

# TECHCAREER DEVOPS BİTİRME PROJESİ

The logo for techcareer.net, featuring the word "tech" in dark blue, "career" in green, and ".net" in dark blue, all in a sans-serif font.

# Veriliş Tarihi

- ▶ 1 Mart 2024 tarihinde verildi.
- ▶ 13 Mart 2024 saat: 23:59 kadar github adresinize pushlama yapabilirsiniz.
- ▶ 14 Mart 2024 tarihinde saat 11:00 Sunum yapılacaktır.
- ▶ **Not:** Sertifika alabilmeniz için Bitirme projesi sunmanız gerekiyor.
- ▶ Şimdiden başarılar diliyorum.

# Bitirme Projesi

- ▶ Bitirme projemizi belirli adımlardan geçmektedir. Bu adımlar bir önceki adıma bağlı olduğu için sizlere tavsiyem adım adım ilerlemeniz.
- ▶ Her bir adımda gereken kodlar varsa altına yazmak , yorumlama ,resimle belgeleme varsa yazalım,ekleyelim lütfen
- ▶ Aşağıdaki her bir işlemi açıklamaları internetten araştırıp sonrasında kendi cümlelerimizde tanımlama yapalım.

- ▶ **1.adım:** Sanal Makine üzerinde (VMBOX) Linux Ubuntu 22.0.4 LTS kuralım (Eğer Linux ve/veya Mac varsa Sanal makine kurmanıza gerek yok)
- ▶ **2.adım:** kurulumları yapalım.
- ▶ Git,JDK-17 , Maven, Apache Tomcat, Docker, Visual Studio codes, derste yüklediğimiz pluginleri yükleyelim (GitHub,maven,deployment vs.)
- ▶ **3.adım:** port ayarları her bir uygulama için farklı portta olduğunda emin olalım → Linux komut terminalinden derste yaptımız
- ▶ `netstat -nlptu`
- ▶ **4.adım:** GitHub repository üzerinden devops\_project adında repository açalım.
- ▶ GitHub repository ReadMe.md adında dosya oluşturulun uygun formatta yazalım sonrasında local bilgisayarımızda pull yapalım

# GİT-1

- ▶ 5.adım: Git nedir ? VCS açılımı nedir ? iyi bir commit özelliklkeri nelerdir ?
  
- ▶ .adım: git ayarlarınıı user.name, user.password yapalım.
  1. `git config --global user.name "hamitmizrak"`
  2. `git config --global user.email "hamitmizrak@gmail.com"`
  
- ▶ 5.adım: Linux komutlarıyla dizin adı "devops" ve dosya adı "jenkins.txt" oluşturalım ve "DevOps öğreniyorum" yazalım.
  1. `mkdir devops`
  2. `cd devops`
  3. `cat >> jenkins.txt`
  4. DevOps öğreniyorum
  5. q

# GİT-1

- ▶ 7.adım: Git staged area ve unstaged area nedir ?
- ▶ 8.adım: bu dizindeki dosyaları oluşturduğumuz github repository gönderelim. NOT: main adında branch oluşturalım.
  1. `git add .`
  2. `git commit -m "first commit"`
  3. `git branch -M "main"`
  4. `git remote add origin https:...`
  5. `git push -u origin master`

# GİT

► **9.adım:** oluşturduğumuz github repository local bilgisayarımızda bir yere clone yapalım

1. `pwd`
2. `git clone githubURL`

► **10.adım:** en son commitimizin commit içeriğini değiştirilelim (tips: `--amend`)

► **11.adım:** `git log` ve `git status` ne iş yapıyordu ?

1. `git log`: bütün commitleri göstermek
2. `git status`: dosyalar ne durumda

# GİT

- ▶ **12.adım:** backend adında branch oluşturalım. bu branche bir takım dizinler ekleyelim. commit yapalım.
- ▶ merge işleminde fast-forward yapalım

1. `git add .`
2. `git commit -m "merge öncesinde commit"`
3. `git branch backend`
4. `git checkout backend`
5. `git add .`
6. `git commit -m "backend"`
7. `git checkout main`
8. `git merge backend`



# GİT

► 13.adım: Git GUI ve Git CLI nedir ?

► 14.adım: frontend adında branch oluşturalım. bu branche bir takım dizinler ekleyelim. commit yapalım.

► merge işleminde no-fast-forward yapalım

1. `git add .`
2. `git commit -m "merge öncesinde commit"`
3. `git branch frontend`
4. `git checkout frontend`
5. `git add .`
6. `git commit -m " frontend "`
7. `git checkout main`
8. `git merge frontend --no-ff`

# GİT

- ▶ **15.adım:** Başka bir github repository açalım ve bu sefer derste öğrendiğimiz SSH-KEY ile github veri gönderme yapalım
- ▶ Linux komutlarıyla dizin adı "devops" ve dosya adı "jenkins.txt" oluşturalım ve "DevOps öğreniyorum" yazalım.
- ▶ **16.adım:** git stash nedir ?
- ▶ **17.adım:** projelerimizi pushlama yaparken acil.txt adında bir iş geldi ve bu iş öncelik olduğu söylendi var olan add yapılmış dosyalarımızı commitleme yapmadan özel bir alanda saklama yapalım tabi bunu git stash ile yapalım.
- ▶ stash adı araf olsun
- ▶ acil.txt işimizi bitirdik bunu pushladık
- ▶ stash araf adındaki stash çağırıp işleyip ve sonrasında silelim.

1. `git add .`
2. `git stash --save "araf"`
3. `git stash list`
4. `git stash apply araf`
5. `git add .`
6. `git commit -m "stash sonrası commit"`
7. `git push -u origin main`
8. `git stash drop araf`

# GİT

- ▶ **18.adım:** `git log --all --oneline --decorate --graph` komutunu `graph` adından `alias` kullanarak kısaltalım ve
- ▶ `git config --global alias.graph "log --all --graph --decorate --oneline"`
- ▶ `git graph`
- ▶ **19.adım:** Rebase ile merge arasındaki fark nedir ?
- ▶ **20.adım:** Git Conflict nedir ? Bir conflict yediğimizde ne yapmamız gerekiyor ?
- ▶ **21.adım:** `git ignore` nedir ?
- ▶ `git` tarafından takip edilmesini istemediğimiz dizinler ve/veya dosya için kullanılır.
- ▶ **22.adım:** `git tag v1.1` ? Bu komu ne iş yapar ?
- ▶ `git tag`: projemizde tag vermemize yarar
- ▶ **23.adım:** `git diff 3b2f0ab 5a2b8de` bu komu ne iş yapar ? NOT: 3b2f0ab 5a2b8de commit numarası
- ▶ `git diff`: iki commit veya dizinlerdeki değişiklikleri göstermek

# DevOps soruları

- ▶ 24.adım: DevOps kültür felsefesi nedir ?
- ▶ 2.adım: DevOps açılımı ?
- ▶ 26.adım: DevOps Yöntemlerinden Continuous /Continuous Delivery-Deployment süreçler hakkında bildilerimizi yazalım
- ▶ 27.adım: DevOps Yöntemlerinden git nedir ?
- ▶ 28.adım: DevOps Yöntemlerinden Agile nedir ?
- ▶ 29.adım: DevOps Continuous monitoring nedir ?
- ▶ 30.adım: Aşağıdaki içeriklerin açılımları ve ne olduğunu yazalım. ?
- ▶ XML nedir ?
- ▶ JSON nedir ?
- ▶ Yaml nedir ?
- ▶ http nedir
- ▶ server nedir ?
- ▶ Java JDK nedir ?
- ▶ 31.adım: Maven nedir ? maven'de → clean install görevi nedir ?
- ▶ clean install cache bellek temizler ve compiler yapar.

# DevOPS soruları

- ▶ **32.adım:** Docker nedir ? Docker Daemon, Docker CLI ne iş yapıyor ?
- ▶ **33.adım:** Aşağıdaki adımları teker teker yazalım?
- ▶ **34.adım:** - docker search nginx
- ▶ **35.adım:** - docker pull nginx
- ▶ **36.adım:** - nginx yandaki özelliklere sahip containerlar özelliklerine göre yazalım.
- ▶ -80 portundan 6666 portuna yönlendiren,
- ▶ -arka planda çalışan(-d)
- ▶ -yeni adı web\_2 (--name)
- ▶ -container kapatıldıktan sonra silinen komut (--rm)
  
- ▶ **37.adım:** bu github adresindeki veriyi github ile clone yapalım. ve yapılacakları aşağıda yazılmıştır.
- ▶ - git clone URL\_ADDRESS
- ▶ - bu JAR dosyasının Dockerfile ve docker-compose.yml yazarak image oluşturalım.
- ▶ -Bu oluşturduğunuz image kendi dockerHub'ta repository'a pushlayalım.
- ▶ - Kendi repository gönderdiğiniz bu image docker pull ... diyerek tekrardan local bilgisayarına veriyi alalım

# DevOPS soruları

- ▶ 38.adım: SonarQube nedir ?
- ▶ 39.adım: Derlenmiş bir kodun statik kod analizini yapalım. (Çalıştırdınız sayfanın ana görünütüsünü görelim)
- ▶ 40.adım: - Bu statik kodlarda smell code, loop, port vs hataları ekran görüntsünü görelim.
- ▶ 41.adım: - GitLab nedir ?
- ▶ 42.adım: - GitLab nedir CI/CD nedir ?
- ▶ 43.adım: - GitLab Runner nedir ?
- ▶ 44.adım: - SSH nedir ?
- ▶ 45.adım: - GitLab ve GitHub SSH-keygen oluşturarak bilgisayarına bağlayın?
- ▶ 46.adım: - Prometheus nedir, Jenkins nedir ?
- ▶ 47.adım: - Ansible nedir ?
- ▶ 48.adım: - Kubernetes nedir ?
- ▶ 49.adım: - Datadog Monitoring nedir ?
- ▶ 50.adım: - Kanban nedir ?
- ▶ 51.adım: - Devops nedir, Süreçleri nelerdir ?

# Orchestration

- ▶ Derste yaptığımız docker orchestration yapıyorsunuz. Docker Lab
- ▶ 5 tane instance oluşturuyorsunuz.
- ▶ 9 tane nginx replicas oluşturuyorsunuz
- ▶ 1 tane aktif manager
- ▶ 2 tane pasif manager
- ▶ 2 tane worker
- ▶ Not: Bunları yaptıktan sonra ekran görüntülerini sunum dosyanıza ekliyorsunuz.

# Docker Bash Script

► Linux'ta çalıştırmak üzere `docker.sh` yazıyorsunuz.

1. Docker kuracak
2. sudo yetkisi verilecek



# Docker-compose & SonarQube

- ▶ Docker compose üzerinden;
  - ▶ Postgresql, Sonarqube kuruyorsunuz.
  - ▶ Bir Java dosyasının maven komutu oluşturup(hazır kodlarda alabilirsiniz), sonarqube maven komutuyla çalıştırıp java kodlarındaki smell-code, güvenlik açıklarını kontrol ediniz.
- 
- ▶ Not: Bunları yaptıktan sonra **ekran görüntülerini** sunum dosyanıza ekliyorsunuz.