

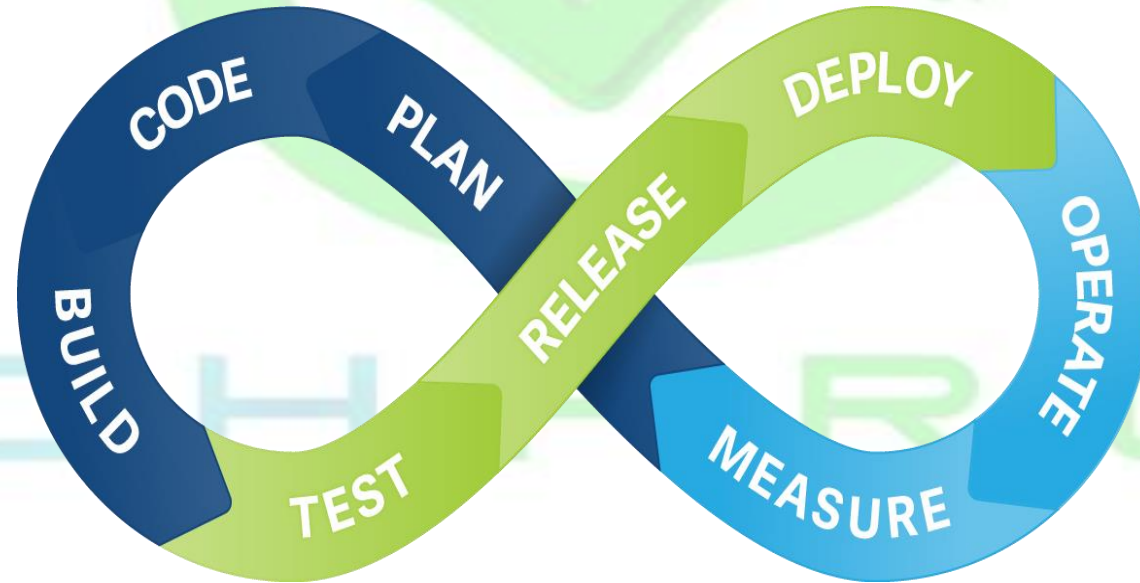
JENKINS

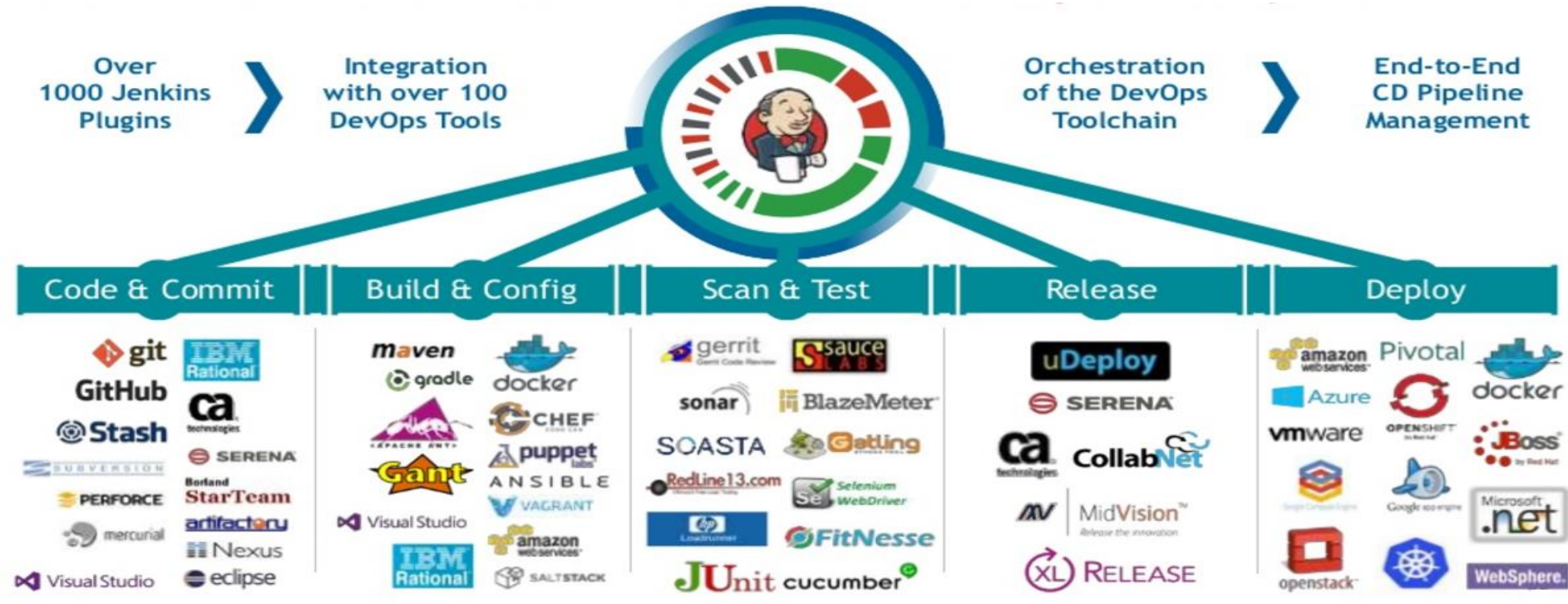


TECHPROED

Jenkins

Jenkins, Sürekli Entegrasyon (CI: Continuous Integration) ve sürekli dağıtım (CD: Continuous Delivery) yöntemi için kullanılan java ile yazılmış açık kaynak kodlu bir otomasyon sunucusudur. Yazılım geliştirme süreçlerini otomatize etmemize yarar. Jenkins, belirli bir sunucuda ve portta çalışır, belirlenen kaynaktan projeye ulaşır ve istenen işlemleri gerçekleştirir. Sonuçlarını belirlenen kişilere iletir. Bu sayede projemiz sürekli test edildiği için hatalar hızlıca tespit edilir ve çalışır bir vaziyette tutulur. Jenkins, manuel olarak yapılan build, test ve deploy gibi işlemleri uygular ve bu süreçte yaşanabilecek tüm aksaklık ve iletişim eksikliğini en aza indirir.





Ekibinizin olmazsa olmazlarından birisidir Jenkins.

- Jenkins, birçok topluluktan destek alan açık kaynak kodlu bir araç,
- Kurulumu basit ve anlaşılır,
- Birçok ihtiyaca yönelik 1000'den fazla eklentisi mevcut,
- Ayrıca mevcut olmayan Jenkins eklentilerini de oluşturmak kolay,
- Java'da yazılmış olması itibari ile neredeyse tüm platformlarda çalışabilir.

Jenkins Kavramları

Job: Bir jenkins projesidir. Otomatize etmek istediğimiz işleri burada belirleriz. Örneğin, job config üzerinden şu repository'i çek, şu şartlarda build et, şu testleri çalıştır ve belirlenen kişilere mail at gibi işlemleri burada belirleriz.

Node: job'un üzerinde çalıştığı sunucuyu ifade eder. Testleri başka bir bilgisayarda koşturmak istediğimizde node oluşturur ve bağlantı için gerekli şartları gerçekleştirdikten sonra node'da testlerimizi çalıştırabiliriz.

Plugin (eklenti): Jenkins saf haliyle yüklenir, ihtiyacımıza göre plugin yükler ve bunları kullanırız. Örneğin, Job çalıştıktan sonra mail atması için "Email Extension" eklentisini yüklemeli ve post-build adımıyla kullanmalıyız.

Pipeline: Sürekli teslim işlem hatlarının Jenkins'e uygulanmasını ve entegre edilmesini destekleyen bir eklentiler paketidir. Bir işin uygulanmasından sonra otomatik olarak diğer işlerin başlatılması sürecidir.

TECHPROED



Jenkins Plugins

TECHPROED

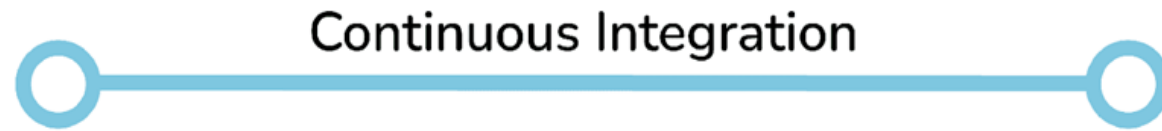
Continuous Integration(CI)

- Continuous Integration(CI):** CI kelime anlamı olarak sürekli entegrasyon anlamına gelmektedir. CI, kodun Git gibi versiyon kontrol sistemlerinden sürekli değişiklik olup olmadığını kontrol edip en küçük değişiklikleri bile entegre etme anlamına gelmektedir.
- CI'nin hedefi, uygulamayı Git reposundan alıp, paketleyip, test edilecek ortama kurulumunu otomatize etme sürecidir.
 - CI sürecinde kod bir takım testlerden geçer. Testler başarılı bir şekilde sonuçlanır ve tüm sistem çalışır haldeyse deploy işlemlerini başlatır. Eğer değişiklikler sistemde kırılmalara neden olmuşsa bu durumdan programcılarını haberdar eder.

TECHPROED

Continuous Deployment (CD)

- CD ise CI'dan başarılı bir şekilde geçen sistemin ilgili yerlere dağıtımı ile ilgilenir. Yani CI süresince kod derlendi, bazı testlerden geçti ve CD süreci ile dağıtımına başlandı.
- Şimdi Jenkins'in neler yaptığına bir daha bakalım. Jenkins, ilgili git reposunu belirli aralıklarla kontrol eder. Değişiklikleri kendi reposuna alır ve derler. Daha sonra sizin belirlediğiniz bir dizi testlerden geçer. Sonrasında istediğiniz yere bu yeni versiyonu deploy eder



Peki Jenkins'ten başka CI toolu var mı?

Aşağıda diğer CI toolarını da görebilirsiniz fakat şu an en popüler olanı Jenkins.

Jenkins

GitLab

Bamboo

CircleCI

Travis CI

BuildBot

Spinnaker

NeverCode

Wercker






TeamCity

Buddy

Semaphore

Codship

GoCD

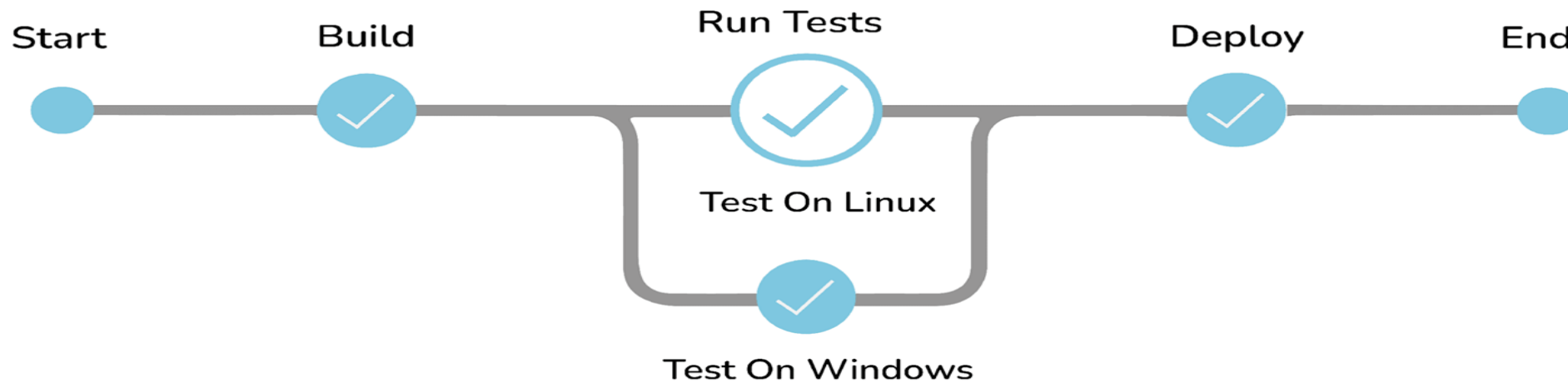
	 Jenkins	 circleci	 TeamCity	 Bamboo	 GitLab
Open source	Yes	No	No	No	No
Ease of use & setup	Medium	Medium	Medium	Medium	Medium
Built-in features	3/5	4/5	4/5	4/5	4/5
Integration	★ ★ ★ ★ ★	★ ★ ★	★ ★ ★ ★	★ ★ ★	★ ★ ★ ★
Hosting	On premise & Cloud	On premise & Cloud	On premise	On premise & Bitbucket as Cloud	On premise & Cloud
Free version	Yes	Yes	Yes	Yes	Yes
Build Agent License Pricing	Free	From \$39 per month	From \$299 one-off payment	From \$10 one-off payment	From \$4 per month per user
Supported OSs	Windows, Linux, macOS, Unix-like OS	Linux or MacOS	Windows, Linux, macOS, Solaris, FreeBSD and more	Windows, Linux, macOS, Solaris	Linux distributions: Ubuntu, Debian, CentOS, Oracle Linux

Pipeline nedir?

“Pipeline” terimi bir projenin ilk geliştirme aşamalarından halka yayınlanmasına kadar olan süreci tanımlamak için kullanılır.

Jenkins pipeline’ındaki üç temel aşama nedir?

Proje oluşturma (build), projeyi test etme (test) ve projeyi dağıtma (deploy)



Maven ile Jenkins arasındaki fark nedir ?

Maven bir build aracıdır. Build ve sürüm kontrolü süreçlerinde yardımcı olur. Jenkins ise sürekli entegrasyon sistemidir. Maven'ı build için kullanır.

Jenkins'de multibranch pipeline nedir?

Multibranch pipeline, aynı projenin farklı dalları için farklı Jenkinsfile'lar uygulamanızı sağlar. Jenkinsfile içeren dallar için pipeline'ları otomatik olarak keşfeder, yönetir ve yürütür.

Soru: Jenkinsfile nedir ?

Jenkinsfile bir jenkins pipeline tanımını içeren bir metin dosyasıdır.

```
pipeline {  
  agent any  
  stages {  
    stage('Build') {  
      steps {  
  
      }  
    }  
    stage('Test') {  
      steps {  
  
      }  
    }  
    stage('Deploy') {  
      Steps {  
  
      }  
    }  
  }  
}
```



TECHPRODEV


```
pipeline {
  environment {
    BUILD_SCRIPTS_GIT="http://10.100.100.10:7990/scm/~myname/mypipeline.git"
    ""

    BUILD_SCRIPTS='mypipeline'
    BUILD_HOME='/var/lib/jenkins/workspace'
    agent any
  }
  stages {
    stage('Checkout: Code') {steps {
      sh "mkdir -p $WORKSPACE/repo;\
      git config --global user.email 'email@address.com';\
      git config --global user.name 'myname';\
      git config --global push.default simple;\
      git clone $BUILD_SCRIPTS_GIT repo/$BUILD_SCRIPTS"
      sh "chmod -R +x $WORKSPACE/repo/$BUILD_SCRIPTS"}}
    stage('Yum: Updates') {steps {
      sh "sudo chmod +x
      $WORKSPACE/repo/$BUILD_SCRIPTS/scripts/update.sh"
      sh "sudo $WORKSPACE/repo/$BUILD_SCRIPTS/scripts/update.sh"}}}
  }
  post {
    always {
      cleanWs()}}}
```