

Bulut hizmet modeli ve bulut türü seçildikten sonra, müşterilerin altyapı mimarisini planlaması gerekmektedir. Altyapı katmanı, bulutun temelini oluşturur ve fiziksel kaynaklardan oluşur.

#### Altyapı Katmanı

- Bulut sağlayıcısının IT ortamı, dünya genelinde birçok Bölge (Region) ve Alan (Zone) içinde dağıtılmıştır.
- Her Bulut Bölgesi, doğal afetler gibi durumlarda diğer bölgelerin çalışmaya devam etmesini sağlamak için birbirinden izole edilmiştir.

#### Veri Merkezleri (Data Centers)

- Bulut veri merkezleri, sunucular, depolama ve ağ ekipmanları gibi fiziksel IT ortamının tüm bileşenlerini içeren büyük odalardır.
- Veri merkezleri, yüksek bant genişliğine sahip ağ bağlantıları ile diğer AZ'ler ve bölgelerle bağlantılıdır.

#### Hesaplama Kaynakları (Computing Resources)

- Bulut sağlayıcıları, sanal sunucular (Virtual Servers), fiziksel sunucular (Bare Metal Servers) ve sunucusuz (Serverless) hesaplama kaynakları sunar.
- Kullanıcılar, ihtiyaç duyduklarında sanal makineleri (VMs) ve fiziksel sunucuları (Bare Metal) sağlayabilirler.

Bu özet, bulut altyapısının temel bileşenlerini ve işleyişini anlamana yardımcı olacaktır.

Sanallaştırma (Virtualization) teknolojisi, bulut bilişim (Cloud Computing) stratejilerinin temelini oluşturur. Aşağıda bu konunun ana hatlarıyla özetlenmiş hali bulunmaktadır:

#### Sanallaştırma Nedir?

- Sanallaştırma, fiziksel sunucular (Physical Servers), depolama (Storage), ağ (Networking) ve uygulamalar (Applications) gibi kaynakların yazılım tabanlı sanal versiyonlarını oluşturma sürecidir.
- Bu sürecin temel bileşeni, kaynakları sanal ortamlara tahsis eden yazılım olan hipervizördür (Hypervisor).

#### Hipervizör Türleri

- **Tip 1 Hipervizör (Type 1 Hypervisor):** Fiziksel sunucunun üzerine doğrudan kurulan ve genellikle daha güvenli olan bare-metal hipervizörlerdir. Örnekler: VMware ESXi, Microsoft Hyper-V.
- **Tip 2 Hipervizör (Type 2 Hypervisor):** Fiziksel sunucu ile hipervizör arasında bir ana işletim sistemi (Host OS) katmanı bulunan, genellikle son kullanıcı sanallaştırması için kullanılan hipervizörlerdir. Örnekler: Oracle VirtualBox, VMware Workstation.

#### Sanal Makineler (Virtual Machines - VMs)

- Sanal makineler, bağımsız işletim sistemleri ve uygulamaları olan yazılım tabanlı bilgisayarlardır.
- Farklı işletim sistemleri (Windows, Linux, UNIX) aynı hipervizör üzerinde çalışabilir ve taşınabilirlikleri yüksektir.

#### Sanallaştırmanın Faydaları

- **Maliyet Tasarrufu (Cost Savings):** Birden fazla sanal ortamın tek bir fiziksel altyapıdan çalıştırılması, fiziksel sunucu sayısını azaltarak maliyetleri düşürür.
- **Hız ve Çeviklik (Agility and Speed):** Sanal makinelerin hızlı bir şekilde oluşturulması, geliştiricilerin yeni ortamlar kurmasını kolaylaştırır.
- **Kesinti Süresinin Azalması (Lower Downtime):** Sunucu arızası durumunda sanal makinelerin başka bir hipervizöre hızlıca taşınabilmesi, iş sürekliliği sağlar.

#### Sanal Makineler (Virtual Machines)

- Sanal makineler, bulut sağlayıcılarına göre farklı adlarla anılabilir ve çeşitli yapılandırmalarla sunulabilir.
- Kullanıcılar, sanal sunucular (virtual servers) oluştururken bölge (Region) ve veri merkezi (Data Center) seçebilirler.

#### Sanal Makine Türleri

- Paylaşımlı veya Kamu Bulut (Shared or Public Cloud) VMs, çoklu kiracı (multi-tenant) yapıda olup, önceden tanımlanmış boyutlarla talep üzerine sağlanır.
- Geçici veya Spot VMs (Transient or Spot VMs), kullanılmayan kapasiteyi düşük maliyetle sunar, ancak her an devre dışı bırakılabilir.

#### Rezerve Edilmiş Sanal Sunucular (Reserved Virtual Server Instances)

- Kullanıcılar, belirli bir süre için kapasite rezerve edebilir ve bu kapasiteyi ihtiyaç duyduklarında kullanabilirler.
- Uzun süreli taahhütler, maliyetleri düşürebilir.

#### Özel Sunucular (Dedicated Hosts)

- Tek kiracı (single-tenant) izolasyonu sağlar ve yalnızca belirli bir kullanıcının sanal makineleri üzerinde çalışır.
- Genellikle uyum ve düzenleyici gereklilikleri karşılamak için kullanılır.

Bu video, **bare metal server** (fiziksel sunucu) kavramını ve özelliklerini açıklamaktadır.

#### Bare Metal Server Tanımı

- **Bare metal server**, tek bir müşteri için ayrılmış, özel bir fiziksel sunucudur.
- Bulut sağlayıcısı, bu sunucuyu bir veri merkezinde (data center) bir rafa yerleştirir ve işletim sistemine (operating system) kadar yönetimini üstlenir.

#### Yönetim ve Özelleştirme

- Müşteri, sunucunun geri kalanını yönetmekten sorumludur.
- Sunucular, önceden yapılandırılmış (pre-configured) veya müşteri spesifikasyonlarına göre özelleştirilebilir (custom-configured).

### Performans ve Kullanım Alanları

- **Bare metal sunucular**, yüksek performans gerektiren uygulamalar (high-performance computing) ve veri yoğun uygulamalar (data-intensive applications) için idealdir.
- Örnek kullanım alanları arasında **ERP**, **CRM**, **AI**, ve **deep learning** bulunmaktadır.

### Fiyatlandırma ve Sağlayıcılar

- Genellikle, bare metal sunucular, sanal sunuculardan (virtual machines) daha pahalıdır.
- Tüm bulut sağlayıcıları bare metal sunucu sunmamaktadır.

### Avantajlar ve Dezavantajlar

- **Avantajlar:** Yüksek güvenlik (high security), tam kontrol (full control), ve özelleştirme (customization) imkanı.
- **Dezavantajlar:** Uzun provisioning süreleri (provisioning times) ve yönetim yükü (management overhead) gerektirmektedir.

Bu video, bulut (Cloud) ortamlarında güvenli ağlar (secure networks) oluşturmanın önemini vurgulamaktadır. Aşağıda ana noktalar özetlenmiştir:

#### Ağ Oluşturma (Network Creation)

- Bulut ağları, fiziksel cihazlar (physical devices) yerine mantıksal örnekler (logical instances) kullanılarak oluşturulur.
- Ağın boyutu (size) ve IP adres aralığı (IP address range) tanımlanarak bulut ağı (Cloud network) sınırları belirlenir.

#### Alt Ağlar (Subnets)

- Bulut kaynakları (Cloud resources) sanal alt ağlar (subnets) içinde dağıtılır ve güvenlik (security) burada uygulanır.
- Her alt ağ, erişim kontrol listeleri (access control lists - ACLs) ile korunur ve güvenlik grupları (security groups) oluşturulabilir.

#### Yük Dengeleme (Load Balancing) ve VPN (Virtual Private Network)

- Uygulamaların sürekli erişilebilirliğini sağlamak için yük dengeleyiciler (load balancers) kullanılır.
- Şirketler, kendi yerel kaynaklarını (on-premises resources) buluta güvenli bir şekilde bağlamak için VPN'ler (VPNs) kullanabilir.

Bu yapı, bulut ortamında yüksek performanslı iş uygulamalarını (business applications) güvence altına almak için gereklidir.

vNICs (Virtual Network Interface Controllers) ve NICs (Network Interface Controllers) ağ bağlantı noktalarıdır, ancak farklı ortamlarda kullanılırlar.

#### **NIC (Network Interface Controller):**

- Fiziksel bir cihazdır ve bir bilgisayarın veya sunucunun (server) ağa bağlanmasını sağlar.
- Genellikle bir ağ kablosu (network cable) aracılığıyla fiziksel bir ağa bağlanır.
- NIC'ler, veri iletimini (data transmission) sağlamak için donanım tabanlı bir çözüm sunar.

#### **vNIC (Virtual Network Interface Controller):**

- Sanal bir versiyonudur ve bulut (cloud) ortamlarında kullanılır.
- Fiziksel bir NIC'in işlevselliğini taklit eder, ancak fiziksel bir cihaza ihtiyaç duymaz.
- vNIC'ler, sanal makineler (virtual machines) gibi sanal kaynakların (virtual resources) ağ bağlantılarını yönetmek için kullanılır.

Özetle, NIC'ler fiziksel ağ bağlantıları sağlarken, vNIC'ler sanal ortamda benzer işlevleri yerine getirir. Bu, bulut bilişimde esneklik ve ölçeklenebilirlik (scalability) sağlar.

**containerization** (konteynerleştirme) kavramını ve avantajlarını açıklamaktadır.

#### **Konteynerlerin Tanımı**

- **Containers** (konteynerler), uygulama kodunun, kütüphanelerin ve bağımlılıkların paketlenildiği yazılım birimidir.
- **Portability** (taşınabilirlik) sağlar; masaüstü, geleneksel IT veya bulut üzerinde çalışabilir.

#### **Konteyner ve Sanal Makineler (VMs) Arasındaki Farklar**

- **Virtual Machines** (sanal makineler), her bir örnekte bir **guest OS** (misafir işletim sistemi) içermelidir, bu da kaynak tüketimini artırır.
- **Containers**, yalnızca gerekli kütüphaneleri ve uygulamayı içerir, bu da daha az kaynak kullanımı sağlar.

#### **Konteyner Oluşturma Süreci**

- İlk adım, bir **manifest** (manifesto) oluşturmaktır; bu, konteynerin tanımını içerir.
- İkinci adım, **image** (görüntü) oluşturmaktır; bu, konteynerin çalışması için gereken tüm bileşenleri içerir.
- Son adım, **container** (konteyner) oluşturmaktır; bu, uygulamanın çalışması için gerekli tüm bileşenleri barındırır.

#### **Konteynerlerin Avantajları**

- **Scalability** (ölçeklenebilirlik) sağlar; aynı uygulamanın birden fazla kopyasını daha az kaynakla dağıtabilirsiniz.

- **Agile DevOps** (çevik DevOps) ve **Continuous Integration/Delivery** (sürekli entegrasyon/teslimat) süreçlerini kolaylaştırır.

Konteynerleştirme (containerization) sürecini uygulamak için aşağıdaki adımları izleyebilirsiniz:

### 1. Manifest Oluşturma:

- Uygulamanın yapılandırmasını tanımlayan bir **Dockerfile** (Docker dosyası) oluştur. Bu dosya, konteynerin nasıl oluşturulacağını ve hangi bileşenlerin dahil edileceğini belirtir.

### 2. Görüntü Oluşturma:

- Dockerfile kullanarak bir **image** (görüntü) oluştur. Bu işlem, uygulamanın çalışması için gerekli tüm bileşenleri içeren bir paket oluşturur. Örneğin, terminalde şu komutu kullanabilirsiniz:

```
docker build -t myapp:latest .
```

### 3. Görüntüyü Kaydetme:

- Oluşturduğun görüntüyü bir **registry** (kayıt yeri) üzerine it. Bu, görüntünün paylaşılmasını ve dağıtılmasını sağlar. Örneğin:

```
docker push myapp:latest
```

### 4. Konteyner Oluşturma:

- Oluşturduğun görüntüyü kullanarak bir **container** (konteyner) başlat. Bu, uygulamanın çalıştığı bir ortam sağlar. Örneğin:

```
docker run -d -p 80:80 myapp:latest
```

### 5. Yönetim ve İzleme:

- Konteynerin durumunu izlemek ve yönetmek için Docker komutlarını kullan. Örneğin, çalışan konteynerleri listelemek için:

```
docker ps
```

Bu adımlar, konteynerleştirme sürecini uygulamak için temel bir rehberdir. Her adımda, uygulamanın ihtiyaçlarına göre özelleştirmeler yapabilirsiniz.