

DevOps Nedir?

DevOps , Bilgi Teknolojileri departmanı içerisinde bulunan iki temel birimi (Developers and Operations) Geliştiriciler(Yazılım Geliştiriciler, Yazılım Testçileri, vb.), Operasyon (Sistem Mimari ve Altyapı Ekipleri,Güvenlik ve Ağ ekipleri vb.) bir arada etkili bir iletişim içerisinde beraber çalışmalarıdır.

DevOps'u aslında bir felsefe, yaklaşım veya bakış açısı olarak değerlendirebiliriz.Yazılım geliştiricilerin alışık olduğu Scrum, Agile, Kanban ve diğer yöntemler gibidir.

Dev takımı,

Oluşturulacak uygulamaya ait planları yapmak,

Uygulamayı oluşturmak(Kodlamak)

Uygulama release ve publish (versiyonlama ve yayınlama)

Uygulama iyileştirme (Update)

Uygulama Test süreçleri

.....

Sorumludur.

Ops takımı,

Oluşan uygulamaların barındırılacağı ve kullanılacağı ortamı tasarlamak,

Uygulamaların çalışması için gereken sistem bileşenleri ile iletişime geçebilmeleri için gerekli ağ ve güvenlik yapılandırmalarının yapılmasını sağlamak,

Uygulamanın kaynak kullanımını belirlemek,

Uygulamanın gerekli izleme (Monitoring) araçları ile takibini sağlamak,

Uygulamanın sistem kaynaklarını kullanım düzeyine göre kaynak arttırımını sağlamak (Scale Up ve Scale Down) ,

.....

Sorumludur.

DevSecOps

DevSecOps, güvenlik uygulamalarının DevOps süreçlerine dahil edilmesi kültürüdür. DevSecOps, DevOps Mühendisleri ve güvenlik ekipleri arasında esnek işbirliğine sahip bir güvenlik kültürü oluşturmayı amaçlar. DevOps'un kendisi gibi, karmaşık yazılım geliştirme süreçlerine çevik yöntemler uygulayarak yeni çözümler getirir. DevSecOps; eski güvenlik modellerinin, günümüzdeki sürekli dağıtım süreçleri üzerinde daha etkisiz kalmasından doğan bir kavramdır. DevSecOps, iki zıt görüş olan “hızlı teslimat” ve “ güvenli kod” görüşlerini karıştırarak basit bir çizgide birleştirir. Amaç; kodun hızlı ve güvenli bir şekilde teslim edilmesini sağlarken, yazılım ekipleri ile güvenlik ekipleri arasındaki geleneksel boşlukları doldurmaktır.

Dockerfile

Dockerfile, belli bir image görüntüsü oluşturmak için var olan tüm katmanların açıklandığı uzantısı olmayan dosyadır. Dockerfile içerisinde hangi Image'ın kullanılacağı, hangi dosyaları içereceği ve hangi uygulamanın hangi parametrelerle çalışacağı yazılır. Docker, Dockerfile dosyasında bulunan komutları sırayla tek tek çalıştırır. Her komut yeni bir katman oluşturur ve Docker build sonunda elimizde uygulaması ait Docker Image oluşur. Docker Image container oluşturarak uygulaması ayağa kaldırabiliriz. Bilinmesi gereken en önemli nokta ise dosya adının kesinlikle *Dockerfile* şeklinde olmasıdır. “dockerfile”, “DockerFile” gibi verilen isimlendirmeler yanlıştır.

Not: Dockerfile dosyasının herhangi bir uzantısı yoktur.

Not : Dockerfile base image ile başlar. Base Image işletim sistemi, docker image veya uygulamaya ait base image olabilir.

Dockerfile Avantajları

- Sunucu yapılandırmaya gerek yoktur.
- Uygulama bağımlılıklardan izole ediliriz.

Jenkins Nedir ?

Jenkins basit tanım ile, bir yazılım projesinde dinamik olarak gerekli olan yapısal işlemleri otomatize ederek projeyi hızlı, kolaylıkla hata raporlaması ve kolay test edilip hataların fixlenebilmesini sağlayan bir CI(Continuous Integration) aracıdır.

Peki nedir bu gereklilikler ?

Şöyle ki ; Projemizi bir versiyon kontrol sistemi(Git) üzerinde bulunduruyoruz. Yazılımın sunucu ortamında çalışması için gerekli olan toollar, eklentiler ve belirli config ayarları gerekiyor. Projenin, koşulması gereken Ui, Unit, Api vb testleri koşuluyor ve stabil olduğundan emin olunarak son kullanıcıya hizmet vermeye devam ediyor. Jenkins bize bu ve daha detaylı süreçler için her bir işlemi üstlenerek ve aralarında bir pipeline kurarak otomatize ediyor.

Aktif olarak hizmet veren ve devamlı geliştirme sağlanan projelerde , yazılımın ve kullanıcının gereksinimleri doğrultusunda devamlı teknik geliştirme söz konusudur. Her bir geliştirme ve deploy süreci kullanıcı nezdine ulaşmadan önce bir test sürecinden geçmesi gerekli ki ürünün stabilizasyonunun devam ettiğinden emin olunmalıdır. Eğer ki yapılan geliştirmeler sonucu derlemede bir sorun yaşanır veya testler pass etmez ise Jenkins'in konsol çıktısı bizi uyaracaktır. Bu gibi fail durumlarda, konfigüre edilmesi durumunda jenkins uyarı olarak mail vb. yollarla notification yolluyor.

Web Server

Web Server, web sayfalarını kullanıcıya sunmak için HTTP(Hypertext Transfer Protocol) kullanır.

HTTP iki programda uygulanır: Bir istemci programı ve bir sunucu programı. Farklı uç sistemlerde(host) çalışan istemci programı ve sunucu programı birbiriyle HTTP mesajlarının alışverişini yaparak konuşur. HTTP bu mesajların yapısını ve istemci-sunucunun mesaj alışverişini nasıl yapacağını tanımlar. Bir web sayfası nesnelerden oluşur. Bir nesne sadece bir tek URL ile adreslenebilen bir dosyadır(bir html dosyası, bir JPEG görüntüsü gibi).

Web nesnelerine ev sahipliği yapan popüler sunucular arasında Apache ve Microsoft Internet Information Server(IIS) örnek olarak verilebilir.

Not: HTTP protokolü 4. katmandaki(taşıma katmanı) TCP'yi kullanır. Çünkü TCP, HTTP'ye güvenilir veri transferi servisini sunar.

Application Server Nedir?

Sadece belli bir uygulamanın veya yazılımın çalıştırılması için kullanılan sunuculardır. Bu uygulamalar son kullanıcının erişerek işlem yaptığı uygulamalar olabileceği gibi son kullanıcıya kapalı, sadece diğer sunucuların farklı bilgilere erişmek için kullandığı sunucular da olabilir. Bir uygulama sunucusu, uygulama programının iş mantığını sağlayan, dağıtılmış ağda bilgisayardaki bir sunucu programdır. Uygulama sunucusu, genellikle bir grafik kullanıcı arabirimi (GUI) sunucusu, bir uygulama (iş mantığı) sunucusu ve bir veritabanı ve işlem sunucusundan oluşan üç katmanlı bir uygulamanın parçası olarak görülür.

Tomcat nedir?

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

Java, Java EE veya Java Teknolojileri içerisinde Java Servlet, JavaServer Pages, Java Expression Language, Java WebSocket gibi çeşitli teknolojiler yer alır.

Bu teknolojiler JCP (Java Community Process) olarak adlandırılan ve genellikle çeşitli firmalardaki geliştiriciler tarafından standart olarak belirlenir.

Standartlar belirlendikten sonra bu teknolojilerin kullanılabilmesi için bu standartların kodlara dökülmesi-implement gerekir.

Apache Tomcat bu standartları uygulayan ve içerisinde web sunucusu yer alan bir Java uygulamasıdır.

Netty Nedir?

Netty, yüksek performanslı, son derece güvenilir ağ sunucusu ve istemci programlarını hızla geliştirmek için eşzamansız, olay odaklı bir ağ uygulama çerçevesi ve araçları sağlar.

Başka bir deyişle, Netty, NIO tabanlı bir istemci ve sunucu tarafı programlama çerçevesidir.

Netty'yi kullanmak, belirli bir protokolü uygulayan bir istemci gibi bir ağ uygulamasını hızlı ve kolay bir şekilde geliştirmenizi sağlayabilir

3. Sunucu uygulaması. Netty, ağ uygulamalarının programlama geliştirme sürecini basitleştirmeye ve kolaylaştırmaya eşdeğerdir, örneğin: TCP ve UDP'ye dayalı soket hizmeti geliştirme.

Basitçe ifade etmek gerekirse Netty, yüksek eşzamanlılığı kaldırabilen bir ağ hizmeti mimarisidir. Netty ile kendi HTTP sunucunuzu, FTP sunucunuzu, UDP sunucunuzu, RPC sunucunuzu, WebSocket sunucunuzu, Redis Proxy sunucunuzu, MySQL Proxy sunucunuzu vb. Uygulayabilirsiniz.

Netty'nin Özellikleri

Netty'nin üç katmanlı ağ mimarisi şu şekilde tasarlanmıştır:

İlk katman: Reaktör iletişim programlama katmanı. Bu katmanın ana sorumluluğu, ağ katmanının verilerini bellek tamponuna okumaktan sorumlu olan ağ bağlantısını ve okuma ve yazma işlemlerini izlemek ve ardından bağlantı oluşturma, bağlantı aktivasyonu, okuma olayları gibi çeşitli ağ olaylarını tetiklemektir. olayları yazın, vb.

Bu olaylar Ardışık Düzen içinde tetiklenir ve ardından Ardışık Düzen'in sonraki işlemlerde olduğu gibi hareket ettiği sorumluluk kanalı.

İkinci katman: sorumluluk zincirinin boru hattı katmanı. Sorumlu olaylar,

sorumluluk zincirinde düzenli bir şekilde ileriye (geriye doğru) yayılır ve aynı zamanda sorumluluk zincirinin dinamik olarak düzenlenmesinden de sorumludur. Pipeline, önemseydiği olayları izlemeyi ve işlemeyi seçebilir.

Üçüncü katman: İş mantığı işleme katmanı, genellikle iki kategoriye ayrılabilir:

a. Günlük, sipariş işleme gibi saf iş mantığı işleme. b. HTTP (S) protokolü, FTP protokolü vb. Gibi uygulama katmanı protokol yönetimi.