


JENKİNS

Jenkins, Open Source bir Continuous Integration (CI) Tooludur.

Continuous Integration bir ürün geliştirme yöntemidir. Yazılım projelerinin sürekli olarak build ve test edilmesini, yazılımcıların projedeki değişiklikleri kolayca entegre edebilmesini ve kullanıcılara yeni buildler ulaşabilmesini sağlar. Böylelikle ortaya çıkan ürün sürekli olarak çokça test edilmiş, hatası azaltılmış ve güçlü bir şekilde kullanıcıya teslim edilebilir.



Sürekli entegrasyon aracı; yaptığı iş kısaca uygulamanızın kaynak kodlarını alıp, sizin istediğiniz süreçlerden geçirip, inşa etmesidir. Örneğin, kodları alır + derler + testlerinizi çalıştırır + ... + ... Tüm zorlu işleri gerçekleştirir.


Aşamaları

- 1.Integrated – Kodlar birleştirilir. Son değişiklikler projede toplanır.
- 2.Built – Kodlar derlenir çalıştırılabilir bir paket haline gelir.
- 3.Tested – Belirlenen testler koşulur.
- 4.Archieved – Versionlanan paket oluşturulur. İstenirse dağıtılabilir.



<https://jenkins.io/download> adresine gidiyoruz ve war dosyasını indiriyorsunuz.


- Ardından indirdiğiniz war dosyasını ister `java -jar jenkins.war` olarak çalıştırabilir isterseniz bilgisayarınıza kurduğunuz Tomcat altına atarak çalıştırabilirsiniz.
- Jenkins'i indirdiğimiz dizine gidiyoruz. Burada konsolu açıyoruz ve **java -jar jenkins.war** yazarak çalıştırıyoruz.



Eğer konsolda Jenkins is fully up and running yazısını görüyorsanız Jenkins'i kurmuşsunuz demektir.

İlk kurulumda consolda size bir şifre vericektir.Onu kullanarak jenkins'e girebilirsiniz.Eğer konsolda bulamazsanız jenkinsin workspace inde secret diye klasör olucak bunun içindeki txt dosyalarında şifrenizi bulabilirsiniz.Ardından tercih ettiğiniz browser a giderek localhost:8080/ yazdığınızda sizden bir şifre isteyecektir.

java -jar jenkins.jar - - httpPort=9090 diyebilirim hangi portta çalışması gerektiğine ben karar verebilirim.



Jenkinsin dili browsera bağlıdır eğer değiştirmek istiyorsanız locale plugin indirebilirsiniz. Ve configure settingsten browser language ignore et demeniz gereklidir.

Jenkins Ayarları

Jenkins plugin(eklenti) mantığıyla çalışır. Örneğin bir git i aktiveştirmeniz için Git Plugin kurmanız gerekir.Sol taraftaki menüden Manage Jenkins tabına tıklıyoruz.Ardından Manage plugin.Burada ikinci sekmedeki Available alanına geliyoruz ve Git Plugin yazıyoruz.




Kullanıcı ekleme kısmı kullanıcıları yönet ve solda kullanıcı oluştur sekmesi bulunur.

Ayarlardan manage extensions eklentileri yönet tıklayıp


Role-based Authorization Strategy

Role -based strateji değişik görevleri olan insanlara projeleri gösterebilir izinler verebilirsiniz.Bir çalışanın tüm projeleri görmesini istemediğimiz zaman kullanabiliriz.



Configure Global Security dan role base seçmemiz lazım bunu yaparsak
adminden baska kimse login olamaz.bunu yaptıktan sonra **MANAGE ASSIGN
ROLE AYARI CIKAR AYARLARDA**

Roller atadıktan sonra rollere göre projeleri görebilir mesele developer dev
projelerini gorur tester tes projelerini görur admin hepsini görebilir.




Projeyi seçerken github linki ile proje oluşturabiliyoruz. Ve trigger da poll scb seciyoruz
schedule kısmına * * * * * yazılır yani eger degisim varsa her dakika tetikler. So five stars
basically means

“every minute of every hour of every day.” You can also provide ranges of values: “* 9-17
* * *” would


mean “every minute of every day, between 9am and 5pm.” You can also space out the
schedule using

intervals: “*/5 * * * *” means “every 5 minutes




Build trigger-> build after other project are built altına proje ismi yazılır.

Projeleri baglarsın 1 -> 2 ->3 şeklinde çalışır.



Delivery pipeline = > Eğer bu seçenek yoksa pluginden ekleyebilirsiniz
initial compenente proje 1 yazın ve grafikleri çıkartır.projelerin üstünde +
işaretinden bunu yapabiliyoruz.


proje 1 build-> proje2 test -> proje3 deploy



Editable mail notification plugin editlenebilen bir mail taslağıdır.Default ayarlarını configure system dan değiştirebilirsiniz.


JUNIT

Junit test oluşturmak için test klasörüne gelip new - > junit Test case oluşturup junit test sınıfınızı yazabilirsiniz. Test fonksiyonuzun önüne **@test** yazmanız gereklidir.bunu çalıştırmak için run as junit test diyip ekrandan sonucu görebilirsiniz.İsterseniz buna runner sınıfı yazıp sonucu consola bastırabilirsiniz.Örneği projenin içinde runner sınıfı vardır.Ama sizin binlerce test sınıfınız olabilir bunlara tek tek runner sınıfı yazacak mıyım hayır tabiki bu testleri bir sınıfın içinde toplamak ve sadece onu test etmek isterseniz junit test suite oluşturmanız gerekir.



Test klasörüne gelip new -> other -> junit test suite oluşturabilirsiniz. Junit otomatik olarak test sınıflarınızı bulup bunları dahil eder sizin bir müdahale etmenize gerek yoktur. Bu test suite classına runner yazıp consoldan görebilirsiniz. Eğer istersenizde ant ile bu test edip raporlarınızı görebilirsiniz. Eğer cmde üzerinden çalıştırmak isterseniz

javac Deneme.java TestJUnit.java TestRunner.java -> java TestRunner



Antta bu juniti testimin raporunu görmek istersem anta yazabilirim.projenin build.xml kısmını inceleyebilirsiniz.Genelde ant dosyasından test suitlerin raporunu istersek bütün test sonuçlarını görebiliriz.tek tek test dosyalarının sonucu tek tek bakmak yerine sadece suiti kotntrol etmek daha kolaydır.



Bu test sınıflarımızı belli test kalıp sınıflarıyla inherit etme şansımız var.

extends TestCase

extends TestResult

implement Test


Bu sınıfların özel fonksiyonları var junit.pdf içerisinde detaylı bilgiyi bulabilirsiniz.

Bazı test fonsiyonlarını ignore etmek için o fonk. Yada classın başına **@ignore** yazmamız yeterlidir.

Bazı testlerin ne kadar süre çalışmasını gerektiğini yazabiliriz **@test(timeout=1000)**

Bazı test classlarının aritmetik hatalarda fail vermemesi için

@Test(expected=Aritmetic.Exception.class) ör. 1/0



Bu ant dosyasını run as ant build olarak çalıştırabiliriz.Eğer bu ant dosyasına parametre vermek istersek propert name olusturmak lazım eğer istersek buna default değer atayabiliriz. Bu ant dosyasını çalıştırırken cmd ye **ant -Dversion=1.1** örneğindeki gibi çalıştırılabilir.Eğer jenkins üzerinden parametre vermek istersek proje ayarlarından **This project is parameterized** e tıklayıp istediğimiz tipte istediğimiz sayıda parametre tanımlayabiliriz.Ama bu ant dosyasındaki parametre ismiyle aynı olmak zorunda.



Örnek mail ayarı smtp.gmail.com yada ip adresi

suffix -> @ias.com.tr


advance -> use ssl -> smtp port 465-> utf-8 ->html

bu email template i için consoldaki outputu mail olarak görmek istiyorsak mail taslağına **<pre>\${BUILD_LOG, maxLines=9999, escapeHtml=false}</pre>** eklemeniz lazım.Burda html taglerini kullanmak mümkün eğer tagleri emailde görürseniz ayarlardaki html e tıklamanız gerekir eğer ondanda aynı sorun çıkarsa jenkins system ayarlarında safe htmlle tıklanınız gerekir.



Jenkins variableleri genellikle `${variablename}` olarak yada `%variablename%` olarak kullanılır.

Birden fazla ant targeti çağırmak istiyorsak arasına , ; koymamıza gerek yok sadece boşluk yeterlidir.

- 
- Ant dan juniti tettiklediğimiz zaman TEST-*.xml dosyaları oluşur bunun içinde bazı bilgisayar bilgileri ve test sonuçları vardır.Jenkins üzerinden bu xml dosyalarını göstererek test sonuçlarını jenkins üzerinden görebiliriz.Ya da ant üzerinden junit report oluşturmak için bu .xml dosyalarını gösterip raporları html şeklinde çıkartabiliriz.