

# Devops Bootcamp Bitirme Projesi Soruları

**Proje github repository linkleri:**

[https://github.com/ekincienes/devops\\_project](https://github.com/ekincienes/devops_project)

<https://github.com/ekincienes/devops>

## **1.adım: Sanal Makine üzerinde (VMBOX) Linux Ubuntu 22.0.4 LTS**

**kuralım** (Eğer Linux ve/veya Mac varsa Sanal makine kurmanıza gerek yok)

*Sunum içerisindeki screenshotlarda mevcuttur.*

## **2.adım: kurulumları yapalım.**

Git,JDK-17, Maven, Apache Tomcat, Docker, Visual Studio codes, derste yüklediğimiz pluginleri yükleyelim (GitHub,maven,deployment vs.)

*Sunum içerisindeki screenshotlarda mevcuttur.*

## **3.adım: port ayarları her bir uygulama için farklı portta olduğunda emin olalım**

netstat -nlptu

*Sunum içerisindeki screenshotlarda mevcuttur.*

## **5.adım: Git nedir ? VCS açılımı nedir ? iyi bir commit özelliklkeri nelerdir ?**

Git, bir versiyon kontrol sistemidir, vcs'nin açılımı da budur. birden fazla developerın aynı proje üzerinde çalışmasını çok yönden kolaylaştırmaya yarar. İyi bir commit ise yapılan değişikliği bir ya da iki kelime ile mesaj atılan committir.

## **6.adım: Linux komutlarıyları dizin adı "devops" ve dosya adı "jenkins.txt" oluşturalım ve "DevOps öğreniyorum" yazalım.**

*Sunum içerisindeki screenshotlarda mevcuttur.*

## 7. adım git staged area ve unstaged area nedir?

Git staged area, git add ile stage kısmına alınan yeni koddaki değişikliklerdir. unstaged area ise stage kısmına alınmayan değişikliklerdir.

## 8.adım: bu dizindeki dosyaları oluşturduğumuz github repository gönderelim. NOT: main adında branch oluşturalım.

Sunum içerisindeki screenshotlarda mevcuttur.

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

Sunum içerisindeki screenshotlarda mevcuttur.

## 10.adım: en son commitimizin commit içeriğini değiştirilelim (tips: --amend)

Sunum içerisindeki screenshotlarda mevcuttur.

## 11.adım: git log ve git status ne iş yapıyordu ?

Git log, atılan commitlerin geçmişini gösterir. git status ise commit edilmemiş değişiklikleri gösterir.

## 12.adım: backend adında branch oluşturalım. bu branche bir takım dizinler ekleyelim. commit yapalım.

merge işleminde fast-forward yapalım

Sunum içerisindeki screenshotlarda mevcuttur.

## 13.adım: Git GUI ve Git CLI nedir ?

Git gui bir arayüz aracılığı ile git kullanmamızı sağlayan tool iken, git CLI command line interface, yani terminalden git komutları yazmamızı sağlayan terminaldir.

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

merge işleminde no-fast-forward yapalım

Sunum içerisindeki screenshotlarda mevcuttur.

## 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.

*Sunum içerisindeki screenshotlarda mevcuttur.*

## 16.adım: git stash nedir ?

Git stash, commit etmek istemediğimiz değişiklikleri tuttuğumuz bir yapıdır. yazdığımız kod bitmeden farklı bir branchte çalışmak veya farklı bir feature eklemek

istediğimizde kullanılabilir.

**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.

*Sunum içerisindeki screenshotlarda mevcuttur.*

**18.adım: git log --all --oneline --decorate --graph komutunu graph adından alias kullanarak kısaltalım.**

## 19.adım: Rebase ile merge arasındaki fark nedir ?

Merge iki branchteki değişiklikleri bir araya getirip yeni bir merge commiti oluştururken, rebase hedef branch üzerine source branch üzerinde yapılan değişikliklerin commitini yazarak ilerler.

## 20.adım: Git Conflict nedir ? Bir conflict yediğimizde ne yapmamız gerekiyor ?

Git conflict bir kodun aynı yerinde bizimle aynı esnada başka birisinin de farklı bir kod yazdığını gösterir. Bu durumda takım lideri ya da proje sorumlusuyla iletişime geçip hangi kodun daha önemli olduğunu seçip onunla ilerlemeliyiz.

## 21.adım: git ignore nedir ?

Git ignore, git repository'sine pushlamak istemediğimiz klasörleri dosyaların ismini yazdığımız bir dosyadır.

## 22.adım: git tag v1.1 ?Bu komu ne iş yapar ?

v1.1 adında bir tag oluşturmaya yarar. bundan sonra atacağımız commitler bu tag altında gözükür.

## 23.adım: git diff 3b2f0ab 5a2b8de bu komu ne iş yapar ?

Git diff, bu iki commit arasındaki yapılan değişiklikleri bize göstermeye yarar, hangi dosyalar üzerinde değişiklik yapıldığını gösterir

## 24.adım: DevOps kültür felsefesi nedir ?

Devops kültürü, developerların ve operasyon ekibinin hep birbiriyle iletişim halinde olmasını ve birbirlerin entegre uyum içerisinde çalışmalarını barındıran bir felsefeden meydana gelir. İçerisinde birçok farklı teknik barındırır.

### *2.adım: DevOps açılımı ?*

Develeopment and Operations.

## 26.adım: DevOps Yöntemlerinden Continous /Continoues Delivery-Deployment süreçler hakkında bildilerimizi yazalım:

Continuous integration, kod üzerinde yapılan değişikliklerin git gibi bir vcs ile yapılması, üzerinde otomatik testler yürütülmesi hatanın önüne geçilmesi gibi yapıları içerirken, continious deployment, yeni bir feature eklendiğinde otomatik olarak bunun canlıya alınmasını sağlayan süreçleri yöneten yapıları içerir.

## 27.adım:DevOps Yöntemlerinden git nedir?

Git bir versiyon kontrol sistemidir. Birden fazla developerın aynı proje içerisinde uyum ve entegrasyon içerisinde çalışmasına, ve proje üzerinde zaman içerisinde yapılan değişikliklerin historysini tutmaya yarayan bir araçtır.

## 28.adım:DevOps Yöntemlerinden Agile nedir?

Agile projeyi küçük parçalara bölerek her parçada feedback alarak geliştirme yapmayı amaçlayarak ilerleme metodolojisi.

## 29.adım: DevOps Continues monitoring nedir?

Continuous monitoring, sürekli olarak sistemin performansını, kaynaklarını, ya da Jenkins üzerindeki pipeline'ı izleme yöntemlerine verilen genel addır. Örnek(datadog, sonarqube,)

## 30.adım: Aşağıdaki içeriklerin açıklamaları ve ne olduğunu yazalım. ?

### *XML nedir?*

Extensible Markup Language, html yapısına benzeyen, ama içerisinde json gibi veri tutmamıza yarayan bir dildir. Json'ın hmtl ize edilmiş hali denebilir.

### *JSON nedir?*

Javascript object notation, içerisinde veri tutmamıza yarayan bir yapıdır. NoSQL veritabanları json veri saklama mantığı üzerine çalışır. Json iç içe bir sürü veri saklayabileceğimiz bir yapıdır.

### *Yaml nedir?*

Ain't markup language, bazı toollar için konfigürasyon verileri içerisinde yazdığımız yapıdır. Örneğin docker yaml dosyası. Python gibi indentation kuralıyla yazıldığı için okuması çok kolaydır.

### *Http nedir?*

Hyper text transfer protocol, web sitelerinin verilerini çağırmamız için hypertextlerden oluşan bir uygulama katmanı bileşenidir. Bir siteye http isteği atılarak o sitenin içeriği fetch edilir.

### *Server nedir?*

Server, web sitelerinin dış dünyaya(internete) açıldığı bilgisayarlar makinalardır. Diğer kullanıcılar bu makinalar üzerinden web sitesine erişim sağlar.

### *Java JDK nedir?*

Java development kit, java geliştirme kiti, java uygulamaları geliştirmemizi, build etmemizi sağlayan kitledir. Geliştirme ve test üzerine toollar içerir.

## 31.adım:Maven nedir?mavende →cleaninstall görevi nedir?

Maven açık kaynak kodlu bir build otomasyon tooludur(java için). Source kodun compile edilmesi veya gerekli dependency'lerin yönetilmesi test scriptlerinin çalıştırılması gibi işlevleri vardır.

Clean install ise maven'a daha önce compile edilmiş bileşenleri sıfırlayıp projeyi tamamen sıfırdan çalıştırmaya yarayan komuttur.

## 32.adım: Docker nedir ? Docker Daemon, Docker CLI ne iş yapıyor ?

Docker, bir containerization teknolojisidir. Linux Kernel üzerinde namespace mimarisini kullanarak, farklı uygulamaları sadece kendilerine yetecek kaynak ayıracak şekilde izole ederek sanal bir şekilde çalıştırmaya yarar. Günümüzde production ortamında son derece avantajlı olan bu teknoloji, birçok şirket tarafından kullanılır hale gelmiştir.

Docker daemon, dockerın driverıdır. Containerların ve image'ların volume'lerin ve diğer core docker bileşenlerinin yönetim merkezidir.

Docker CLI ise docker komutları yazarak bizim bu docker daemon üzerinde işlem yapmamıza yarayan terminal interfaceidir. Mesela container oluşturma, image pull etme veyahut volüme oluşturma gibi komutlar gibi.

## 33.adım: Aşağıdaki adımları teker teker yazalım?

### 34.adım: *-docker search nginx*

Docker-hub üzerinde nginx isimli image'ları arayan komuttur.

### 35.adım: *-docker pull nginx*

Docker-hub üzerinden nginx:latest tagli default image'ı indirmemizi sağlayan komuttur.

### 36.adım: *-nginx yandaki özelliklere sahip containerlar özelliklerine göre yazalım.*

-80 portundan 6666 portuna yönlendiren,

```
docker run --name my_container(container_ismi) -p 80:6666 nginx(image_ismi)
```

-arka planda çalışan(-d)

```
docker run --name my_container(container_ismi) -p 80:6666 -d nginx(image_ismi)
```

-yeni adı web\_2 (--name)

```
docker run --name web_2(container_ismi) -p 80:6666 nginx(image_ismi)
```

-container kapatıldıktan sonra silinen komut (--rm)

```
docker run --name(container_ismi) --rm -p 80:6666 nginx
```

### **37.adım: bu github adresindeki veriyi github ile clone yapalım.**

Sunum içerisindeki screenshotlarda mevcuttur.

### **38.adım: SonarQube nedir ?**

SonarQube, projemizin kodları üzerinde kalite kontrolü ve monitoring yapmamızı sağlayan bir tool'dur.

39.adım: Derlenmiş bir kodun statik kod analizini yapalım. (Çalıştırdığınız sayfanın ana görünümünü görelim)

Screenshotlar içerisinde mevcuttur.

### **40.adım: -Bu statik kodlarda smell code, loop, port vs hataları ekran görüntüsünü görelim.**

Sunum içerisindeki screenshotlarda mevcuttur.

### **41.adım: -GitLab nedir ?**

GitLab, git kullanımı için özelleştirilmiş bir cloud git repositorysidir. Ancak içerisinde bunun da ötesinde bir çok devops toolu da barındırır.

### **42.adım: -GitLab nedir CI/CD nedir ?**

GitLab CI/CD, gitlab üzerinde entegre çalışabilen CI/CD pipeline'ları üretip bunları çalıştırmamızı sağlayan tool'dur. Halihazırda gitlab üzerindeki repository üzerinden veya başka bir repo üstünden çalışabilir.

### **43.adım: -GitLab Runner nedir ?**

GitLab runner, GitLab CI/CD pipeline'ı içerisinde belirlenmiş olan görevleri yapan worker'ları barındıran bileşendir.

### **44.adım: -SSH nedir ?**

Secure Shell, uzaktaki bir bilgisayara, servera veya clouda güvenli bir şekilde bağlanmamızı sağlayan bir network protokolüdür. 22 portundan çalışır.

## **45.adım: -GitLab ve GitHub SSH-keygen oluşturarak bilgisayarına bağlayın?**

Sunum içerisindeki screenshotlarda mevcuttur.

## **46.adım: -Prometheus nedir, Jenkins nedir ?**

Prometheus, bir monitoring tooludur. Kubernetes, Jenkins gibi platformlar üzerindeki değişiklikleri izleyebilmemize veya anormal durum oluştuğunda bunun farkına varmamızı sağlar.

Jenkins, tıpkı GitLab CI/CD gibi bir CI/CD pipeline'ları oluşturmamızı, içerisindeki joblar ile devops otomatizasyonu yapmamıza olanak sağlayan bir araçtır.

## **47.adım: -Ansible nedir ?**

Ansible, sistem konfigürasyonu, kod deploy etme, ci/cd pipeline'larını orchestrate etme gibi işlemler yapabildiğimiz bir IT otomasyon tooludur. Ansible infrastructure as a code yapısı için örnek verilebilir.

## **48.adım: -Kubernate nedir ?**

Kubernetes, birden fazla container'ı orchestrate etme, yönetme, ölçekleme veya deploymentını otomatize etmemizi sağlayan bir platformdur. Docker swarm, bu teknolojiye bir alternatif olarak gösterilebilir.

## **49.adım: -Datadog Monitoring nedir ?**

Datadog, cloud tabanlı bir monitoring platformudur. Serverların, containerların, ya da diğer kaynakların, sağlığı, performansı kaynak tüketimi hakkında gözlemlene ve monitoring yapmamızı sağlamanın yanı sıra, bunları log olarak tutar ve geçmişe yönelik bilgi edinmemizi de sağlar.



## 50.adım: -Kanban nedir ?

Kanban, görsel ve grafiksel bir agile yönetim metodudur. İş akışını görselleştirerek, bu akıştaki darboğazları belirlemek, ve optimum hızda işi halletmek için kullanılır.

## 51.adım: -Devops nedir, Süreçleri nelerdir ?

Devops, development and operations, yazılım geliştiricilerin ve altyapı yöneticilerinin birbiriyle iletişimine ve beraber çalışması ilkesine dayanan, ve bu sayede, yapılan geliştirmelerin ve uygulamaların çok daha hızlı biçimde birbirine entegre olmasını sağlayan kültüre dair felsefeleri içerisinde barındıran metodların tümüne verilen genel addır.

Devops süreçleri temel olarak 3 aşamada incelenebilir.

*Otomasyon:* iş akışlarını, yeni yazılan kodların test edilmesini uygulamanın canlıya alınmasının otomatize edilmesi.

*İterasyon & Continious Improvement:* bu işlemlerin sürekli olarak düzenli bir biçimde tekrar etmesini ve her güncellemede tekrar etmesi işlemleri,

*Collabroration:* Yazılım geliştirme yapan ekiplerin birbiriyle entegre bir biçimde çok daha hızlı çalışmasını ve feedback alarak geliştirme yapılmasını sağlayarak proje üzerindeki verimliliği ve kaliteyi artırmaya yarayan yapı.

## Orchestration

Derste yaptığımız docker orchestration yapıyorsunuz. Docker Lab

*Sunum içerisindeki screenshotlarda mevcuttur.*

## Docker Bash Script

Linux'ta çalıştırmak üzere docker.sh yazıyorsunuz.

*Sunum içerisindeki screenshotlarda mevcuttur.*

## Docker-compose & SonarQube

Docker compose üzerinden; Postgresql,Sonarqubekuruyorsunuz.

Bir Java dosyasının maven komutu oluşturup(hazır kodlarda alabilirsiniz), sonarqube maven komutuyla çalıştırıp java kodlarındaki smell-code, güvenlik açıklarını kontrol ediniz.

*Sunum içerisindeki screenshotlarda mevcuttur.*