merge backend
Updating a5713a5..d34c866
Fast-forward
  backend_directory/newerfile.txt | 0
  backend_directory/newfile.txt  | 0
  2 files changed, 0 insertions(+), 0 deletions(-)
   create mode 100644 backend_directory/newerfile.txt
   create mode 100644 backend_directory/newfile.txt
○ gulinyilmaz@Gulins-Laptop devops_project % █
```

13. Adım: git gui, git cli

- Git GUI yani graphical user interface, kullanıcıya grafiksel bir interface sağlar ve git işlemlerini bu interface üzerinden yapmasına olanak sağlar. Git işlemlerini daha kullanıcı dostu yapmak için tasarlanmıştır.
- Git CLI yani command line interface ile ise command line üzerinden git komutlarını yazarak işlemlerimizi yapabiliriz. Bu yöntem kullanıcılara esneklik sağlar ve yazılım süreçlerini otomatikle etme konusunda güçlündür.



14. Adım: frontend branch, no-fast-forward

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL zsh + v ⌂

● gulinyilmaz@Gulins-Laptop devops_project % mkdir frontend_directory
● gulinyilmaz@Gulins-Laptop devops_project % touch frontend_directory/frontend.txt
● gulinyilmaz@Gulins-Laptop devops_project % touch frontend_directory/frontend_new.txt
● gulinyilmaz@Gulins-Laptop devops_project % git commit -m "frontend"
On branch frontend
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    frontend_directory/
nothing added to commit but untracked files present (use "git add" to track)
● gulinyilmaz@Gulins-Laptop devops_project % git add .
● gulinyilmaz@Gulins-Laptop devops_project % git commit -m frontend
[frontend 985f530] frontend
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 frontend_directory/frontend.txt
 create mode 100644 frontend_directory/frontend_new.txt
● gulinyilmaz@Gulins-Laptop devops_project %
```

```
EXPLORER ... jenkins.txt newerfile.txt
DEVS_PROJECT ...
  backend_directory
    newerfile.txt
    newfile.txt
  devops
  jenkins.txt
  frontend_directory
    frontend_new.txt
    frontend.txt
  README.md

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL zsh + v ⌂

gulinyilmaz@Gulins-Laptop devops_project % git commit -m "frontend"
On branch frontend
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    frontend_directory/
nothing added to commit but untracked files present (use "git add" to track)
gulinyilmaz@Gulins-Laptop devops_project % git add .
gulinyilmaz@Gulins-Laptop devops_project % git commit -m frontend
[frontend 985f530] frontend
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 frontend_directory/frontend.txt
 create mode 100644 frontend_directory/frontend_new.txt
gulinyilmaz@Gulins-Laptop devops_project % git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)
● gulinyilmaz@Gulins-Laptop devops_project % git merge frontend --no-ff
Merge made by the 'ort' strategy.
  frontend_directory/frontend.txt | 0
  frontend_directory/frontend_new.txt | 0
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 frontend_directory/frontend.txt
 create mode 100644 frontend_directory/frontend_new.txt
● gulinyilmaz@Gulins-Laptop devops_project %
```

gulinylimaz / devops_project

<> Code Issues Pull requests Actions Projects Wiki ...

Mid Level Devops Bootcamp Project

0 stars 0 forks 1 watching 1 Branch 0 Tags Activity

Private repository

main Go to file + <> Code

gulinylimaz Merge branch 'frontend' without ff 17 minutes ago

backend_directory merge oncesinde commit 30 minutes ago

devops edited commit 1 hour ago

frontend_directory frontend 18 minutes ago

README.md Initial commit 4 days ago

README

devops_project

Mid Level Devops Bootcamp Project

newerfile.txt newerfile.txt

backend_directory > newerfile.txt

1 backend dir file

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL zsh +

gulinylimaz@Gulins-Laptop devops_project % git commit -m "frontend"
On branch frontend
Untracked files:
(use "git add <file>..." to include in what will be committed)
[frontend_directory/](#)

nothing added to commit but untracked files present (use "git add" to track)
gulinylimaz@Gulins-Laptop devops_project % git add .
gulinylimaz@Gulins-Laptop devops_project % git commit -m frontend
[frontend 985f530] frontend
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 frontend_directory/frontend.txt
create mode 100644 frontend_directory/frontend_new.txt
gulinylimaz@Gulins-Laptop devops_project % git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
gulinylimaz@Gulins-Laptop devops_project % git merge frontend --no-ff
Merge made by the 'ort' strategy.
frontend_directory/frontend.txt | 0
frontend_directory/frontend_new.txt | 0
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 frontend_directory/frontend.txt
create mode 100644 frontend_directory/frontend_new.txt
gulinylimaz@Gulins-Laptop devops_project %

+ Create new branch...
+ Create new branch from...
+ Checkout detached...
+ main 96451270
+ frontend 985f530b
+ backend d34c8663
+ origin/HEAD Remote branch at 96451270
+ origin/main Remote branch at 96451270

15. Adım: yeni repository, SSH key

Gülin Yılmaz (gulinyilmaz)
Your personal account

- Public profile
- Account
- Appearance
- Accessibility
- Notifications

Access

Billing and plans

Emails

SSH keys

This is a list of SSH keys associated with your account. Remove any keys that you no longer need.

Authentication keys

Name	Type	Added	Last used	Permissions
macbook_key	SSH	Added on Mar 9, 2024	Last used within the last week	Read/write

```
|gulinyilmaz@Gulins-Laptop Desktop % git clone git@github.com:gulinyilmaz/temp_repository.git
Cloning into 'temp_repository'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
|gulinyilmaz@Gulins-Laptop Desktop % cd temp_repository
|gulinyilmaz@Gulins-Laptop temp_repository % mkdir devops
|gulinyilmaz@Gulins-Laptop temp_repository % cd devops
gulinyilmaz@Gulins-Laptop devops % cat >> jenkins.txt
DevOps ogreniyorum:wq
^C
gulinyilmaz@Gulins-Laptop devops %
```

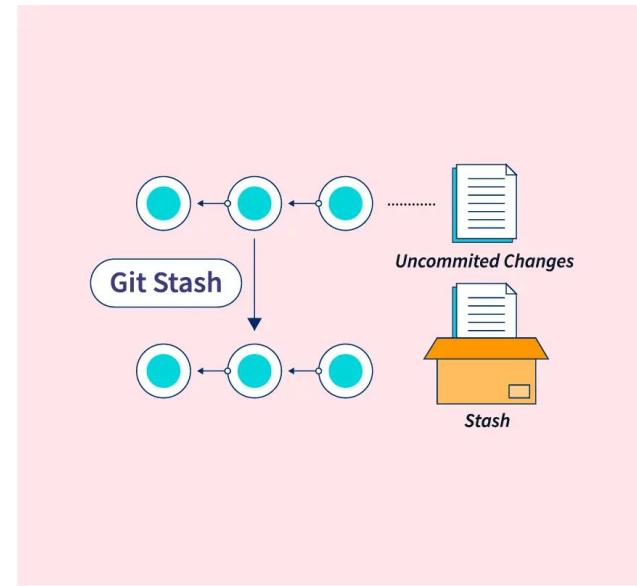
Repository link:
https://github.com/gulinyilmaz/temp_repository

Secure Shell Protocol



16. Adım: git stash

- Git stash çalışma dizinimizdeki geçici değişiklikleri saklamak ve daha sonra geri almak için kullanılan bir komuttur. Genellikle aniden başka bir iş üzerinde çalışmaya başlanınca veya branch değiştirince kullanılır. Başka bir branch geçip ordaki değişiklikleri kaydedip tekrar stashten önceki değişikliklerimizi alarak devam edebiliriz.

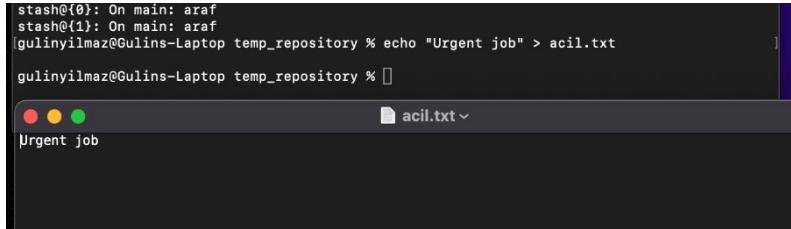


17. Adım: stash

```
gulinyilmaz@Gulins-Laptop temp_repository % git add .
gulinyilmaz@Gulins-Laptop temp_repository % git stash save "araf"
Saved working directory and index state On main: araf
gulinyilmaz@Gulins-Laptop temp_repository % git stash list
stash@{0}: On main: araf
stash@{1}: On main: araf
gulinyilmaz@Gulins-Laptop temp_repository %
```

1. Değişiklikler stash'e eklendi.
2. Yeni iş txt dosyası olarak eklendi.

```
stash@{0}: On main: araf
stash@{1}: On main: araf
[gulinyilmaz@Gulins-Laptop temp_repository % echo "Urgent job" > acil.txt
]
gulinyilmaz@Gulins-Laptop temp_repository %
```



The screenshot shows a terminal window with the following output:
stash@{0}: On main: araf
stash@{1}: On main: araf
[gulinyilmaz@Gulins-Laptop temp_repository % echo "Urgent job" > acil.txt
]
gulinyilmaz@Gulins-Laptop temp_repository %

Below the terminal, a file browser window is open. It shows a single file named 'acil.txt' with a preview of the text 'Urgent job'. The file has a small red dot icon next to it, indicating it's a modified file.

```
[gulinyilmaz@Gulins-Laptop temp_repository % git commit -m "acil done"
[main 32dbc29] acil done
 1 file changed, 1 insertion(+)
 create mode 100644 acil.txt
[gulinyilmaz@Gulins-Laptop temp_repository % git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 300 bytes | 300.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:gulinyilmaz/temp_repository.git
 0@224c3..32dbc29  main -> main
branch 'main' set up to track 'origin/main'.
gulinyilmaz@Gulins-Laptop temp_repository % ]
```

3. Acil iş push'landı.
4. Stash apply kullanıldı
stash pushlandı.

```
branch 'main' set up to track 'origin/main'.
gulinyilmaz@Gulins-Laptop temp_repository % git stash list
stash@{0}: On main: araf
stash@{1}: On main: araf
gulinyilmaz@Gulins-Laptop temp_repository % git stash apply araf
error: araf is not a valid reference
gulinyilmaz@Gulins-Laptop temp_repository % git stash apply stash@{1}

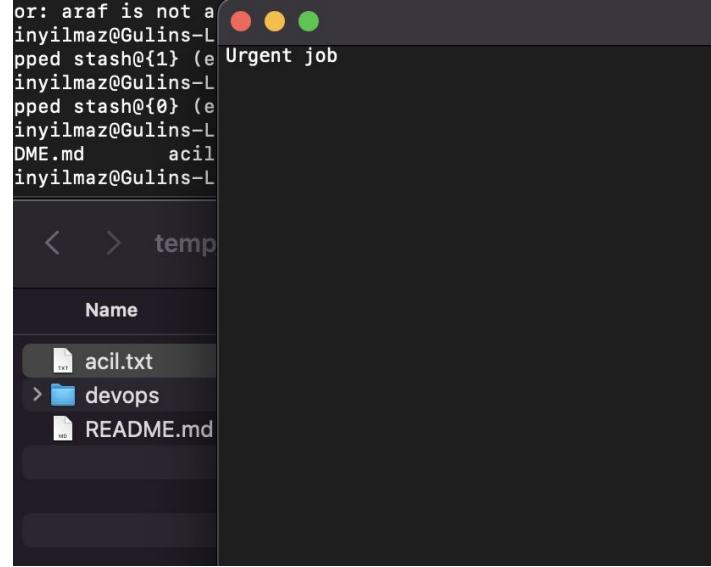
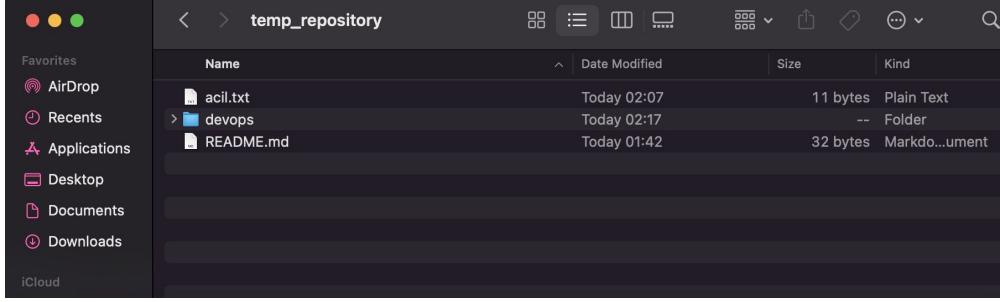
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:  devops/.DS_Store
    new file:  devops/jenkins.txt

gulinyilmaz@Gulins-Laptop temp_repository % git add .
gulinyilmaz@Gulins-Laptop temp_repository % git commit -m "stash sonrası commit"
[main ae2329b] stash sonrası commit
 2 files changed, 1 insertion(+)
 create mode 100644 devops/.DS_Store
 create mode 100644 devops/jenkins.txt
gulinyilmaz@Gulins-Laptop temp_repository % git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 675 bytes | 675.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:gulinyilmaz/temp_repository.git
  32dbc29..ae2329b  main -> main
branch 'main' set up to track 'origin/main'.
gulinyilmaz@Gulins-Laptop temp_repository % ]
```

```
[ungan main] Set up to track origin/main  
gulinylimaz@Gulins-Laptop temp_repository % git stash drop araf  
error: araf is not a valid reference  
gulinylimaz@Gulins-Laptop temp_repository % git stash drop stash@{1}  
Dropped stash@{1} (e6aa492034f03dfc023e6afa61c2ba2bec48fb49)  
gulinylimaz@Gulins-Laptop temp_repository % git stash drop stash@{0}  
Dropped stash@{0} (e99d9fcfa9f72c3fe6cd57f2c4a09d340155c37ce8)  
gulinylimaz@Gulins-Laptop temp_repository %
```

5. Stash droplandi.



Repository'nin son hali.

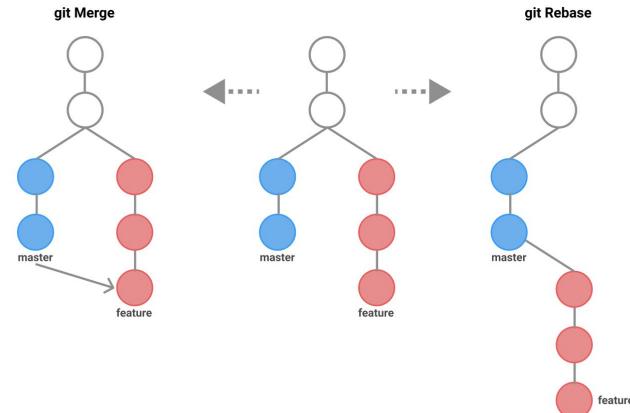
18. Adım: alias

```
 README.md      ascii.txt      devops
gulinyilmaz@Gulins-Laptop temp_repository % git config --global alias.graph "log --all --one
                                         ate --graph"

[gulinyilmaz@Gulins-Laptop temp_repository % git graph
* ae2329b (HEAD -> main, origin/main, origin/HEAD) stash sonrası commit
* 32dbc29 acil done
* 0a224c3 Initial commit
gulinyilmaz@Gulins-Laptop temp_repository % ]
```

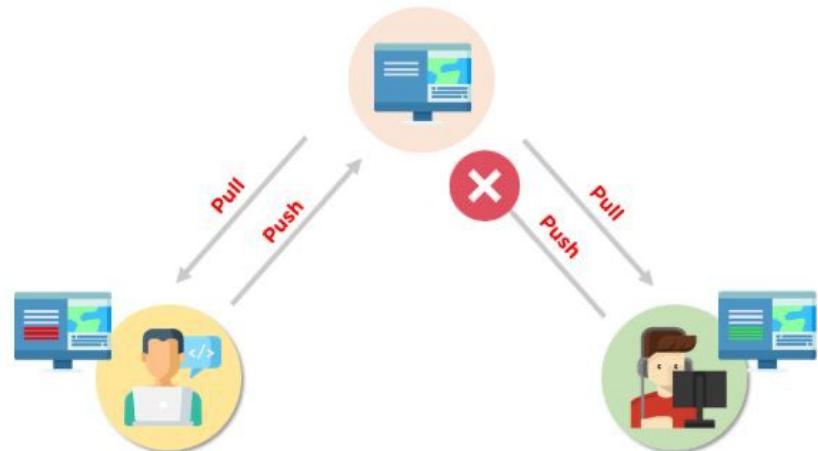
19. Adım: Rebase & Merge

- Merge işlemi iki branchteki değişiklikleri bir araya getirir ve birleştirilen commits oluşturur. İki branchin de commit geçmişi korunur ve birleşim işlemini temsil eden yeni bir merge commit oluşur. Merge sonucunda geçmişte yapılan committer korunur ve bunların tarihçesi rahat bir şekilde görülebilir.
- Rebase işlemi ise birleştirilecek branchteki değişikliklei alır ve yeni commitleri hedef branchin en son commitine ekler. Linear bir proje geçmişi kullanılmak istediginde rebase işlemi tercih edilebilir. Fakat branch işlemi geçmişi değiştirdiği için bir ekiple yapılan projelerde shared branchlerde dikkatli kullanılmalıdır.



20. Adım: git conflict

- Bir conflict yediğimizde ne yapmamız gerekiyor?
Öncelikle en son değişiklikleri mergelememiz gerekiyor. Sonrasında conflict çıkarsa bunları görmek için git status komutunu kullanabiliriz. Bir text editör yardımıyla conflictleri görüntüleyebiliriz. Bu editörde conflict eden kısımlar görünecektir, kullanmak istediğimiz kısımları seçmeliyiz ve conflictleri bu şekilde resolve edebiliriz. Değişiklikleri kaydedip editörü kapattıktan sonra conflict çözülmüş demektir. Commit ve push yapabiliriz.



21. Adım: git ignore

- Gitignore dosyası projemizin rootunda oluşturulan düz metin dosyasıdır. Göz ardı etmek istediğimiz, yani git tarafından takip edilmesini istemediğimiz dizinler veya dosyalar için kullanılır. Projenin gerçek source kodu olmayan yani ide eklentileri, hassas bilgiler, log dosyaları, yararsız sistem dosyaları gibi dosyaların git tarafından göz ardı edilmesini sağlar.



22. Adım: git tag

- Git tag komutu git reposunda belirli bir commit'e bir tag eklemek için kullanılır. Etiketler, genellikle bir sürüm numarasını veya bir önemli milestonu temsil eder ve projenin belirli bir durumunu işaretler.
- Örnek olarak, **git tag v1.1** komutu, mevcut commit'in üzerine "v1.1" adında bir tag ekler. Bu, projenin v1.1 sürümüne ulaşıldığı bir noktayı işaretler.



23. Adım: git diff

- Git diff komutu iki commit arasındaki veya iki dizin arasındaki farkları göstermek için kullanılır. git diff 3b2f0ab 5a2b8de ise, 3b2f0ab ve 5a2b8de commitleri arasındaki değişiklikleri gösterir. Yeşil renkteki satırlar eklenen satırları temsil ederken, kırmızı renkteki satırlar silinen satırları temsil eder.

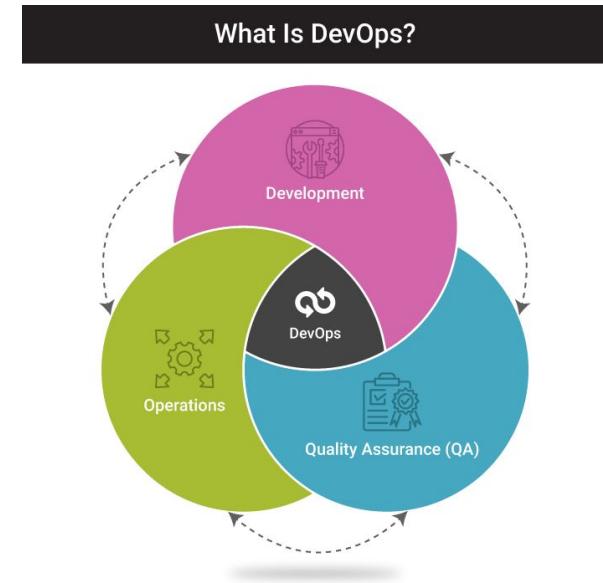
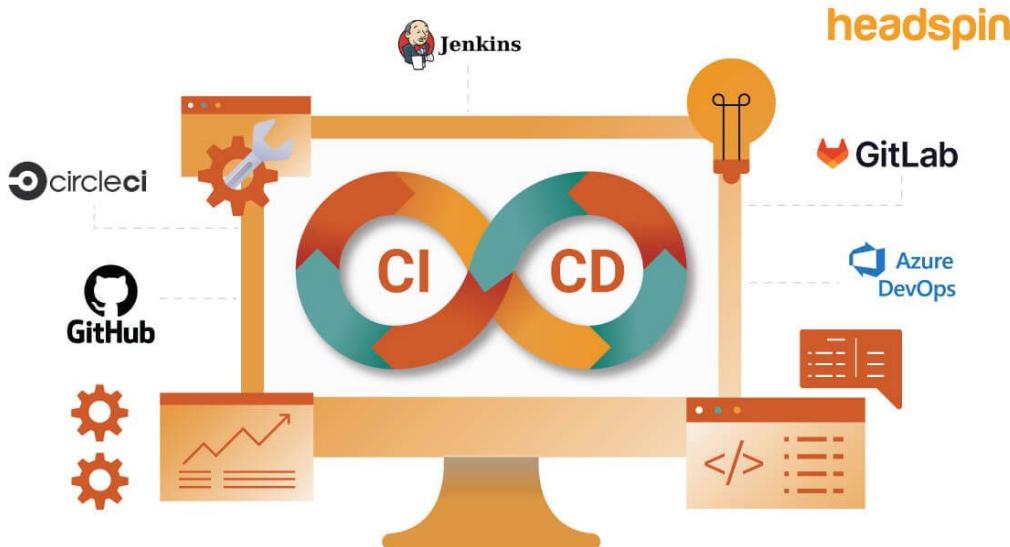
24. Adım: DevOps Kültür Felsefesi

DevOps kültürü bir organizasyonun developer (geliştirme) ve operations (işletme) ekipleri arasındaki işbirliği ve etkileşime dayanan bir çalışma yaklaşımını ifade eder. Verimliliğin arttırılması için sürekli bir öğrenme ve adaptasyon sürecine dayanır. DevOps kültürün prensiplerinden bazıları şunlardır:

- **İşbirliği ve iletişim:** Ekipler arasında açık ve sürekli iletişim olmalıdır.
- **Otomasyon:** Continuous integration, continuous delivery (CI/CD) gibi süreçler benimsenir. Hızlı ve güvenilir yazılımlar kullanmak ve üretmek amaçlanır.
- **Sürekli iyileştirme:** Ekipler geribildirimlere önem verir süreçler düzenli değerlendirip iyileştirilir.
- **Güvenlik ve Güvenilirlik:** Güvenlik yazılım süreçlerine entegre edilir, ön planda tutulur.
- **İzleme ve geribildirim:** Uygulama ve altyapısı izlenir, analizler yapılır.

25. Adım

- DevOps kelimesi "Development" (Geliştirme) ve "Operations" (İşletme) kelimelerinin birleştirilmesiyle ortaya çıkmıştır. Dolayısıyla bu ekipler arasındaki işbirliği ve iletişime vurgu yapar.



26. Adım: CI/CD

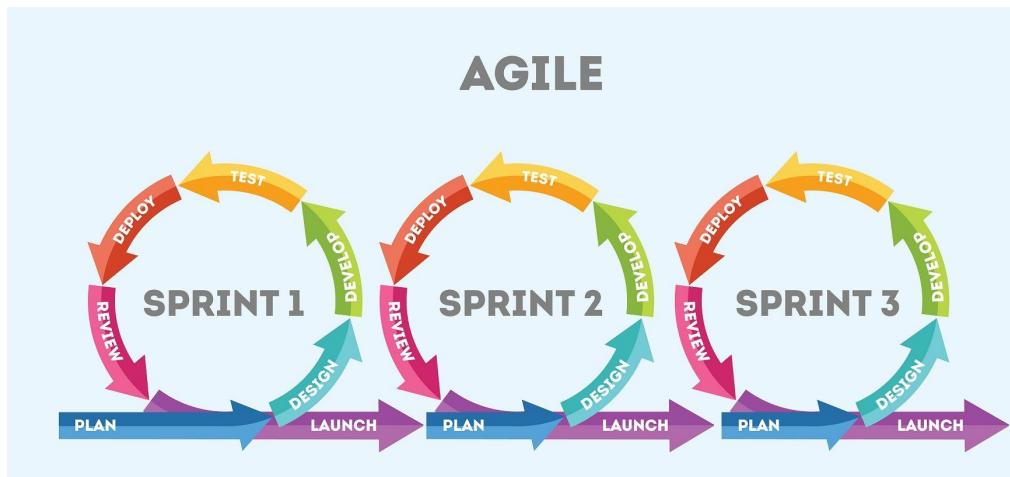
- Continuous Integration: Sürekli entegrasyon source codelarının geliştirildikten sonra entegre olmadan sürekli bir kontrol mekanizmasının olmasıdır. Her yazılımcı kendi kodunu geliştirdikten sonra onu mergelemelidir fakat bu süreçte hata olmaması için kontroller yapılması gereklidir. Bu sorunu çözmek için otomatik testler kullanılır. Testler bir CI sunucunda yapılır, kodun merge edilebilmesi için testler başarılı olmalıdır. Bu testler yapıldığı zaman aynı zamanda sunucular bizim için analiz de yapmaktadır, bu sayede source code'unun performansı ölçülür ve kalite kontrolü yapılır. Bu süreç sürekli olduğunda ve otomatikleştirildiğinde CI sağlanmış olur.
- Continuous Delivery ve Deployment: Sürekli doğrulama yazılımın her an kullanılabilir ve dağıtıma hazır olmasını amaçlar. Sürekli dağıtım ise Sürekli doğrulamanın bir adım ötesidir. Yazılımların testleri tamamlanıp onaylandıktan sonra otomatik olarak üretim ortamına taşınmasını amaçlar. Bu sayede kullanıcırlara sürekli bir dağıtım sunulur.

27. Adım: git

- Git bir versiyon kontrol sistemidir ve DevOps süreçleri için önemli bir rolü vardır. Bir projedeki değişiklikleri ve versiyonları takip etmek, ve ekiplerin iş birliği yapması için kullanılır. Geliştiriciler projenin farklı özelliklerini implement etmek için branch'ler oluşturur ve git sayesinde kendi branch'lerinde çalışır. Ardından yine git kullanarak geliştirdikleri yazılım parçalarını merkezi repository'ye gönderebilir. Git sayesinde kod tabanındaki değişiklikler izlenebilir ve gerektiğinde tekrar kullanabilir. Git ayrıca sürekli entegrasyon ve teslimat süreçlerinde de önemli rol oynar. Geliştiriciler kodlarını merkezi depoya göndererek sürekli entegrasyonu destekler.

28. Adım: Agile

- Agile yazılım geliştirme süreçlerini yöneten yazılım geliştirme yöntemlerinden biridir. Müşteri memnuniyetine odaklanır. Değişken projelerde sık sık ürünün kullanılabilir versiyonlarını müşteriye sunarak alınan geribildirimle müşteriyle birlikte süreci ilerletmeyi amaçlar. Değişen gereksinimlere adaptasyon, iletişim, işlevsel yazılım teslimatı, sürdürülebilir gelişme, teknik mükemmeliyet, sürekli geri bildirim önemli öğeleridir.



29. Adım: Monitoring

- Continuous monitoring bir uygulamanın performansını, güvenilirliğini ve durumunu sürekli olarak izlemek demektir. Bu izleme metrikler ve loglar kullanılarak yapılabilir. Loglar sistemin durumu ve resourceları kullanımı hakkında bize bilgi verir. Aynı zamanda uyarılar ve alarm sistemleri eklenebilir. Uygulamamızı sürekli olarak izleyerek hataları hızlıca tespit edebilir ve müdahale edebiliriz böylece sistemin performansı istediğimiz gibi seyreder.



30. Adım

XML:

- Extensible markup language. Verilerin depolanması ve taşınması için kullanılan işaretleme dilidir. Hem insanlardan tarafından okunabilir hem de makineler tarafından eşlenebilir.

JSON:

- Javascript object notation. Veri değişim formatıdır. Programlar arası veri aktarımı için en sık kullanılan formatlardandır. İnsanlar tarafından okunabilir ve yazılabilir. Genellikle API'lar ve web uygulamalarında kullanılır.

Yaml:

- YAML Ain't Markup Language. Veri serileştirme formatıdır. İnsanlar tarafından yazılıp okunabilir. Configuration, veri transferi ve yapılandırma dosyalarında kullanılır.

HTTP:

- HyperText Transfer Protocol. Web tarayıcıları ile sunucular arasında bilgi alışverişini yöneten bir protokoldür. Bu protokol, web sayfalarının, resimlerin, videoların ve diğer içeriklerin tarayıcıdan sunucuya veya sunucudan tarayıcıya iletimini düzenler.

Server:

- Türkçesi sunucudur. Sunucu, webde diğer cihazlara hizmet sağlayan özel bir bilgisayar sistemidir. Bu sistemler, kullanıcı taleplerini karşılamak, dosya depolamak, veritabanı hizmetleri sunmak veya diğer ağ görevlerini yerine getirmek amacıyla tasarlanmıştır.

Java JDK:

- Java Development Kit Java dilinde program yazabilmek için kullanılan geliştirme kitidir. JDK Java kütüphaneleri, JRE ve geliştirme yapmak için gerekli olan araçları içerir. JRE, yani Java Runtime Environment Java uygulamalarını runlamak için gereken kütüphaneleri ve JVM'i içerir. JVM ise Java bytecode'nu execute etmek için gerekli olan sanal makinedir.

31. Adım: Maven

- Maven open source bir proje yönetim ve otomasyon aracıdır. Projelerin yapılandırılması, dependencylerinin yönetilmesi, test edilmesi ve dağıtılmasını kolaylaştırmak için kullanılır. XML tabanlıdır. clean install komutu ise clean ile bir projenin önceki derleme sonuçlarını yanı belleği temizler, install ile de derlenen projeyi local Maven deposuna yükler. Yani bir projeyi baştan sona compile etmek ve çalıştırmak için kullanılır.

32. Adım: Docker

- Docker, yazılımlarımızı hızlı bir şekilde derleme, test etme ve dağıtma imkanı sunan bir yazılım platformudur. Docker, yazılımları librarylerden sistem araçlarına, kod ve çalışma zamanına kadar ihtiyaç duyulan tüm unsurları içeren standartlaştırılmış konteyner birimleri olarak paketleyerek işlev görür. Docker'ı kullanarak, uygulamalarımızı kolayca farklı ortamlara dağıtabilir, ölçeklendirebilir ve kodumuzun her zaman sorunsuz bir şekilde çalışmasını sağlayabiliriz.
- Docker Daemon, sistem üzerinde çalışan arka plandaki süreçtir ve Docker konteynerlerini oluşturma, yönetme ve izleme görevlerini gerçekleştirir.
- Docker Command Line Interface), kullanıcının Docker Daemon ile etkileşime girmesini sağlayan command line arayüzüdür.



33. Adım

34. docker search nginx

35. docker pull nginx

```
gulinyilmaz@Gulins-Laptop ~ % docker search nginx
NAME                                     DESCRIPTION                                              STAR
nginx                                    Official build of Nginx.                                1967
unit                                      Official build of NGINX Unit: Universal Web ...          25
nginxinc/nginxx-unprivileged             NGINX and NGINX Plus Ingress Controllers fo...        88
nginxinc/nginxx-prometheus-exporter      Unprivileged NGINX Dockerfiles                      141
nginxinc/nginxx-s3-gateway               NGINX Prometheus Exporter for NGINX and NGIN...       38
nginx/unit                               Authenticating and caching gateway based on ...       6
nginx/nginxx-ingress-operator            This repository is retired, use the Docker o...       64
nginxinc/amplify-agent                  NGINX Ingress Operator for NGINX and NGINX P...       2
nginx/nginxx-quic-qns                  NGINX Amplify Agent docker repository                 1
nginxinc/ingress-demo                  NGINX QUIC interop                                       1
nginxproxxy/nginxx-proxy                Ingress Demo                                         4
nginxproxxy/acme-companion              Automated nginx proxy for Docker containers ...     132
bitnami/nginx                            Automated ACME SSL certificate generation fo...     130
bitnami/nginx-ingress-controller        Bitnami nginx Docker Image                           183
ubuntu/nginx                            Bitnami Docker Image for NGINX Ingress Contr...       32
nginxproxy/docker-gen                   Nginx, a high-performance reverse proxy & we...     112
nginxinc/nginmesh_proxy_debug           Generate files from docker container meta-da...       16
nginxinc/mra-fakes3                     An Nginx image based off nginx:alpine and in...       7
kasmweb/nginx                           0
nginxinc/nginmesh_proxy_init            0
rancher/nginxx-ingress-controller      12
nginxinc/mra_python_base               0
nginxinc/nginx-rust-tool               0
gulinyilmaz@Gulins-Laptop ~ % docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
f546e941f15b: Pull complete
2d258780861a: Pull complete
a7d6e9feb830: Pull complete
42e0f9421c7a: Pull complete
14a95f763a2f: Pull complete
164c21b63fde: Pull complete
5b452a5fd809: Pull complete
Digest: sha256:c26ae7472d624ba1fafd296e73cecc4f93f853088e6a9c13c0d52f6ca5865107
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

36. Adım

80 portundan 6666 portuna yönlendiren, arka planda çalışan(-d), yeni adı web_2 (--name), container kapatıldıktan sonra silinen komut (--rm)

```
What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview nginx
gulinyilmaz@Gulins-Laptop ~ % docker run -d -p 6666:80 --name web_2 --rm nginx
4b993abbf4aec3479b6e299047082f1c7908864a000e83184a8aa1309b06f588
gulinyilmaz@Gulins-Laptop ~ % docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4b993abbf4ae nginx "/docker-entrypoint..." 3 minutes ago Up 3 minutes 0.0.0.0:6666→80/tcp web_2
gulinyilmaz@Gulins-Laptop ~ % curl http://localhost:6666

<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
gulinyilmaz@Gulins-Laptop ~ %
```

37. Adım

The screenshot shows a terminal window with the following output:

```
=> => exporting layers
=> => writing image sha256:e450db80ec74c9aa0d33cf6035aaba350dbe1c72f0cc2cfeb0
=> => naming to docker.io/library/task_container_new
[+] Running 1/1
 ✓ Network image_making_default  Created
 ⚡ Container task_container_new  Creating
Error response from daemon: Conflict. The container name "/task_container_new" 54cf548708a674366a3d8c7d6a50b246da24ae9a5d0f99834a3e28". You have to remove (or e that name.
● gulinyilmaz@Gulins-Laptop image_making % cd ..
● gulinyilmaz@Gulins-Laptop devops_project % cd ..
● gulinyilmaz@Gulins-Laptop Desktop % docker image ls
REPOSITORY          TAG      IMAGE ID      CREATED        SIZE
task_container_new  latest   e450db80ec74  34 minutes ago  517MB
<none>              <none>  6dc986d29941  34 minutes ago  517MB
task_container       latest   cda8a4141c2eb  About an hour ago  520MB
<none>              <none>  f47fd7e13486  About an hour ago  520MB
my_spring           latest   73d0eff1de2  About an hour ago  520MB
my_spring_react     latest   bdef126a8355  About an hour ago  517MB
mongo               latest   35e5c11c442d  10 days ago   721MB
ubuntu              latest   2b7cc08cdedb  13 days ago   69.2MB
nginx               latest   760b7cba31e   3 weeks ago   192MB
alpine              latest   ace17d5d883e  6 weeks ago   7.73MB
mysql               latest   e68e2614955c  7 weeks ago   638MB
httpd               latest   50a1bd9b297f  7 weeks ago   194MB
hello-world         latest   ee301c921b8a  10 months ago  9.14kB
centos              latest   e6a0117ec169  2 years ago   272MB

```

Docker-compose ve Dockerfile kullanılarak image oluşturuldu.

```

centos           latest   2020/11/10 2 years ago    272MB
● gulinyilmaz@Gulins-Laptop Desktop % docker login
Authenticating with existing credentials...
Login Succeeded
● gulinyilmaz@Gulins-Laptop Desktop % docker image tag task_container_new:latest gulinyilmaz/new_docker_image
● gulinyilmaz@Gulins-Laptop Desktop % docker push gulinyilmaz/new_docker_image
Using default tag: latest
The push refers to repository [docker.io/gulinyilmaz/new_docker_image]
2735343707ea: Pushed
5b3adafc0d75: Pushed
413bf99eec92: Mounted from library/openjdk
2c457ba2cc4c: Mounted from library/openjdk
67ea23ef4616: Mounted from library/openjdk
latest: digest: sha256:0ee2680e4ba39cf7b1c9a9b9fc200a265b441fdf0c2255daf656760d3a1f55e3 size: 1779

```

Yeni oluşturulan bu image DockerHub'a pushlandı.

The screenshot shows the Docker Hub interface for the repository `gulinyilmaz/new_docker_image`. The top navigation bar includes links for Explore, Repositories (which is selected), and Organizations. A search bar and various icons are also present.

General Tab:

- Repository Name: `gulinyilmaz/new_docker_image`
- Last updated: 1 minute ago
- Description: Trying out this platform 🖊

Docker commands:

To push a new tag to this repository:

```
docker push gulinyilmaz/new_docker_image:tagname
```

Automated Builds:

M	TAG	OS/ARCH	Last pull	Compressed Size
E	latest	linux/arm64/v8	---	249.5 MB
A	Digest			
S	dee2680e4ba3			

Tags:

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest	Operating system: linux Architecture: arm64	Image	---	a minute ago

[See all](#)

```
latest: digest: sha256:0ee2680e4ba39cf7b1c9a9b9fc200a265b441fdf0c2255daf656760d3a1f55
gulinyilmaz@Gulins-Laptop Desktop % docker pull gulinyilmaz/new_docker_image
Using default tag: latest
latest: Pulling from gulinyilmaz/new_docker_image
Digest: sha256:0ee2680e4ba39cf7b1c9a9b9fc200a265b441fdf0c2255daf656760d3a1f55e3
Status: Image is up to date for gulinyilmaz/new_docker_image:latest
docker.io/gulinyilmaz/new_docker_image:latest
gulinyilmaz@Gulins-Laptop Desktop %
```

Image DockerHub'dan pullandi.

38. Adım: SonarQube

- SonarQube bir yazılım kalitesi yönetim platformudur. Kod kalitesini ölçmek, projeyi izlemek ve iyileştirmek için bu uygulamaya başvurulur. Open source'tur. CI/CD süreçlerine entegre edilebilir. Kodu çalıştırmadan önce statik kod analizi yapar ve kalitesini değerlendirir, sorunlarını bize gösterir. Projelerin sürdürülebilirlik, performans ve güvenlik konusunda geliştirilmesi için faydalı bir araçtır.



39. Adım

The screenshot displays two windows illustrating the SonarQube analysis process for a Java project.

Left Window (Terminal):

```
[INFO] Sensor Java CPD Block Indexer
[INFO] Sensor Java CPD Block Indexer [done] | time=28ms
[INFO] SCM Publisher No SCM system was detected. You can use the 'sonar.scm.provider' property to explicitly specify it.
[INFO] CPD Executor 32 files had no CPD blocks
[INFO] CPD Executor Calculating CPD for 34 files
[INFO] CPD Executor CPD calculation finished [done] | time=4ms
[INFO] Analysis report generated in 66ms, dir size=366.4 kB
[INFO] Analysis report compressed in 161ms, zip size=186.6 kB
[INFO] Analysis report uploaded in 29ms
[INFO] ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=spring_boot_sonar
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at http://localhost:9000/api/ce/task?id=Ay47eA76VLYBQ_JvFS
[INFO] Analysis total time: 13.326 s
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 34.854 s
[INFO] Finished at: 2024-03-13T20:46:39+03:00
[INFO] -----
gulinylimaz@Gulins-Laptop FullStackDeveloper11-master %
```

Right Window (SonarQube UI):

The SonarQube interface shows the analysis results for the 'FullStackDeveloper11' project. The 'Issues' tab is selected, displaying 216 issues across various components:

- Issues in new code:** 212 issues, categorized by Type (Bug, Vulnerability, Code Smell) and Severity (Blocker, Critical, Major, Minor, Info).
- Comments:** 6 issues, all of which are Code Smells labeled as unused.
- src/.../FullStackDeveloper11/Application.java:** 1 issue, a Code Smell labeled as unused.
- src/.../Annotation/AnnotationUniqueEmailAddress.java:** 1 issue, a Code Smell labeled as unused.
- src/.../Annotation/AnnotationUniqueRoleName.java:** 1 issue, a Code Smell labeled as unused.
- src/.../Annotation/AnnotationValidation.java:** 1 issue, a Code Smell labeled as unused.

The interface also shows filters for Issues, Security Hotspots, Measures, Code, and Activity, along with Project Settings and Project Information sections.

40. Adım

The screenshot shows the SonarQube dashboard for the project "FullStackDeveloper11". The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and a search bar. A message indicates the last analysis was on March 13, 2024, at 8:46 PM, for Version 0.0.1-SNAPSHOT.

The main content area displays the "Overall Code" measures. The "New Code" tab is selected. The "Overall Code" section shows:

- 1 Bug (Reliability: B)
- 3 Vulnerabilities (Security: E)
- 5 Security Hotspots (Reviewed: 0.0%, Security Review: E)
- 3d 7h Debt
- 212 Code Smells (Maintainability: A)
- Coverage: 0.0% (Coverage on 485 Lines to cover)
- 1 Unit Tests
- Duplications: 0.0% (Duplications on 1.9k Lines)
- 0 Duplicated Blocks

A note at the bottom encourages setting up analysis in favorite CI tools.

41. Adım: GitLab

- GitLab bir yazılım geliştirme platformudur ve git tabanlı versiyon kontrolü yapmaya yarar. CI/CD süreçlerini destekler, wiki, proje denetimi ve takibi için özellikler sunar. Ayrıca docker ve Kubernetes gibi konteyner teknolojileriyle uyumludur. Güvenlik açıklarını tespit etmek ve düzenlemek için bazı araçlara da sahiptir.



GitLab

42. Adım: GitLab CI/CD

- GitLab CI/CD, GitLab'e entegre edilmiş ve Continuous Integration ile Continuous Delivery süreçlerini destekleyen bir özelliktir. Yazılım geliştirme sürecini otomatize eder, hızlı ve güvenlikli bir şekilde kodun test edilmesi, entegre edilmesi ve dağıtımasına olanak sağlar. GitLab CI otomatik testleri yapar ve kullanıcıya hızlı geri bildirim sunar. CD süreci ise CI süreci başarılı geçerse otomatik olarak dağıtımını gerçekleştirir.

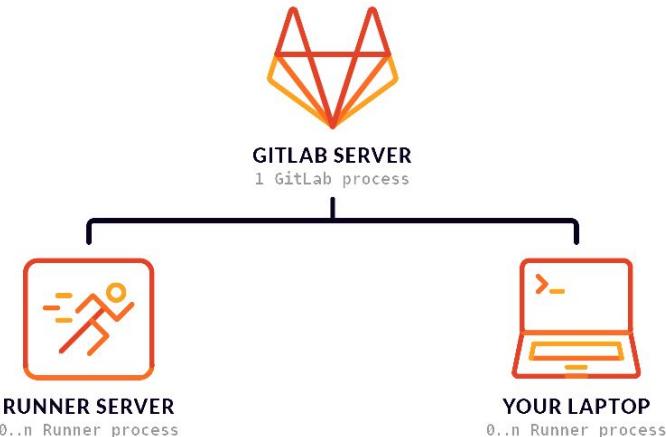


CI  CD

The text "CI" and "CD" is in blue, separated by a yellow lightning bolt icon.

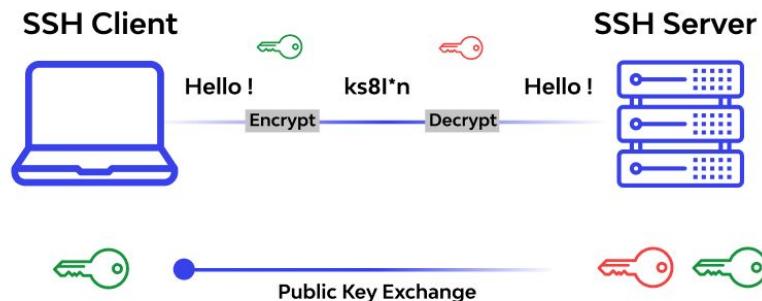
43. Adım: GitLab Runner

- GitLab Runner, CI/CD süreçlerini otomatize etmek için kullanılır. Pipeline'daki işleri runlamak için GitLab CI/CD ile çalışan bir araçtır. Pipeline için gerekli ortamı sağlar ve bu işleri bir konteyner ya da makine üzerinde çalıştırır. CI/CD işlerini çalıştırması için .gitlab-ci.yml dosyasını kullanır. Bu dosyadaki tanımlı işler projenin başlatılması, test edilmesi, paketlenmesi ve dağıtılması gibi aşamalardan oluşur. GitLab runner sayesinde işler için gerekli ortam oluşur, dış kaynaklara erişim sağlanır ve kullanıcı arayüz üzerinden pipeline ilerlemesini izleyebilir.



44. Adım: SSH

- Açılmı Secure Shell olan SSH kullanıcılarının sunucularına internet üzerinden erişmesini, kontrol edebilmesini ve düzenlemeler yapabilmesini sağlayan bir uzaktan yönetim güvenlik protokolüdür. Güvenliği sağlayan bir şifreleme teknigi vardır. Bu teknik sayesinde kullanıcılar gelen ve giden verileri korur. İki sunucu arasında gerçekleşen iletişim, uzaktaki bir kullanıcının kimliğini doğrulayan yüksek şifreleme yöntemiyle korunur.



45. Adım (GitLab ve GitHub SSH)

The screenshot shows a dark-themed GitLab profile page for a user named "Gülin Yılmaz (gulin.yilmaz)". The sidebar on the left includes links for "Public profile", "Account", "Appearance", "Accessibility", "Notifications", "Access", "Billing and plans", and "Emails". The main content area is titled "SSH keys" and displays a list of authentication keys. One key is visible: "macbook_key" (SSH), added on Mar 9, 2024, last used within the last week (Read/write).

SSH Key: glnylmaz@gmail.com

This screenshot shows the "Key details" page for the SSH key "macbook_key". It includes sections for "Key details", "SSH Key", and "Fingerprints".

Usage type	Created	Last used	Expires
Authentication & Signing	Mar 13, 2024 3:50pm	Never	Mar 13, 2025 12:00am

SSH Key:
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQACnTdceXWzWKBosib1HFmhxgbJ2DrjQCdFSE/kZLz9lDF0HciQrH3B7k/Ak+RbAm5VALEQJ8ngpaWtB9UGCftP4oel7Tt6cy

Fingerprints:

MD5	fa:fa:93:34:36:ef:00:a3:5b:46:d4:12:96:b3:0e:7f
SHA256	BaNheoKwN7M3/6A4yuY/KBz0F5dJFh1h8I0FxnYk+Kg

46. Adım: Prometheus & Jenkins

- Prometheus open source bir monitoring ve alerting sistemidir. Prometheus metrik verileri zaman serisi şeklinde depolar ve bu sayede performans verileri zamana göre izlenip analiz edilebilir. Metrik verileri organize etmek ve filtrelemek için labeling kullanılır. Gerektiği zamanlarda örneğin belirtilen değerlerin aşılması durumunda uyarı sistemleriyle bildirim gönderebilir.
- Jenkins ise yazılım geliştirme süreçlerini otomatize etmek için kullanılır ve open source bir CI/CD aracıdır. Yazılımın sürekli olarak test edilmesi ve dağıtılmasını sağlar. Birçok eklenti ile uyumludur, cloud platformları , sürüm kontrol sistemleri ve çeşitli araçlarla entegre edilebilir. İşlerin başarılı veya başarısız olmasına göre uyarılar gönderebilir.



Prometheus



Jenkins

47. Adım: Ansible

- Ansible, IT altyapısını otomatikleştirmek, yapılandırmak ve yönetmek için kullanılan bir otomasyon aracı ve yapılandırma yönetimi sistemidir. Bu açık kaynaklı yazılım, sunucu yapılandırması, uygulama dağıtıımı ve yazılım güncellemeleri gibi çeşitli görevleri basit ve anlaşılır senaryolarla destekleyerek etkili bir şekilde gerçekleştirebilir. YAML tabanlı bir dil kullanması, kullanıcıların işlemleri daha anlaşılır ve yönetilebilir hale getirmelerine olanak tanır.



48. Adım: Kubernetes

- Kubernetes, konteyner orkestrasyonu için kullanılan open source bir platformdur. Docker gibi konteyner teknolojileri kullanılarak geliştirilen uygulamaların dağıtımını, ölçeklendirilmesini ve yönetimini kolaylaştmak için tasarlanmıştır. Google mühendisleri tarafından 2014 yılında geliştirilmiştir. Kubernetes, uygulamaların güvenli, hızlı ve tutarlı bir şekilde dağıtılmamasını sağlamak için bir dizi otomatikleştirilmiş işlevi yönetir.
- En önemli bileşenleri kümeler, düğümler ve kontrol düzlemdir.
- Konteynerler ile uygulamalarınızın çalışmak için ihtiyaç duydukları her şeyle paketlendiğinden emin olabilirsiniz. Ama genellikle mikro hizmetler içeren konteynerler ekledikçe, bunları Kubernetes kullanarak otomatik olarak yönetebilir ve dağıtabilirsiniz.



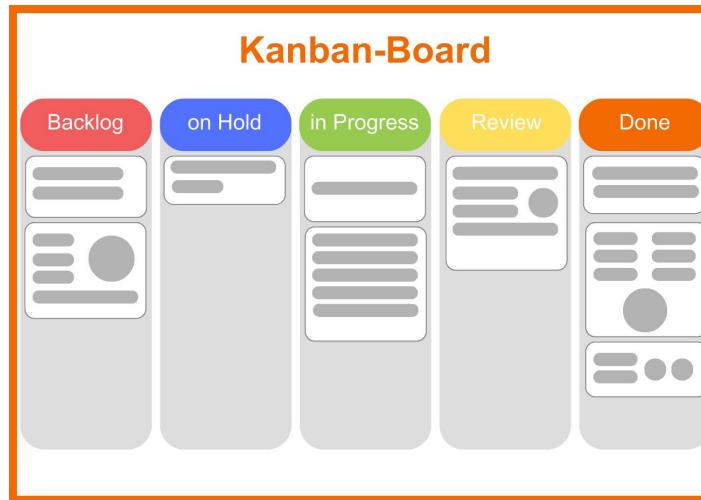
kubernetes

49. Adım: Datadog Monitoring

- Datadog Monitoring, cloud temelli bir izleme ve performans analizi çözümüdür. Bu çözüm, uygulamaların, altyapının ve ağın performansını değerlendirmek, sorunları tanımlamak ve çözmek amacıyla kullanılır. Datadog Monitoring, gerçek zamanlı veri analizi, etkili görselleştirme araçları vealarmlar aracılığıyla projeleri yönlendirir ve geliştiricilere hizmet verir .

50. Adım: Kanban

- Kanban, Japonca'da 'görsel tablo' veya 'görsel işaret' anlamına gelir. İlk olarak Toyota arafından geliştirilip uygulanmıştır. Bu yöntem, başlangıçta just-in-time üretim planlamasında kullanılmıştır ve üretim sürecinde malzeme ile ürünlerin akışını düzenlemek amacıyla benimsenmiştir. Kanban sistemi, gerekli malzeme ve ürünlerin doğru miktarlarda ve doğru zamanda temin edilmesini sağlar. Bu sayede üretim sürecinde gereksiz stok birikimini önler. Aynı zamanda Kanban sistemi, sorunların hızlı bir şekilde belirlenmesine ve çözülmesine de olanak tanır.



51. Adım: DevOps süreçleri

- DevOps, Development ve işletme Operations ekipleri arasındaki işbirliğini artırmayı, yazılım dağıtımını hızlandırmayı ve sürekli iyileştirmeyi hedefleyen bir kültür, hareket ve pratikler bütünüdür.
- Süreçleri:
 - Planlama
 - Kod
 - Sürekli entegrasyon (CI)
 - Sürekli Teslim ve Dağıtım (CD)
 - Monitor etmek



52. Adım: Orchestration Lab

The screenshot shows a web-based interface for an orchestration lab. On the left, there's a sidebar with a timer at 03:30:02, a 'CLOSE SESSION' button, and a list of instances: node1, node2, node3, node4, and node5. Below the instances is a 'GIVE FEEDBACK' section. The main area displays session details for instance node1, including IP (192.168.0.8), Memory usage (1.54% / 61.72MiB / 3.906GiB), CPU usage (2.09%), and an SSH connection status. The terminal window shows the following log output:

```
[node1] (local) root@192.168.0.8 ~
$ #####
#           WARNING!!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY! discouraged. Any consequences of doing so are
# completely the user's responsibilites.
#
# The PWD team.
#####
[node1] (local) root@192.168.0.8 ~
$ docker swarm init --advertise-addr 192.168.0.8
Swarm initialized: current node (7qh04shsi6eqd9vaswrx09w4) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-4tei3emkk3u3d0wt89o0lua4roxwax9sf53nbi5ns2y9bsti2x-6af00gvyjrsssmi243ubk6b5y 192.168.0.8:2
377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

[node1] (local) root@192.168.0.8 ~
$ docker swarm join-token manager
To add a manager to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-4tei3emkk3u3d0wt89o0lua4roxwax9sf53nbi5ns2y9bsti2x-chbisbaogog20rzptyv49ozrb 192.168.0.8:2
377
```

+ ADD NEW INSTANCE

192.168.0.8
node1

192.168.0.7
node2

192.168.0.6
node3

192.168.0.5
node4

192.168.0.4
node5

GIVE FEEDBACK

Managerlar eklendi.

SSH
ssh ip172-18-0-4-cnmqchqim2rg00cneet0@direct.labs.pl

DELETE

EDITOR

```
#####
# WARNING!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY! discouraged. Any consequences of doing so are
# completely the user's responsibilites.
#
# The PWD team.
#####
[node2] (local) root@192.168.0.7 ~
$ #####
# WARNING!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY! discouraged. Any consequences of doing so are
# completely the user's responsibilites.
#
# The PWD team.
#####
[node2] (local) root@192.168.0.7 ~
$ docker swarm join --token SWMTKN-1-4tei3emkk3u3ddocker swarm
bsti2x-chbisbaogog20rzrptyv49ozrb 192.168.0.8:2377
This node joined a swarm as a manager.
[node2] (local) root@192.168.0.7 ~
$ █
```

192.168.0.8
node1

192.168.0.7
node2

192.168.0.6
node3

192.168.0.5
node4

192.168.0.4
node5

GIVE FEEDBACK

DELETE

EDITOR

```
#####
# WARNING!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY! discouraged. Any consequences of doing so are
# completely the user's responsibilites.
#
# The PWD team.
#####
[node3] (local) root@192.168.0.6 ~
$ #####
# WARNING!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY! discouraged. Any consequences of doing so are
# completely the user's responsibilites.
#
# The PWD team.
#####
[node3] (local) root@192.168.0.6 ~
$ docker swarm join --token SWMTKN-1-4tei3emkk3u3d0wt89o07
This node joined a swarm as a manager.
[node3] (local) root@192.168.0.6 ~
$ █
```

192.168.0.8
node1

192.168.0.7
node2

192.168.0.6
node3

192.168.0.5
node4

192.168.0.4
node5

GIVE FEEDBACK

DELETE

EDITOR

```
#####
#           WARNING!!!!
# This is a sandbox environment. Using
# is HIGHLY! discouraged. Any consequen
# completely the user's responsibilites
#
# The PWD team.
#####
[node4] (local) root@192.168.0.5 ~
$ #####
#           WARNING!!!!
# This is a sandbox environment. Using
# is HIGHLY! discouraged. Any consequen
# completely the user's responsibilites
#
# The PWD team.
#####
[node4] (local) root@192.168.0.5 ~
$ docker swarm join --token SWMTKN-1-4t
7
This node joined a swarm as a worker.
[node4] (local) root@192.168.0.5 ~
$ 
```

Workerlar eklendi.

192.168.0.8
node1

192.168.0.7
node2

192.168.0.6
node3

192.168.0.5
node4

192.168.0.4
node5

GIVE FEEDBACK

DELETE

EDITOR

```
#####
#           WARNING!!!!
# This is a sandbox environment. Using
# is HIGHLY! discouraged. Any consequen
# completely the user's responsibilites
#
# The PWD team.
#####
[node5] (local) root@192.168.0.4 ~
$ #####
#           WARNING!!!!
# This is a sandbox environment. Using
# is HIGHLY! discouraged. Any consequen
# completely the user's responsibilites
#
# The PWD team.
#####
[node5] (local) root@192.168.0.4 ~
$ docker swarm join --token SWMTKN-1-4t
7
This node joined a swarm as a worker.
[node5] (local) root@192.168.0.4 ~
$ 
```

```
[node1] (local) root@192.168.0.8 ~
```

```
$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE	VERSION
7qh04shsi6eqd9vaswxrx09w4 *	node1	Ready	Active	Leader	24.0.7	
wgaznskgc3biz5esat6zjquka	node2	Ready	Active	Reachable	24.0.7	
l58zty39jwkez73h95mzy644v	node3	Ready	Active	Reachable	24.0.7	
t1zhaguc990c6hh8zh4k450h6	node4	Ready	Active		24.0.7	
j2r2py5bbxb6d8p24tz560h7k	node5	Ready	Active		24.0.7	

```
[node1] (local) root@192.168.0.8 ~
```

```
[node1] (local) root@192.168.0.8 ~ $ docker service create --name web_server -p 7777:80 --replicas=9 nginx
```

```
k7dsvhhnwfzycj504bjtqs3hl
```

```
overall progress: 9 out of 9 tasks
```

```
1/9: running
```

```
2/9: running
```

```
3/9: running
```

```
4/9: running
```

```
5/9: running
```

```
6/9: running
```

```
7/9: running
```

```
8/9: running
```

```
9/9: running
```

```
verify: Service converged
```

```
[node1] (local) root@192.168.0.8 ~ $ docker service ls
```

ID	NAME	MODE	REPLICAS	IMAGE	PORTS
k7dsvhhnwfzy	web_server	replicated	9/9	nginx:latest	*:7777->80/tcp

9 replika
oluşturuldu.

```
[node1] (local) root@192.168.0.8 ~
$ docker service ls
ID          NAME      MODE      REPLICAS  IMAGE
k7dsvhhnwfzy  web_server  replicated  9/9      nginx:latest  *:7777->80/tcp
[node1] (local) root@192.168.0.8 ~
$ docker ps
CONTAINER ID  IMAGE      COMMAND
b183b4eede8a  nginx:latest "/docker-entrypoint...."
2 minutes ago
Up 2 minutes  80/tcp
web_server.5.c54m0uxdq803w72e8erq
27o6r
```

```
[node2] (local) root@192.168.0.7 ~
docker ps
CONTAINER ID  IMAGE      COMMAND
9adb9d13abd6  nginx:latest "/docker-entrypoint...."
2 minutes ago
Up 2 minutes  80/tcp
web_server.8.mog
a4da3
7d4861990cdb  nginx:latest "/docker-entrypoint...."
2 minutes ago
Up 2 minutes  80/tcp
web_server.4.tfc
tzj4d
```

Örnek olarak node1 ve node2'nin containerları.

```
[node1] (local) root@192.168.0.8 ~
$ docker service ps web_server
ID          NAME      IMAGE      NODE      DESIRED STATE  CURRENT STATE
v5tfmjlvkbb3  web_server.1  nginx:latest  node3    Running
                Running 3 minutes ago
bdm7vcnpz1xm  web_server.2  nginx:latest  node4    Running
                Running 3 minutes ago
8gvd0izc4h6r  web_server.3  nginx:latest  node4    Running
                Running 3 minutes ago
tfcmz7mgsf3o  web_server.4  nginx:latest  node2    Running
                Running 3 minutes ago
c54m0uxdq803  web_server.5  nginx:latest  node1    Running
                Running 3 minutes ago
rtaf87ewtgd7  web_server.6  nginx:latest  node5    Running
                Running 3 minutes ago
yidir5i71hkc  web_server.7  nginx:latest  node5    Running
                Running 3 minutes ago
mog4s7ibuxbf  web_server.8  nginx:latest  node2    Running
                Running 3 minutes ago
0248zs8h178h  web_server.9  nginx:latest  node3    Running
                Running 3 minutes ago
```

53. Adım

```
##### Docker Installation #####
+ brew install docker
-- Downloading https://formulae.brew.sh/api/formula.json
#####
==== Downloading https://formulae.brew.sh/api/cask.json 100.0%
==== Downloading https://formulae.brew.sh/api/cask.json 100.0%
Warning: Treating docker as a formula. For the cask, use homebrew/cask/docker or specify the `--cas k` flag.
Warning: docker 25.0.4 is already installed and up-to-date.
To reinstall 25.0.4, run:
  brew reinstall docker
+ sleep 2
+ echo -e '\n##### Docker Compose #####'

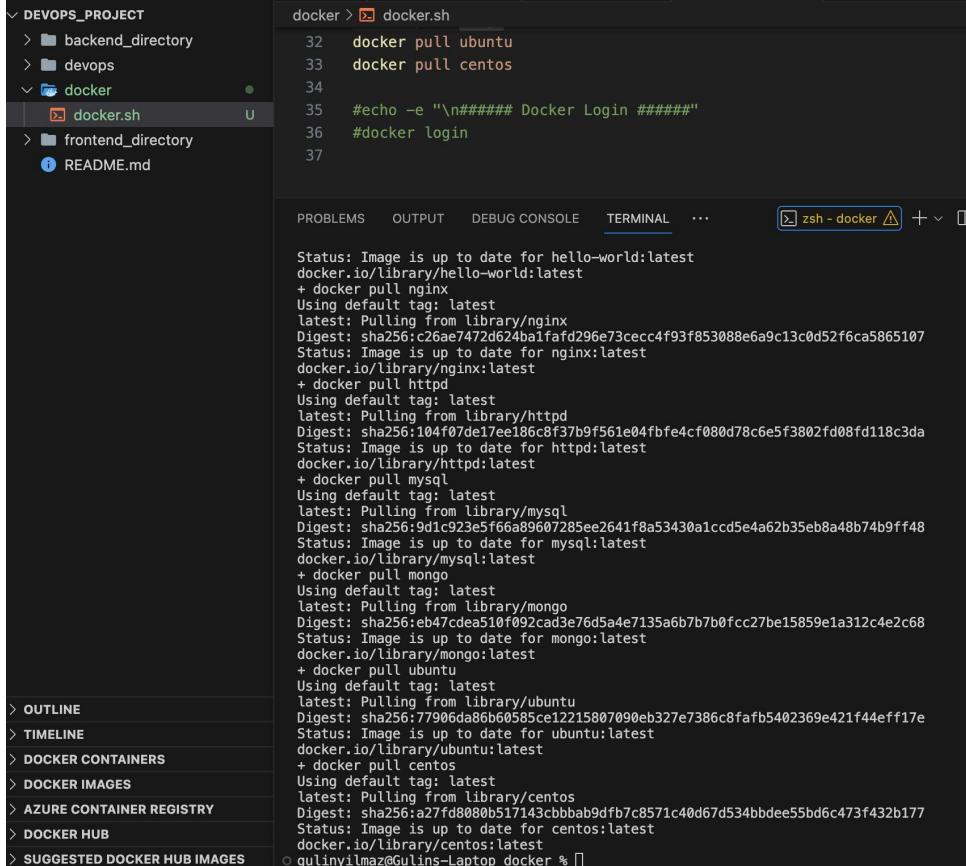
#####
+ brew install docker-compose
Warning: docker-compose 2.24.7_1 is already installed and up-to-date.
To reinstall 2.24.7_1, run:
  brew reinstall docker-compose
+ sleep 2
+ echo -e '\n##### Docker User Permissions #####'

#####
Docker User Permissions #####
+ sudo dscl . append /Groups/docker GroupMembership gulinyilmaz
Password:
append: Invalid Path
<dscl _cmd> DS Error: -14009 (eDSUnknownNodeName)
+ sleep 2
+ echo -e '\n##### Docker Bash Completion #####'

#####
Docker Bash Completion #####
+ brew install bash-completion
Warning: bash-completion 1.3.3 is already installed and up-to-date.
To reinstall 1.3.3, run:
  brew reinstall bash-completion
+ sleep 2
+ echo -e '\n##### Docker Pulling Images #####'

#####
Docker Pulling Images #####

```



The screenshot shows a terminal window with several Docker pull commands being executed. The commands are:

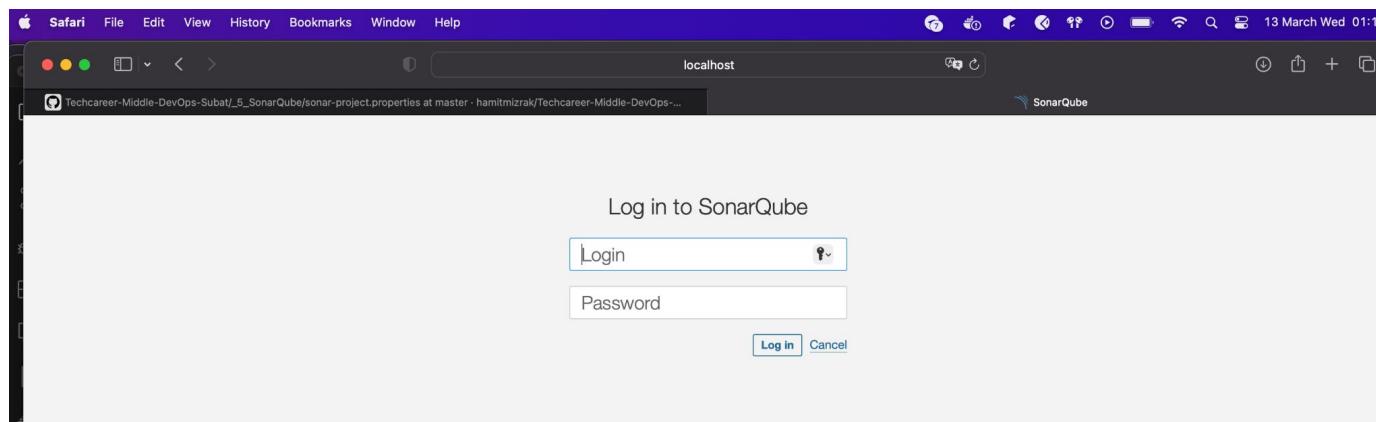
- docker pull ubuntu
- docker pull centos
- #echo -e "\n##### Docker Login #####"
- #docker login

The terminal also displays status messages for each image being pulled, such as "Status: Image is up to date for hello-world:latest" and "Status: Image is up to date for nginx:latest".

docker.sh ile Docker kurulumu yapıldı. Repository'ye pushlandı.

54. Adım

```
gulinyilmaz@Gulins-Laptop java_sonarqube_project % docker-compose ps
● NAME           IMAGE          COMMAND            SERVICE      CREATED        STATUS
● PORTS          db             "docker-entrypoint.s..."  db          19 hours ago  Up 19 hours
java_sonarqube_project-db-1    postgres:12      5432/tcp
java_sonarqube_project-sonarqube-1  sonarqube:lts-community  "java -jar /opt/sonarqube/dock...  sonarqube  19 hours ago  Up 19 hours
gulinyilmaz@Gulins-Laptop java_sonarqube_project %
```





Administration

Configuration ▾ Security ▾ Projects ▾ System Marketplace

Home Dir	/opt/sonarqube
Data Dir	/opt/sonarqube/data
Temp Dir	/opt/sonarqube/temp
Processors	8

Database

Database	PostgreSQL
Database Version	12.18 (Debian 12.18-1.pgdg120+2)
Username	sonar
URL	jdbc:postgresql://db:5432/sonar
Driver	PostgreSQL JDBC Driver
Driver Version	42.5.1
Default transaction isolation	TRANSACTION_READ_COMMITTED

Server Push Connections

SonarLint Connected Clients	0
-----------------------------	---