Kodluyoruz & Patika Airties Cloud-DevOps Bootcamp

Hafta 3 Ödevler

Mehmet Ulaş Akduman

+90 531 256 19 15

[akdumanmehmetulas@gmail.com](mailto:akdumanmehmetulas@gmail.com)

15.05.2022

**week-3-homework-1**

1. Bilgisayarda kaç port vardır?
2. 1024 altı olan portlar neden dış dünyaya açılmıyor? 1024 altı portlarin genel adı nedir? (well-known ports)
3. 80 portu dışarıya 8585 açılmış neden?
4. Nginx nedir?
5. Devops nedir? Devops yapısını araştırınız. (sonsuzluk işaretli diagramdaki başlıkların araştırılması)
6. Bilgisayarda 65.536 adedi TCP tipinde, 65.536 adedi de UDP tipinde olmak üzere, toplamda 131.072 adet port vardır. Bu portların birçoğu önceden bazı uygulamalar için ayrılmıştır, mesela 80 nolu TCP portu web uygulamaları için, 23 nolu TCP portu ise telnet uygulamaları için ayrılmıştır.
7. Priviliged portlar yani ayrıcalıklı portlar 1024'ün altındaki portlardır ve TCP/IP bağlantı noktası numaraları, normal kullanıcıların üzerlerinde sunucu çalıştırmasına izin verilmediğinden özeldir. Bu bir güvenlik özelliğidir, çünkü bu bağlantı noktalarından birindeki bir hizmete bağlanma durumunda bazı hacker veya zararlı yazılım tarafından sistem ihlali olayları yaşanabilir.
8. 1024 altında bulunan port yapıları güvenlik dolayısıyla direkt olarak kullanılamayacağı için 80 portu özel olarak 8585 portu ile dışarıya açılmaktadır.
9. Nginx Rus yazılım mühendisi Igor Sysoev tarafından geliştirilmiş mail sunucusudur. Nginx yüksek performanslı ve düşük hafıza kullanımına odaklanılarak tasarlanmış bir web sunucusudur.
10. Devops develop ve operations kavramlarının birleştirilmesiyle oluşturulmuş bir kültür, çalışma stili ve teknikler bütünlüğünü oluşturan terimdir. Üretilen yazılım projelerinin büyürken ve dağıtılırken yapılacak işlemlerin otomatize olmasını ve hızlı bir şekilde gerçekleştirilebilmesini sağlamaktadır. DevOps lifecycle içerisinde bulunan diagram başlıkları ve açılımları aşağıdaki gibidir:

* 1) Continuous Development – Sürekli Geliştirme

Bu faz planlama ve programın kodlanması süreçlerini içerir.

* 2) Continuous Integration – Sürekli Entegrasyon

Bu aşama DevOps süreçlerinin en önemli aşamalarından birisidir. Geliştiricilerin kaynak kodunda daha sık değişiklik yapmasını gerektiren bir yazılım geliştirme yöntemidir. Yapılan şey temel olarak yeni işlevselliği destekleyen kod, mevcut kodla sürekli olarak entegre edilmesidir. Bu nedenle bu süreçle birlikte sürekli yazılım geliştirme vardır. Güncellenen kodun, değişiklikleri son kullanıcılara yansıtabilmesi için sistemlerle sürekli ve sorunsuz bir şekilde entegre edilmesi gerekir. Bu aşamada Jenkins gibi araçlar kullanılır.

* 3) Continuous Testing – Sürekli Test

Bu aşama geliştirilen yazılımın hata testlerinin yapıldığı aşamadır. Sürekli test için TestNG, Selenium gibi otomasyon test araçları kullanılmaktadır. Bu aşamada, test ortamını taklit etmek için Docker’ın sunduğu container gibi teknolojiler kullanılır. Yapılacak olan testler otomatik bir halde yapıldığı için zaman ve emek tasarrufu sağlanmış olur. Yapılan testle sonucunda ise öngörülemeyen veya istenmeyen durumlar tespit edilmiş olur.

* 4) Continuous Monitoring – Sürekli İzleme

Bu süreçte DevOps işlemleri yapılırken operasyonel çıktıların elde edildiği, yazılımın kullanımıyla ilgili önemli bilgilerin kaydedildiği ve eğilimleri bulmak ve sorunlu alanları belirlemek için dikkatle işlendiği ve izlendiği bir aşamadır.

* 5) Continuous Feedback – Sürekli Geri Besleme

Uygulama geliştirme, yazılımın operasyonlarından elde edilen sonuçların analiz edilmesiyle sürekli olarak iyileştirilir. Bu işlemler mevcut yazılım uygulamasının bir sonraki sürümünün geliştirilmesi ve operasyonlar arasına sürekli geri bildirimin kritik olarak gerçekleşmesi ile mümkün olabilir. Süreklilik, bir yazılım uygulamasını geliştirme aşamasından çıkarmak, sorunlarını bulmak için kullanmak ve daha sonra daha iyi bir sürüm üretmek için gereken gereksiz adımları ortadan kaldırdığı için DevOps'ta temel faktördür.

* 6) Continuous Deployment – Sürekli Dağıtım

Bu aşamada kod, üretim sunucularına dağıtılır. Ayrıca, kodun tüm sunucularda doğru bir şekilde kullanıldığından ve çalıştığında emin olmak temel amaçtır. DevOps yaşam döngüsü yeni kod sürekli olarak dağıtılır ve yapılandırma yönetimi araçları, görevlerin sık ve hızlı bir şekilde yürütülmesinde önemli bir rol oynar. Container teknolojileri bu aşamada oldukça önem teşkil etmektedir. Bu araçlar, geliştirme, hazırlama, test etme ve üretim ortamında tutarlılık oluşturmaya yardımcı olur. Container araçları, uygulamanın test edildiği, geliştirildiği ve dağıtıldığı ortamlarda tutarlılığın korunmasına yardımcı olur. Böylelikle üretim ve dağıtım süreçlerinde hata ve arıza olasılığı azaltılmış olur.

* 7) Continuous Operations – Sürekli Operasyon

Bu aşama tüm DevOps operasyonları, yayın sürecinin tam otomasyonu ile sürekliliği temel alır ve kuruluşun toplam pazara sunma süresini sürekli olarak hızlandırmasına olanak tanır. Bu aşama kullanılarak herhangi bir yazılım ürününü daha verimli hale getirebilir ve ürününüzle ilgilenen müşterilerin toplam sayısı arttırılabilir.

**week-3-homework-2**

1. Statik kod analizi: Sonarqube
2. Kubernetes yapısı
3. Orchestration yapısı
4. npm nedir ve niçin kullanılır?
5. Agile-Scrum nedir?
6. SonarQube, geliştirilen projedeki kodların güvenlik, tekrar geliştirilebilirlik gibi unsurlar ile projeleri denetleyen bir yazılım aracıdır. Kodlarda hata olup olmadığına, güvenlik açığı olup olmadığına veya gereksiz kısımları olup olmadığına bakar. Projeye farklı açılardan puan vererek projenin testlerde başarılı olup olmadığını belirtir. Ayrıca hata olan veya hata olduğu düşünülen yerlerde sorunların nasıl çözülebileceği ile alakalı olası çözümler üretir.
7. Kubernetes, container orkestrasyonunu gerçekleştiren açık kaynaklı bir yazılımdır. Kubernetes tıpki Docker gibi kapsayıcılı uygulamaları yönetebilir, ölçeklendirebilir ve dağıtabilir bir teknolojidir. Kubernetes’in sağladığı yüksek düzeyde otomasyon, onu konteyner alanında farklı kılmaktadır. Container düzenleme için endüstri standardı haline gelen Kubernetes, ekiplerin bulutta yerel uygulamalara doğru ilerlerken mimarilerinde büyük değişikliler yapmalarına izin verir.
8. Orchestration yapısı, containerlerin dağıtımını, yönetimini, ölçeklenmesini ve ağ oluşturmasını otomatikleştirir. Oldukça fazla sayıda kişiye ulaşan ürünlerde sistemlerde geliştirme aşamalarında ve DevOps süreçlerinde oldukça fazla kullanılır. Orchestration yapısı ile birlkte aynı uygulamayı veya container yapısını yeniden tasarlamanıza gerek kalmadan farklı ortamlarda dağıtmamıza yardımcı olabilir.Containerlerdeki mikro hizmetler, depolama, ağ iletişimi gibi hizmetleri düzenlemeyi kolaylaştırır.
9. Npm 3. parti yazılımları yüklemeyi sağlayan bir araçtır. npm kendisiyle birlikte gelen Paket yöneticisidir ve bütün Node gelişmelerine yardımcı olur. Node JavaScript geliştiricileri tarafından araç paylaşmak, çeşitli modüller yüklemek ve bağımlılıklarını yönetmek için yaygın biçimde kullanılan bir teknolojidir.
10. Agile, proje yönetiminde, yazılım geliştirme süreçlerinde kullanılan ve karşılaşılan problemlerin seri ve tekrarlayan bir şekilde çözmek üzerine geliştirilmiş bir proje yönetim metodudur. Yazılımın sık sık teslim ve yenilendiği, değişikliği temel alan bir yazılım geliştirme modelidir.

Scrum yaklaşımı en popüler Agile yöntemlerinden birisidir. Büyük ve karışık yazılım süreçlerinin yönetilmesinde tercih edilen Scrum ile bütünü parçalamaya ve tekrara dayalı bir yöntem elde edilir. Scrum yönteminin en büyük faydalarından biri projenin açık ve net olmasından dolayı hem zaman kazandırılması hem de projenin başarılı sonuçlanmasını sağlamasıdır. Projenin anlaşılabilir ve yönetilebilir parçalara ayrılması olası sorunları hızlı bir şekilde tespit etmekte ve düzeltmekte zaman kazandırır. Yazılımı geliştiren takım üyelerinin birbirleri ile sık sık görüşme ve iletişim kurmasından dolayı ortaya çıkan sorunlarda hızlıca çözüm getirilebilir. Sık sık yapılan görüşmeler ve toplantılar sonucunda müşteri ile geliştiriciler arasında güven oluşur ve böylelikle projenin başarılı sonuçlanması beklenir.

**week-3-homework-3**

1. Ansible nedir?
2. XML nedir? Açılımı nedir?
3. JSON nedir?
4. YAML nedir?
5. http, https, status kodların araştırılması (1 ile başlayanlar, 2 ile başlayanlar...)
6. Ansible, platformların kendi aralarında bilgisayar desteği sağlayabilmesi için basit ve güçlü otomasyon desteği sağlayan yazılım aracıdır. Özellikle uygulama dağıtımı, sunuculardaki güncellemeler, yapılandırma yönetimi ve bir sistem yöneticisinin haftalık veya günlük olarak yaptığı hemen hemen her şey için kullanan bilgi teknolojileri uzmanlarına yönelik bir araçtır.
7. XML ‘’Extensible Markum Language’’ olan veya ‘’Genişletilebilir İşaretleme Dili’’ olarak geçen terimin temel amacı XML ile hazırlanmış belgelerin hem insanlar hem de işletim sistemleri tarafından okunabilmesi ve veri transferi yapabilmesidir. Bu nedenle neredeyse tüm cihaz ve işletim sistemleri tarafından okunabilir ve kullanılabilir.
8. JSON (JavaScript Object Notation), basit veri yapılarını temsil etmek için tasarlanmış bir veri değişim formatıdır. Esas olarak iki sistem arasındaki veri alışverişinde kullanılır. Örneğin JSON kullanarak sunucu ile web uygulaması arasında veri aktarılabilmesi mümkündür. JSON, veri yapısından oluşan veri nesnelerini iletmek için insanlar tarafından okunabilir metinler kullanır. Tarayıcı eklentileri ve web siteleri gibi JavaScript uygulamalarıyla kullanılabilir. Tüm tarayıcıları desteklemekte ve çok sayıda işletim sistemiyle uyumludur. Okuması, yazması, oluşturması ve yönetmesi kolaydır. API’ler ve web hizmetlerinde verileri biçimlendirmek ve aktarmak için kullanır. JSON’daki veri yapıları, anahtar / değer çiftlerine dayanır. JSON sözdizimi, verilerin kolay ayrıştırılmasını ve daha da hızlı uygulanmasını sağlar. Çoğu modern programlama dilinde kullanılabilir. Metin tabanlı yapısı nedeniyle Javascript, C, C ++, C #, Perl, Java, Python, Php gibi programlama dilleri ve platformları JSON’u destekler. JSON, çok karmaşık büyük verileri işlemek için uygun değildir. Zengin metin veya resimler gibi multimedya biçimlerini desteklememektedir.
9. YAML veya Ain’t Markup Language olarak geçen bu terim hem insan tarafından kolaylıkla okunabilen hem de sayısal olarak kolaylıkla işleme alınabilen ve tüm programlama dilleri tarafından kullanılabilen bir veri değişim formatıdır.Temel amacı oldukça kompleks bir yapı sunan XML’e alternatif olabilmek için YAML geliştirilmiştir. YAML kolay okunabilir, farklı programlama dilleri ile kullanılabilir, tutarlıdır, geliştirilebilir bir yapı sunar ve sistemlere entegre edilmesi kolaydır.
10. HTTP dönüş kodları http isteğinin sonucuyla alakalı bilgiler içeren kodlardır. Aşağıda görüldüğü gibi farklı farklı durumlara göre belirli aralıklarda durum kodları mevcuttur. Temelde 5 kategoride yer alan durum kodlar ise şöyledir:

* Bilgi vermek için kullanılan kodlar (100–199)
* İşlemin başarılı olduğu ile ilgili kullanılan kodlar (200–299)
* Yönlendirme kodları (300–399)
* Client tarafında gerçekleşen hataları bildiren kodlar (400–499)
* Client tarafında gerçekleşen hataları bildiren kodlar (500–599)

**week-3-homework-4**

1. URL, URI
2. DNS yapısı, Domain nedir? Host yapısı
3. GNU nedir?
4. Git
5. CLİ, GUİ ve farkı
6. URI Uniform Resource Identifier’in kısaltılmış halidir. İnternet üzerinde bir kaynağın tam yerine işaret eden standart formata uygun bir karakter dizisidir. Kısaca bir URL’nin altında bulunan kaynağın tam yoluna işaret etmektedir. Örneğin https://www.google.com/intl/tr/gmail/about/ bir URI’dir.

URL Uniform Resource Locator’ın kısaltılmış halidir. Basit tanımlı olan URL, internet üzerinde kaynağın yerine işaret eden standart bir formata uygun karakter dizisidir. Örneğin https://www.google.com/ bir URL’dir.

1. **DNS**: DNS**,** Domain Name System teriminin kısaltmasıdır. Alan isimlendirme sistemi olarak Türkçe’ye çevrilebilir. DNS, isimlerin IP adresine dönüştürülesi maksadıyla kullanılan ve 256 karaktere kadar da büyüyebilen bir sistemdir. DNS, bilgisayar ya da bilgisayar sistemleri ile çalışan cihazların IP’lerini çözerek birbirleri ile haberleşmelerini sağlar. DNS sistemi sunucu ve çözümleyici isimli iki faktör tarafından ortaklaşa çalıştırılır. Sunucular, IP adresi bilgilerini muhafaza etmektedir Çözümleyicilerde DNS sunucuları ve sunucuların adresleri bulunmaktadır. Bir çözümleyici, bilgisayarın adına karşılık gelen bir IP adresini bulmak istediğinde sunucuya yönlenir, internet adresleri öncelikle ülkelere göre sınıflandırılır. Örneğin Amerika’da kullanılan USA, bu sınıflandırmayı ifade eder. Her ülkenin kendine özel bir sınıflandırma sembolü vardır. Sınıflandırılan internet siteleri daha sonra "com", "gov", "edu" gibi alt bölümlere ayrılır. Bu ifadelerin anlamlarını şu şekilde sıralayabiliriz:

* com: İngilizce "commercial" kelimesinin kısaltmasıdır. Ticari kuruluş olan İnternet sitelerini gösterir.
* org: İngilizce "organization" kelimesinin kısaltmasıdır. Ticari niteliğe sahip olmayan kurumları temsil eder.
* mil: İngilizce "military" kelimesinin kısaltmasıdır. Askeri kurum sitelerini gösterir.
* gov: İngilizce "government" kelimesinin kısaltılmasıdır. Devlet'e bağlı olarak faaliyet gösteren İnternet sitelerinde kullanılır.
* edu: İngilizce "education" kelimesinin kısaltmasıdır. Eğitim kuruluşlarını ifade eder.
* net: "Network" kelimesinin kısaltmasıdır. Uluslararası faaliyet alanına sahip kurumları ifade eder. Alan adları, belirlenen kurallar çerçevesinde verilir. Herkes istediği alan adını kullanamaz.

**Domain:** Web üzerinde yayın yapan siteleri ziyaret etmek için Google Chrome, Opera, Internet Explorer gibi tarayıcılar kullanılır. Herhangi bir siteyi ziyaret etmek için bu tarayıcıların “adres çubuğu” kısmına yazdığımız, “http” ya da “https” ile başlayan ve .com, .net gibi bir uzantıyla biten site adresine ise “domain” veya dilimizdeki ifadesi ile “alan adı” denmektedir. Bu basit anlatımdan sonra teknik olarak da detaylandırmaya başlayalım. Internet üzerinde yayın yapan her sitenin rakamlardan oluşan benzersiz bir adresi bulunmaktadır. Bu benzersiz adres aynı zamanda alan adını yani domain adresi de nitelemektedir. Yani herhangi bir web sitesini ziyaret etmek ya da yayınlamak için, aslında domain adresine ihtiyaç yoktur. Ama günde yüzlerce siteyi ziyaret ettiğinizi ve bu sitelerden birçoğuna da arama motorlarını kullanmadan, tarayıcıya adres yazarak ziyaret gerçekleştirdiğinizi düşünürsek; her adrese ait sayı grubunu aklınızda tutmanız imkansızdır. Bu sebeple DNS burada kullanılır. En basit şekilde ifade etmek gerekirse bu teknoloji; ziyaret edilmek istenilen sitenin IP adresini, domain adreslerine çevirir.

**Host**: İnternet siteleri de tıpkı gerçek hayattaki siteler gibi belirli bir alan üzerine kurulurlar. Bir internet sitesini yayınlamak için gerekli olan işleme Host adı verilir. Türkçe’ye “barındırma alanı” olarak çevrilebilir. Hosting; web sayfanıza ait tüm dosyaların, kesintisiz bir şekilde bilgisayarda saklanması ve erişime açık tutulması hizmetidir. Bu bilgisayarlara ise “server” veya “web sunucusu” adı verilir. Web sunucuları, evinizde ya da ofisinizde bulunan bilgisayarlardan donanımsal olarak daha özellikli makinalardır ve host işlemlerinin gerçekleştirilmesinde kullanılırlar.

1. GNU özgür yazılım prensibine sahip olan bir işletim sistemidir. Kullanıcıların özgürlüğüne saygı duyar ve geliştirmeler bağımsız olarak kişisel tarafından yapılabilir. GNU işletim sistemi GNU paketlerinden ve üçüncü taraflarca yayımlanan özgür yazılımdan oluşur.
2. Git yazılım projelerinde kullanılan bir versiyon kontrol sistemidir. Git ile farklı kişilerin aynı anda aynı projenin farklı kısımlarında çalışıp projeyi ilerletmesi mümkündür. Git sayesinde projeler bozulduğunda, hata alındığında veya geliştirme aşamasında sorun yaşandığında hata alınan kayda gidilip en güncel çalışır proje durumu elde edilebilir. DevOps sürecinde bulunan araçlar ve teknolojiler ile uyumlu olup yazılım projesi geliştirmenin temel yapıtaşlarından birisidir.
3. Yazılımın ve donanımın ilerlemesi ile birlikte kullanılan cihazların işletim sistemlerine sahip olması gerekmektedir. İşletim sisteminin bilgisayar donanımı ile doğrudan etkileşime giren kısmı, kernel(çekirdek) olarak bilinir. Uygulamalarla ve kullanıcıyla etkileşime giren bölüme ise shell(kabuk) denir. Kullanıcı komut satırı arayüzünü (CLI) veya grafik kullanıcı arayüzünü (GUI) kullanarak shell ile etkileşime girebilir. GUI ve CLI arasındaki temel kullanım farkı GUI tarafında kullanıcının pencereler, simgeler, menüler gibi grafiksel öğelerin sistemle etkileşime girmesine izin verirken, CLI tarafında kullanıcının komutları kullanarak sistemle etkileşime girmesini sağlar.

**week-3-homework-5**

1. Kernel ile Shell farkı
2. Docker portainer
3. Cross origin
4. Application server: Tomcat, Wildfly, Nginx
5. CVCS – DVCS
6. Kernel ve Shell arasındaki farklar aşağıda sıralanmıştır.

* Kernel, bilgisayarın işletim sisteminin çekirdeği olarak işlev gören ve sistemdeki her şey üzerinde denetime sahip olan bir bilgisayar programıdır. Shell, işletim sistemi tarafından sağlanan hizmetlere erişmek için arayüz olarak çalışan bir bilgisayar programıdır.
* Kernel, sistemin tüm görevlerini kontrol eden sistemin özüdür. Shell ise kernel ve kullanıcı arasındaki arayüzdür. Kernelin farklı türleri olmamasına karşın, Shell yapısı birden çok türe sahip olabilir.
* Kernel ve Shell arasındaki fark, çekirdeğin, sistemin tüm görevlerini kontrol eden işletim sisteminin çekirdeği olmasıdır. Shell ise kullanıcının çekirdeği ile iletişim kurmasını sağlayan arayüzdür.

1. Docker portainer, Docker Cluster yapısının yönetilmesini sağlayan bir management UI’dır. Docker’ı terminalden kullanmak görsel olarak bazı şeyleri kontrol edememeyi veya düşünememeye neden olabilir. Docker bir arayüz ile kullanmak istenildiğinde portainer kullanılır.
2. Cross origin, bir web sayfası veya web uygulaması üzerinden sunulan kaynaklara (medya, css ve js dosyaları, fontlar, vb.) kaynağın sunulduğu alan adının (hostname) dışından gerçekleştirilen isteklerin yönetimini sağlayan bir işlemdir.
3. Tomcat

Apache Tomcat veya Tomcat Java tabanlı web uygulamalarını yayınlamak için kullanılan web sunucusudur.

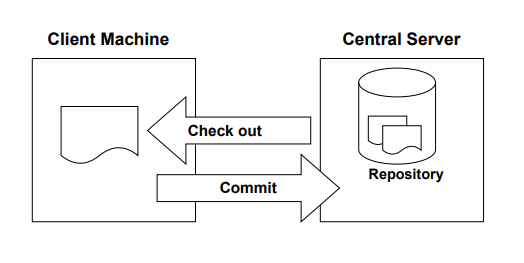
Wildfly

Java’da Sunucular temel olarak 2’ye ayrılmaktadır. Bunlar container özellik gösteren ve uygulama sunucusu özelliği gösteren sunuculardır. WildFly Red Hat tarafından geliştirilen JBoss’un yazdığı bir uygulama sunucusudur. WildFly Java ile yazılmıştır.

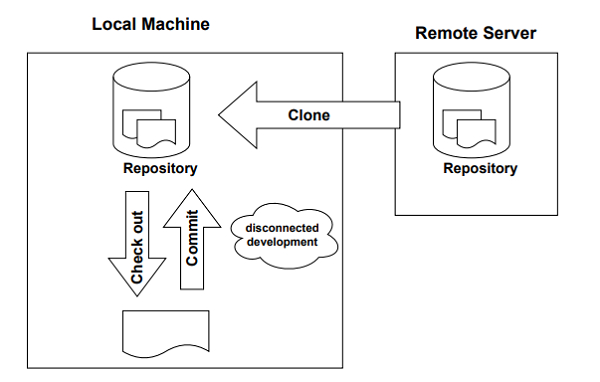
Nginx

Nginx Rus yazılım mühendisi Igor Sysoev tarafından geliştirilmiş mail sunucusudur. Nginx yüksek performanslı ve düşük hafıza kullanımına odaklanılarak tasarlanmış bir web sunucusudur.

1. CVCS (Centralised Version Control System) ve DVCS(Distributed Version Control System) farklı başlıklar ve özellikler altında toplanmış versiyon kontrol sistemleridir. Versiyon kontrol sistemi ile birlikte bir proje üzerinde farklı kişiler, gruplar ve organizasyonlar çalışabilir ve ürün ortaya çıkartabilir. CVCS tipi versiyon yönetim sistemlerinde bir sunucu-istemci modeli vardır. Sunucu, belirli bir süre boyunca kod tabanında yapılan değişikliklerin geçmişiyle birlikte projenin tek bir merkezi kopyasını tutar. Merkezi bir sürüm kontrol sistemi ile çalışmanın temel iş akışı aşağıda verilmiştir.



DVCS tipi versiyon yönetim sistemleri ise CVCS sistemler aksine daha eşitlikçi ve yenilikçi bir yaklaşım izler. Yapılacak şeyler CVCS sistemlerindeki gibidir, değişiklik yapar ve tekrar kontrol edersiniz. İki tür arasındaki temel fark ise merkezileştirilmiş sürüm kontrol sistemlerindeki gibi değişiklik geçmişini merkezi bir sunucuda tutması yerine ortak bir sunucuda tutup, farklı farklı kişiler tarafından çalışmanın en son sürümünün kullanılabilmesidir. DVCS tipi sistemlerin çalışma mantığını gösteren şekil ise aşağıda belirtilmiştir.



|  |  |
| --- | --- |
| **CVCS** | **DVCS** |
| Merkezi bir sunucuya sahip. | Her kullanıcı kendi içinde ve ortak olarak kullanılabilir sunuculara sahip. |
| Sunuya network vasıtasıyla bağlanılabilir. | Bütün kullanıcılar offline olarak çalışabilir fakat yapılan çalışmaların paylaşılabilmesi için network gerekir. |
| Subversion, Perforce Revision control system örnek olarak kullanılan sistemlerdir. | Git, Mercurial, BitKeeper örnek olarak kullanılan sistemlerdir |

**week-3-homework-6**

1. İyi bir commit mesajı nasıl olmalıdır?
2. FIFO ile LIFO sıralama şeklini araştırınız.
3. git merge rebase farkı, rebase yapısı nedir?
4. Sourcetree'yi inceleyiniz.
5. Interpreter-Complier nedir? Aralarındaki farklar nelerdir?
6. Sp engine
7. Servlet
8. İyi bir commit mesajı kolayca okunabilmesi ve anlaşılabilmelidir. Aşırı ayrıntı ve gereksiz konulara değinmemeli, ilgili konuyu kısa ve net bir şekilde aktarmalıdır. Ayrıca diğer takım üyeleri veya kullanıcılar için ileriye dönük anlaşılabilir bir yapıya sahip olmalıdır.
9. FIFO(first in, fist out) ve LIFO(last in first out) farklı işlemlerde kullanılan sıralama şekilleridir. FIFO’da sıraya ilk giren kişi ilk olarak çıkar. FIFO’ya örnek olarak banka kuyruğu örnek verilebilir; bankada sıraya ilk giren kişi ilk işleme alınacaktır. LIFO ise sıraya en son girenin ilk işlem göreceği sıralama şeklidir. Bunu bavula yerleştirilen elbiselerin alınma sırası gibi düşünebiliriz; bavula en son konan elbise ilk kullanılacak elbise olacaktır. En son kullanılanacak elbise ise bavula ilk başta koyulan elbisedir. LIFO ve FIFO farklı problemlerin ve algoritmaların çözümünde kolaylık ve işlem rahatlığı sağlamaktadır.
10. merge ve rebase arasındaki farkın anlaşılabilmesi için bu terimlerin ne olduğuna ve ne iş yaptığına bakmak gerekmektedir. Git merge ile bir branchte yapılmış çalışmalar başka bir branch ile birleştirilir. Örneğin UI tarafı ve back-end tarafında çalışılan iki branch merge yapılır ve back-end’i çalışan bir UI elde edilmiş olur. Git rebase git merge ile aynı sorunu çözmektedir. Bu komutların her ikisi de değişiklikleri bir daldan başka bir dalla bütünleştirmek için tasarlanmıştır; sadece bunu çok farklı şekillerde yaparlar. Tek bir özellik geliştirildiğinde ve bu özellik master’a getirilmek istendiğinde merge, geliştirmeye başlandıktan sonra başka bir geliştirici çalışmada alakasız bir geliştirme yaptıysa rebase kullanılır.
11. Sourcetree, sadece kodlamaya konsantre olunabilmesi için Git depolarıyla etkileşim kurma şeklini basitleştiren grafik kullanıcı arabirimidir. Bu arabirimle depoların görselleştirilmesi ve yönetilmesi kolaylaştırır. Ayrıca verimli ve tutarlı bir geliştirme süreci sağlamak için Mercurial ile entegre olunabilir.
12. Interpreter

Yorumlayıcı veya interpreter, birçok üst düzey programlama dilinden birinin kullanılarak yazılan ve talimatlarını doğrudan yürütmek için kullanan bir bilgisayar programıdır. Yorumlayıcı, üst düzeyde bulunan programı, daha sonra yürüttüğü bir ara dile dönüştürür veya üst düzey kaynak kodunu ayrıştırabilir. Bu işlemlerin ardından satır satır veya ifadeden ifadeye yapılan komutları doğrudan gerçekleştirir.

Compiler

Derleyici veya compiler, bir programlama dilinin kaynak kodunu makine koduna, bayt koduna veya başka bir programlama diline çeviren özel bir programdır. Kaynak kodu genellikle Java veya C++ gibi yüksek düzeyde, insan tarafından okunabilir bir dilde yazılır. Bir programcı, kaynak kodunu bir kod düzenleyicide veya bir düzenleyici içeren tümleşik geliştirme ortamında (IDE) yazar ve kaynak kodunu bir veya daha fazla metin dosyasına kaydeder. Kaynak programlama dilini destekleyen bir derleyici dosyaları okur, kodu analiz eder ve onu hedef platforma uygun bir formata çevirir.

Interpreter ve compiler arasındaki farklar ise şöyledir :

* Interpreter, bir seferde programın yalnızca bir ifadesini makine koduna çevirir.

Derleyici tüm programı tarar ve tamamını bir kerede makine koduna çevirir.

* Bir yorumlayıcının kaynak kodunu analiz etmesi çok daha az zaman alır. Ancak, süreci yürütmek için toplam süre çok daha yavaştır.

Bir derleyicinin kaynak kodunu analiz etmesi çok zaman alır. Ancak, süreci yürütmek için geçen toplam süre çok daha hızlıdır.

* Bir yorumlayıcı bir aracı kod oluşturmaz. Bu nedenle, bir tercüman hafızası açısından oldukça verimlidir. Bir derleyici her zaman bir aracı nesne kodu üretir. Daha fazla bağlantıya ihtiyaç duyacaktır. Bu nedenle daha fazla belleğe ihtiyaç vardır. İlk hata ile karşılaşılıncaya kadar programı sürekli çevirmeye devam eder. Herhangi bir hata tespit edilirse çalışmayı durdurur ve dolayısıyla hata ayıklama kolaylaşır.
* Bir derleyici, yalnızca programın tamamını taradıktan sonra hata mesajını oluşturur ve bu nedenle, bir derleyici ile çalışırken hata ayıklama nispeten daha zordur. Tercümanlar, örneğin Ruby ve Python gibi programlama dilleri tarafından kullanılır. Derleyiciler, örneğin C ve C++ gibi programlama dilleri tarafından kullanılır.

1. JSP’ler temel olarak servletlerdir, script seklinde html in içine yazılıp sonra jsp engine tarafindan servlet ile otomatik olarak compile edilirler, daha sonraki aşamalar servlet engine tarafından yürütülür.
2. Servlet Java EE içerisinde yer alan sunucu (server) – istemci (client) tabanlı uygulamaların haberleşmesini yönetmek için kullanılan sınıf, arayüz ve paket topluluğudur. Servlet sunucuya bir istek geldiğinde sırayla çalışacak metotları belirtir. Bu metotlar Servlet yaşam döngüsü olarak adlandırılır. Servlet yaşam döndüğünse Servlet sunucusuna istek geldiğinde **javax.servlet.Servlet** arayüzünde yer alan metotlar-işlemler çalıştırılır.

**week-3-homework-7**

Git adımları aşağıda verilmiştir.

1. adım ==> git unstaging staging kavramları nedir
2. adım ==> Git config ayarlarını bakalım user.name ve user.email
3. adım ==> git\_code adında linux komutlarıyla dizin oluşturalım
4. adım ==> git\_code dizine içinde data.txt dosyası oluşturalım
5. adım ==> git init yapalım
6. adım ==> linux ve git adında 2 tane branch oluşturalım
7. adım ==> linux branchine linux.txt dosyasının içine aklıma gelen linux komutları yazar her 2-3 satır sonunda add ve commit yapalım
8. adım ==> git branchine git.txt dosyasının içine aklıma gelen linux komutları yazar her 2-3 satır sonunda add ve commit yapalım
9. adım ==> GitHub için SSH-key oluşturarak GitHub pus yapalım
10. adım ==> git log --oneline --decorate alias olarak bir isim verelim
11. adım ==> .gitignore dosyası oluşturalım. ve git tarafından takip edilmesini istemediğimiz secret adındaki dizini git ignore verelim
12. adım ==> Bizim belirlediğimiz iki commit arasındakıi farka bakalım?
13. adım ==> GitHub bir proje clone yapalım
14. adım ==> git log,git statusu sürekli kullanalım
15. adım ==> Tag versiyonalamada kullanalım
16. adım ==> GitHub branch açıp , pull request yapıp kendi localimizde pull yapalım
17. adım ==> en son durumda linux fast forward merge ile git branchini no-ff merge ile yapalım
18. adım ==> branchlerimizi listelemek ,silmek v.b branch işlemleri yapalım
19. adım ==> conflict nedir? çözümü
20. adım ==> merge ile rebase arasındaki fark nedir?
21. Staging
22. Config ayarları ile kullanıcıya ait kullanıcı adı, şifre, mail gibi ayarlamalar yapılır. Kullanımı ise örnek olarak şöyledir:

git config –global user.name “mehmet”

git config –global user.email [mehmet@gmail.com](mailto:mehmet@gmail.com)

1. “mkdir git\_code” komutuyla dizin oluşturulur.
2. oluşturulan klasörün içine girdikten sonra touch parametresi ile txt dosyası oluşturulur.

cd git\_code

touch data.txt

1. Klasörün içindeyken “git init” komutu git dizinin oluşturur.
2. “git branch -b linux” ve “git branch -b git” komutları ile branchler oluşturulur.
3. “git add .” ve ‘git commit -m “mesaj”’ komutlarıyla commit edilir.
4. “git add .” ve ‘git commit -m “mesaj”’ komutlarıyla commit edilir.
5. SSH derste gösterildiği gibi oluşturulmuştur.
6. Bu adımdan sonraki işlemler git ile ilgili sorun yaşadığım için çözemedim.