

1. Jenkins : Jenkins, Open Source bir Continuous Integration (CI) Tooludur.
2. 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.
3. 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. Archived – 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.

Her seferinde bunu consolda yazmak yerine bat dosyasına yazabiliriz.

- Eğer consolda 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 consolda bulamazsanız jenkinsin workspace inde secret diye klasör olacak 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 eger degistirmek istiyorsan locale plug in indirebilirsiniz. Ve configure settingsten browser language i ignore et demeniz gereklidir.

Jenkins Ayarları

Jenkins plugin(eklenti) mantığıyla çalışır. Örneğin bir git i aktifleştirmeniz 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.

Localhost=127.0.0.1

Jenkinsin dizinini degistirmek için environment path editle jenkins home java home gibi

command line interfaces => localhost:8080/cli
burdan komutlara bak cmd üzerinden calıştır.
burdan url ile indir jenkins-cli.jar

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

Ayarlardan manage extensions eklentileri yönet ordan

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 secmemiz lazım bunu yaparsak adminden baska kimse login olamaz.bunu yaptktan sonra MANAGE ASSIGN ROLE AYARI CIKAR AYARLARDA

roller atadıktan sonra rollere gore projeleri gorebilir mesele developer dev projelerini gorur tester tes projelerini gorur admin hepsini gorur.

System mesajını duzenelerken html kodlarından yararlanabilirsin <h1> safaf </h1>

Eger bu calismazsa configure security tan safe html ayarını secebilirsiniz.

projeyi secerken github linki ile proje olusturabiliyoruz free sytle dan sonra github linki eklenecek yer yoksa github pluginini eklemek lazim. Ve trigger da poll scb seciyorz schedule kismina * * * * * yazilir 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,

github degilde windows cmd deyı secerek command ları oraya yazarak calıstırabiliriz.

Projeyi kaydettikten sonra build edebiliriz.

3 tane proje olusturup bunlar birbirlerini tetikleyebilir. Build -> deploy-> test

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

Projeleri baglarsın 1 -> 2 ->3 diye calısır.

Delivery pipeline = > Eğer bu seçenek yoksa plug inden eklersın initial compenente proje 1 yaz ve grafikleri cikartır.projelerin ustünde + ısaletinden yapıyoruz

proje 1 build-> proje2 test -> proje3 deploy

```
cd \Users\salih\eclipse-workspace\rmi-gui\dist\20170718
```

```
rmi-gui-server.jar
```

```
rmi-gui-client.jar
```

genelde free style proje acılır.

Eger farklı ortamlardda test etmek ıstersen multiconfiguration job olusturabilirsin.

Mailed report edebiliyor bunun ıcın yine plug in yuklemen lazim.editable mail notification plug in editlenebilen bir mail taslağıdır.Default ayarlarını configure system dan deęiştirebilirsin.

ALIRKEN:

git fetch

git merge origin/master

GÖNDERİRKEN:

git status yap

eklenecekleri ekle git add "sdf"

git commit -am "commit mesajı"

git push origin master -u

sarı kagıttaki notları yaz.

JUnit is a unit testing framework for Java programming language. This approach is like "test a little, code a little, test a little, code a little."

[c:/](#) de junit klasoru olusturup junit.jarı oraya attım JUNIT_HOME = [C:\JUnit](#)

Set the environment variable CLASSPATH to %CLASSPATH%;

%JUNIT_HOME%\junit4.10.jar;.

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 kontrol 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 test sınıfınıza birden fazla parametre ile test etmek mümkün.Detaylı örnek projedeki prime number checker classı ve testine bakabilirsiniz.

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 calıstırırken 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.

Eğer projemizim sonucunu mail almak istiyorsak editable email pluginini indirmemiz lazım ardından cconfigure systemden email ayarlarını yapmak gerekir.Bu emailin kim tarafınfan gönderildiği kimlere gönderileceği mailin template kısmını değiştirebiliriz.jenkinsin variablelarından kullanabiliriz.

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.

Eğer projenizdeki junit testleri ant üzerinden sonucu almak rapor almak istiyorsanız junit.jar dosyasını lib eklemeniz gerekecektir.

Release plug in indirin -> configure release build -> parametre ekle -> ve ardından nasıl bir release istiyorsan önce server çalıştır sonra bu antı çalıştır gibi şeyler ekleyebilirsiniz. Ve ardından build kısmında release menusu açılacak ve burdan version numrası next versiyon felan çıkar.

[Parameterized Trigger plugin](#)

[Version Number Plug-In](#)

[Locale plugin](#)

[JUnit Plugin](#)

[Email Extension Plugin](#)

[Email Extension Template Plugin](#)

[description setter plugin](#)

[build-name-setter](#)

ant plugin

git plugin

Jenkins variableları genellikle \${variablename} olarak yada %variablename% olarak kullanılır.

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