# Azure DevOps ile Uygulama Derleme ve Dağıtma

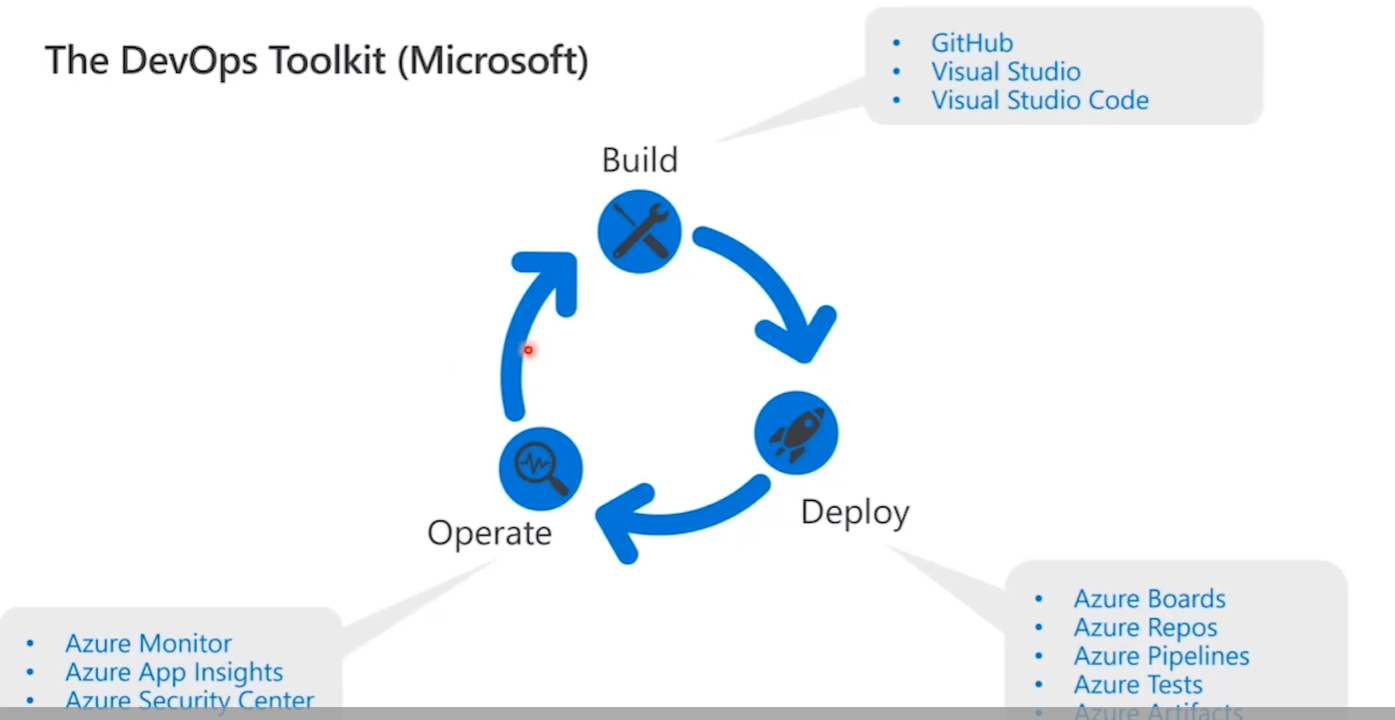
Mehmet Kut

### DevOps nedir?

“Devops is the union of the people, process ve products to enable continuous delivery of value to business and it end-users.../ insan, süreç ve ürünlerin sürekli şekilde son kullanıcıya bir değer üretmesine Devops denir” Donovan Brown, Microsoft

The cycle of build-develop-deploy

### Introducing Azure DevOps



#### Azure DevOps Tool kitleri

Azure Boards – agile tools to plan,track and discuss

Azure Pipelines – build, test and deploy with CI/CD that works with any lang. Platform and cloud. Connect to github or etc.

Azure Repos – cloud - hosted private git repos and collobrate to build better code with pull requests

Azure Test Plans – test and ship

Azure Artifacts – create, host, and share packages w/ your team, add artifacts to your CI/CD pipelines w/ a single click

### Benefits

Quick set/up

Maintenance-free operations

Easy collobration across domains

Elastic scale

Rock-solid security

Access to cloud-running

### Azure Pipeline

Host any lang, platform, any cloud

The deployment not for only azure cloud, enable for in addition to azure, AWS,GCP and on-prem

Building a pipeline for .net core app

1.New build pipeline

2. Select source control env.

3. Select repo

4. Azure pipelines analyzes the src code and prvd “options” for build

5. This results in a “Azure-pipelines.yml” file

6. Create and Run your Build

### Demo

Yukarıdaki adımlar izlenerek Azure pipeline’da pipeline create ediliyor. Ardından VS üzerinde basic bir ASP.Net Core uygulaması açıldı Git’e dahil edilerek Azure’da publish seçeneğine tıklanıyor, ardında repo seçilerek repo publish ediliyor. Bu şekilde oluşturulan repo Azure Pipeline üzerine geçiriliyor.

Azure pipeline ASP .Net Core olarak configure ediliyor ve commit ediliyor. Ardından build aşamalarından geçiyor ve pipeline oluşması sonucu bir artifact dosyası yaratılıyor.

### Deploy süreci

Azure Pipeline Release özelliği ile deployment yapılabiliyor. Bu aşamada release özelliği tıklanarak stage aşamaları hangisi tercih ediliyorsa oluşturuluyor(prod,staging,test etc.), deploy edilecek stage Azure Web App seçilerek Linux ortamda .Net Core runtime’a sahip bir stage’de release edilebiliyor.

Bu aşamada Create new Pipeline denerek, Azure Web App seçiliyor, az önce repo edilen demo artifact ve production içerisinde Azure subscribation seçilerek, App Type Web App on Linux seçiliyor. Save edilerek Pipeline tasarlanmış oluyor.

Save ettikten sonra ise Create Release butonu oluşuyor. Bu buton tıklandıktan sonra mevcut artifact seçiliyor ve release oluşuyor ve auto deployment seçiliyken production pipeline üzerinde deployment başlatılıyor ve portal.azure.com’da deployment başarılı bir şekilde sonlanarak demo canlıya çıkıyor.

Örneğin 3 servisi olan(staging,test, prod) bir uygulamada 100 deployment yapılmak istendiğinde monolitik yöntemle 300 deployment yapılması gerekir. Ancak Azure Pipeline ve release ile çok daha sürdürülebilir bir CI/CD süreci mümkün hale geliyor.

Open Source projeler için Azure Pipeline’da sınırsız build minutes mevcut. Bunun dışında 1 agent değil 10 ücretsiz paralel Windows, Linux ve macOS veriliyor.

## Agents

### Microsoft-Hosted Agents

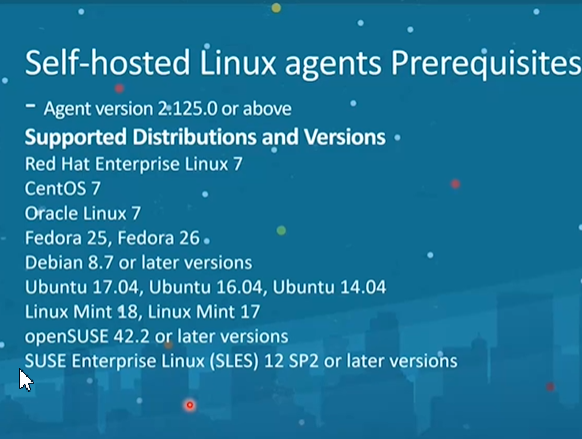
Microsoft’un daha önce kullanım için hazırladığı sanal linkinler olarak tanımlamak mümkün. Bunlara erişiminiz olmuyor ancak uygulamanız örneğin Ubuntu veya VS gerektiriyorsa bunu seçebiliyorsunuz.

<https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/hosted?view=azure-devops&tabs=yaml>

Bu link üzerinden hangi ortamlar ve arkaplanda derleme yapılması için hangi toolların kullanıldığı ile ilgili detaylar mevcut. Chocolatey ile mevcutta bulunmayan toollar da kurulabilir. Ancak bu kurulum örn. 2dk sürmekteyse pipeline her açıldığında o süre kadar bekler. Build çıktısı olarak max. 10 GB bir çıktı limiti mevcut. Build agentlar en fazla 6 saat çalışabiliyor ve Windows administrator ve şifresiz Linux sudo userı kullanıcısı supervisor olarak çalışıyor. Windows makinaya RDP veya başka bir yöntemle login olunamıyor.

### Self-Hosted Agents

Bu agent formatında siz kendi makinalarınızı build veya release agenta çeviriyorsunuz, bu sayede esnek olarak derlemeyi kendi ortamlarınızda yapabiliyorsunuz. Olmayan toolları her defasında yeniden kurmak durumunda kalmıyorsunuz.



### Microsoft-Hosted Agents vs. Self-Hosted Agents

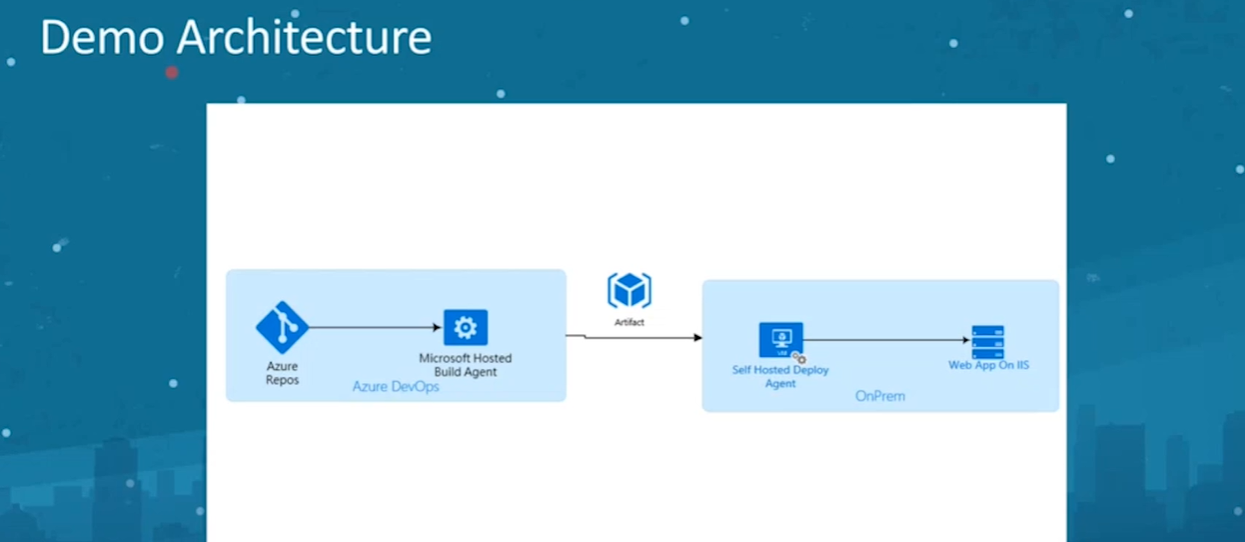
Microsoft Hosted-Agents bakım gerektirmiyor buna karşın Self-Hosted Agents bakım gerektiriyor. MS host-agents ayda 1 otomatik olarak güncelleniyor.

MS host agents firewall arkasında çalışmıyor(çalışabiliecek şekilde kurgulanabilir) ancak Self hosted agents çalışabiliyor.

MS host agents self hosted agentsa göre derleme ve artifact çıkma hızında cachinglerinden dolayı daha yavaş kalabiliyor.

MS host agentta çıktığınız releasein süresi yoğunluğa bağlı artabiliyor, selfte size ait olduğunda eğer çalışıyorsa direk başlıyor.

### Demo 2



Azure repos’a bir ASP.NET uygulaması gönderilerek, bu repoyu MS-Hosted Agentta derleyip, çıktı artifact bir ISS makinası Self-Hosted olarak kurgulanarak uygulama ISS üzerine deploy ediliyor.

MVC bir ASP .Net uygulaması oluşturuldu. Bunun pipeline’ı hazırlandı. Bu MVC ASP.NET uygulamasından artifact çıkarılarak release self hosted Agent deployment grouptan Azure’a register ediliyor ve makine üzerine deploy ediliyor.