2025-Yaz Dönemi

**Doç. Dr. Orhan ECEMİŞ**

**Sanallaştırma ve Bulut Teknolojileri**

**2025-Yaz Dönemi**

Sanallaştırma ve Bulut Teknolojileri



BU BÖLÜM GOOGLE ASSOCIATE BULUT MÜHENDİSLİĞİ SERTİFİKASYON SINAVININ AŞAĞIDAKİ AMACI KAPSAMAKTADIR:

✓ 1.0 Bulut projeleri ve hesap kurulumu

Google Cloud, Google'ın kendi ürünlerini sunmak için kullandığı teknolojilerin bazılarını sunan bir genel bulut hizmetidir. Bu bölüm, Google Cloud'un en önemli bileşenlerini ve şirket içi veri merkezi tabanlı bilgi işlemden nasıl farklılaştığı ele alınmıştır.

# **Bulut Hizmetleri Türleri**

Google, Amazon ve Microsoft gibi genel bulut sağlayıcıları, çok çeşitli iş hizmetleri ve uygulamaları çalıştırmak için bilgi işlem, depolama, ağ ve diğer altyapıları dağıtmak için bir dizi hizmet sunar. Bazı bulut kullanıcıları, bulutta başlayan yeni şirketlerdir. Hiçbir zaman kendi donanım altyapılarına sahip olmamışlardır. Diğer bulut müşterileri, veri merkezlerini desteklemek için genel bulutları kullanan birden fazla veri merkezine sahip işletmelerdir. Bu farklı kullanıcı türlerinin farklı gereksinimleri vardır.

Bulutta başlayan bir şirket, mevcut altyapıyı dikkate almak zorunda kalmadan uygulamasına ve mimari ihtiyaçlarına en uygun hizmetleri seçebilir. Örneğin, bir girişim, tüm kimlik doğrulama ve yetkilendirme ihtiyaçları için Kimlik Erişim Yönetimi hizmetleri de dahil olmak üzere Google Cloud'un Bulut Kimliğini kullanabilir. Kimlik yönetimi için Microsoft Active Directory çözümüne yatırım yapmış bir şirket, yalnızca bulutun kimlik yönetim sistemiyle çalışmak yerine bu sistemi kullanmak isteyebilir. Bu, iki sistemi entegre etmek ve senkronize tutmak için ek çalışmaya yol açabilir.

Kendi altyapısına sahip işletmeler için bir diğer endişe alanı, şirket içi kaynakları ile genel bulut kaynakları arasında güvenli bir ağ kurmak ve sürdürmektir. Şirket içi sistemler ile genel bulut arasında yüksek hacimli ağ trafiği olacaksa, işletmenin veri merkezi ile genel bulut sağlayıcısının bir tesisi arasında özel ağ yatırımı yapması gerekebilir. Trafik hacmi, tesisler arasında özel bir bağlantının maliyetini karşılamıyorsa, şirket genel İnternet üzerinden çalışan sanal özel bir ağ kullanabilir. Bu, yalnızca bulutta bulunan bir şirketin ele almak zorunda kalmayacağı ek ağ tasarımı ve yönetimi gerektirir.

Genel bulut sağlayıcıları, dört geniş kategoriye giren hizmetler sunar:

* Hesaplama kaynakları
* Depolama
* Ağ
* Makine öğrenimi hizmetleri gibi özel hizmetler

Bulut müşterileri genellikle bu kategorilerden birden fazlasındaki hizmetlerden yararlanır.

**Bilgi İşlem Kaynakları**

Genel bulutlarda çeşitli biçimlerde bilgi işlem kaynakları bulunur.

**Sanal Makineler**

Sanal makineler (VM'ler), bilgi işlem kaynaklarının temel birimidir ve bulutu deneyimlemek için iyi bir başlangıç noktasıdır. Bir bulut sağlayıcısında hesap oluşturup fatura bilgilerinizi sağladıktan sonra, Google Cloud kaynaklarının mantıksal bir gruplandırması olan bir proje oluşturabilirsiniz. Bir projeniz olduğunda, bir portal veya komut satırı araçlarını kullanarak bir proje içinde VM'ler oluşturabilirsiniz. Google Cloud, değişen sayıda vCPU ve bellek miktarına sahip çeşitli önceden yapılandırılmış VM'ler sunar. Önceden yapılandırılmış seçenekler ihtiyaçlarınızı karşılamıyorsa özel bir yapılandırma da oluşturabilirsiniz. Bir VM oluşturduktan sonra, ona giriş yapabilir ve istediğiniz gibi yönetebilirsiniz. VM'ye tam erişiminiz vardır, bu nedenle dosya sistemlerini yapılandırabilir, kalıcı depolama ekleyebilir, işletim sistemini güncelleyebilir veya ek paketler yükleyebilirsiniz. VM üzerinde ne çalıştıracağınıza, kimlerin erişimi olacağına ve VM'yi ne zaman kapatacağınıza siz karar verirsiniz. Yönettiğiniz bir VM, ofisinizde tam yönetici haklarına sahip olduğunuz bir sunucuya sahip olmak gibidir.

Elbette, farklı işletim sistemleri ve uygulamalar çalıştıran birden çok VM oluşturabilirsiniz. Google Cloud ayrıca, dağıtılmış bir arka uca tek bir erişim noktası sağlayan yük dengeleyiciler gibi hizmetler de sunar. Bu, uygulamanız için yüksek kullanılabilirlik gerektiğinde özellikle yararlıdır. Bir kümedeki VM'lerden biri arızalanırsa, iş yükü kümedeki diğer VM'lere yönlendirilebilir. Otomatik ölçeklendiriciler, iş yüküne bağlı olarak kümeden VM ekleyebilir veya çıkarabilir. Buna otomatik ölçeklendirme denir. Bu, hem gerekenden fazla VM çalıştırmayarak maliyeti kontrol etmeye yardımcı olur hem de iş yükleri arttığında yeterli bilgi işlem kapasitesinin mevcut olmasını sağlar.

**Yönetilen Kubernetes Kümeleri**

Google Cloud, sunucu kümeleri oluşturmak ve yönetmek için ihtiyacınız olan tüm araçları sunar. Birçok bulut kullanıcısı, sunucu kümesini çalışır durumda tutmak için gereken görevler yerine uygulamalarına odaklanmayı tercih eder. Bu kullanıcılar için yönetilen kümeler iyi bir seçenektir.

Yönetilen kümeler, konteynerleri kullanır. Bir konteyner, aynı sunucudaki diğer konteynerlerde çalışan işlemlerden yalıtılmış, hafif bir VM gibidir. Yönetilen bir kümede, çalıştırmak istediğiniz sunucu sayısını ve üzerlerinde çalışması gereken konteynerleri belirtebilirsiniz. Ayrıca, çalışan konteyner sayısını optimize etmek için otomatik ölçeklendirme parametreleri de belirtebilirsiniz.

Yönetilen bir kümede, konteynerlerin sağlığı sizin için izlenir. Bir konteyner arızalanırsa, küme yönetim yazılımı bunu algılar ve başka bir konteyner başlatır.Konteynerler, ortamınızda çalışan birden çok mikro hizmete bağımlı uygulamaları çalıştırmanız gerektiğinde iyi seçeneklerdir. Hizmetler konteynerler aracılığıyla dağıtılır ve küme yönetim hizmeti izleme, ağ oluşturma ve bazı güvenlik yönetimi görevlerini üstlenir.

Tablo K8s Nedir?

|  |
| --- |
| Şimdi Gaziantep'tesin, değil mi? Diyelim ki büyük bir kebap restoranı işletiyorsun. Müşterilerin siparişlerini yetiştirmek, malzemeleri düzenlemek, personeli yönetmek gibi bir sürü işin var. Tek başına hepsine yetişmek zor olabilir. İşte Kubernetes kümeleri de tam olarak bu noktada devreye giriyor, ama kebap yerine uygulamalarımızı yönetmek için.  **Temel Fikir: Uygulamaları Yöneten Bir Orkestra Şefi**  Kubernetes (genellikle "K8s" olarak kısaltılır), birden fazla sunucuyu (biz bunlara "node" diyoruz) tek bir mantıksal birim gibi yönetmemizi sağlayan açık kaynaklı bir sistemdir. Bu sunucular üzerinde uygulamalarımızın küçük parçaları olan "konteynerler" çalışır. Kubernetes, bu konteynerlerin ne zaman çalıştırılacağını, nerede çalıştırılacağını, nasıl ölçekleneceğini (yani ihtiyaç arttıkça daha fazla konteyner eklenmesini veya ihtiyaç azaldıkça azaltılmasını) ve sağlıklı kalmalarını otomatik olarak yönetir. Tıpkı bir orkestra şefinin tüm müzisyenleri uyum içinde yönetmesi gibi.  **Kubernetes'in Temel Bileşenleri:**   * **Node (Düğüm):** Uygulamalarımızın (konteynerlerimizin) üzerinde çalıştığı fiziksel veya sanal makinelerdir. Her node, uygulamaları çalıştırmak için gerekli olan Docker (veya benzeri bir konteyner motoru) ve Kubernetes'in bir parçası olan "kubelet" adlı bir aracı çalıştırır. Düğüm, işçinin kendisidir. * **Pod (Pod):** Kubernetes'in en temel yapı taşıdır. Bir veya daha fazla yakından ilişkili konteyneri (genellikle aynı uygulama veya birbirine sıkı sıkıya bağlı servisler) barındırır. Pod'lar, aynı IP adresini ve portlarını paylaşır ve birbirleriyle kolayca iletişim kurabilirler. Bir nevi "uygulama birimi" diyebiliriz. Örneğin, bir web uygulamasının ön yüz konteyneri ve arka uç API konteyneri aynı pod içinde çalışabilir. * **Control Plane (Kontrol Düzlemi):** Kubernetes kümesinin beynidir. Kümenin genel durumunu yönetir ve istenen durumu korumaktan sorumludur. Başlıca bileşenleri şunlardır:   + **kube-apiserver:** Kubernetes API'sini sunar. Kullanıcılar, komut satırı araçları (kubectl gibi) veya diğer API istemcileri aracılığıyla küme ile bu API üzerinden iletişim kurar.   + **etcd:** Kümenin tüm yapılandırma verilerini güvenli ve tutarlı bir şekilde saklayan yüksek oranda erişilebilir bir anahtar-değer deposudur.   + **kube-scheduler:** Yeni oluşturulan pod'ları hangi node üzerinde çalıştırılacağına karar verir. Kaynak kullanımı, donanım gereksinimleri ve node'ların mevcut durumu gibi faktörleri dikkate alır.   + **kube-controller-manager:** Kümedeki çeşitli kontrol döngülerini yönetir. Örneğin, "replication controller" istenen sayıda pod'un her zaman çalışır durumda olmasını sağlar. Bir pod çökerse, otomatik olarak yenisini başlatır.   + **cloud-controller-manager (isteğe bağlı):** Bulut sağlayıcısına özgü işlemleri yönetir (örneğin, yük dengeleyicileri oluşturma, depolama birimleri sağlama). * **kubectl:** Kubernetes kümesiyle etkileşim kurmak için kullanılan komut satırı aracıdır. Uygulamaları dağıtmak, kümeyi incelemek ve yönetmek için kullanılır.   **Kubernetes'in Faydaları:**   * **Otomatik Ölçeklendirme:** Uygulamanızın yükü arttığında otomatik olarak daha fazla pod (ve dolayısıyla daha fazla kaynak) ekleyebilir, yük azaldığında ise kaynakları geri çekebilir. Restoran örneğimizde, yoğun saatlerde daha fazla garsonun çalışması gibi düşünebilirsin. * **Yüksek Erişilebilirlik:** Uygulamalarınızın her zaman çalışır durumda kalmasını sağlar. Bir pod veya node arızalanırsa, Kubernetes otomatik olarak yenilerini başlatır. Restorandaki bir aşçının hastalanması durumunda yerine başka birinin geçmesi gibi. * **Kolay Dağıtım ve Yönetim:** Uygulamaların dağıtımını, güncellenmesini ve geri alınmasını kolaylaştırır. Yeni bir menü eklemek veya mevcut bir yemeğin tarifini güncellemek gibi. * **Kaynak Verimliliği:** Kaynakların daha verimli kullanılmasını sağlar. Pod'lar, ihtiyaç duydukları kadar kaynak (CPU, bellek vb.) kullanır ve bu kaynaklar dinamik olarak yönetilir. * **Taşınabilirlik:** Farklı bulut sağlayıcılarında veya şirket içi altyapılarda tutarlı bir şekilde çalışabilir. Restoranının şubelerini farklı şehirlerde açmak gibi.   **Yönetilen Kubernetes Kümeleri (Managed Kubernetes Clusters):**  Google Cloud'daki "Managed Kubernetes Clusters" (örneğin Google Kubernetes Engine - GKE), Kubernetes kümelerinin kurulumu, yönetimi ve bakımı gibi karmaşık görevlerin büyük bir kısmını Google'ın üstlendiği bir hizmettir. Sen sadece uygulamalarını konteynerleştirip kümeye nasıl dağıtılacağını belirtirsin, gerisini Google halleder. Bu, kendi Kubernetes altyapını kurup yönetmekle uğraşmak yerine doğrudan uygulamalarına odaklanmanı sağlar. Tıpkı restoranının tüm altyapısı ve yönetimiyle ilgilenmek yerine sadece yemek yapmaya ve müşterilere odaklanmak gibi. |

Tablo 2Konteyner Nedir?

|  |
| --- |
| Şimdi, kebap restoranınızın mutfağında farklı görevleri yerine getiren ustalarınız var, değil mi? Biri sadece etleri marine ediyor, diğeri şişe diziyor, bir başkası ızgarayı yönetiyor ve bir diğeri de salata yapıyor. Her ustanın kendi uzmanlık alanı, kendi araçları ve kendi çalışma ortamı var. Birinin işi diğerini çok fazla etkilemiyor.  İşte **konteynerler**, tıpkı restoranınızdaki bu **uzmanlaşmış mutfak ustaları gibi düşünebilirsiniz**.   * **Her konteyner, belirli bir işi yapmak üzere paketlenmiş, kendi kendine yeten bir "mini-ortamdır."** Bu mini-ortamın içinde uygulamanızın çalışması için gereken her şey bulunur: kod, kütüphaneler, bağımlılıklar, yapılandırma dosyaları... Tıpkı et marine eden ustanın kendi baharatları, kapları ve belki de özel bir tarifi olması gibi. * **Konteynerler birbirinden izoledir.** Yani, bir konteynerin içinde olan biten, diğer konteynerleri doğrudan etkilemez. Şişe dizen ustanın elindeki kebap malzemeleri, salata yapan ustanın domateslerini karıştırmaz. Bu izolasyon, uygulamalarınızın daha güvenli ve kararlı çalışmasını sağlar. Bir konteynerde bir sorun çıksa bile, diğerleri etkilenmez. * **Konteynerler hafiftir ve hızlı başlatılır.** Yeni bir müşteri geldiğinde hemen yeni bir kebap şişi hazırlanması gerektiği gibi, bir konteyner de hızlı bir şekilde başlatılabilir ve durdurulabilir. Bu, uygulamalarınızın hızlı bir şekilde ölçeklenmesini sağlar. * **Konteynerler taşınabilirdir.** Tıpkı bir ustanın kendi özel araçlarını ve tariflerini alıp başka bir şubede de aynı işi yapabilmesi gibi, bir konteyner de farklı ortamlarda (geliştirme bilgisayarınızda, test sunucunuzda, bulutta) tutarlı bir şekilde çalışabilir.   **Kubernetes bağlamında düşünürsek:**  Kubernetes, bu "uzmanlaşmış usta" konteynerleri yöneten bir **"baş aşçı" veya "restoran müdürü"** gibidir.   * Hangi ustanın (konteynerin) ne zaman çalışması gerektiğine karar verir. * Eğer bir usta (konteyner) hastalanırsa (çökerse), otomatik olarak yerine yenisini bulur (yeni bir konteyner başlatır). * Yoğun saatlerde (yüksek trafik) daha fazla ustanın (konteynerin) çalışmasını sağlar, sakin saatlerde ise sayıyı azaltır (otomatik ölçeklendirme). * Tüm ustaların (konteynerlerin) malzemelere (kaynaklara) eşit ve verimli bir şekilde erişmesini sağlar.   Özetle, **konteynerler uygulamalarınızı paketleyip izole eden, hafif, hızlı başlatılan ve taşınabilir "birimlerdir".** Kubernetes ise bu birimleri yöneterek uygulamalarınızın verimli, güvenilir ve ölçeklenebilir bir şekilde çalışmasını sağlayan orkestra şefidir. |

**Sunucusuz Bilgi İşlem**

Hem sanal makineler hem de yönetilen Kubernetes kümeleri, bilgi işlem kaynaklarını yapılandırmak ve yönetmek için bir miktar çaba gerektirir. Sunucusuz bilgi işlem, geliştiricilerin ve uygulama yöneticilerinin kodlarını, sanal makineler veya Kubernetes kümeleri kurmayı gerektirmeyen bir bilgi işlem ortamında çalıştırmalarına olanak tanıyan bir yaklaşımdır.

Google Cloud'da üç sunucusuz bilgi işlem seçeneği bulunur: App Engine, Cloud Run ve Cloud Functions. App Engine, web sitesi arka ucu, satış noktası sistemi veya özel iş uygulaması gibi uzun süre çalışan uygulamalar ve konteynerler için kullanılır. Cloud Run, Kubernetes Engine'in tüm özelliklerine ihtiyaç duyulmadığında konteynerleri çalıştırmak için de kullanılır. Cloud Run, durum bilgisiz uygulamalarınız için tam olarak yönetilen bir hizmet ve hızlı otomatik ölçeklendirme istediğinizde konteynerler için kullanılır. Cloud Functions, bir dosyayı yükleme veya bir mesaj kuyruğuna mesaj ekleme gibi bir olaya yanıt olarak kod çalıştırmak için bir platformdur. Bu sunucusuz seçenek, bir işlevde kodlanmış kısa bir süreci çalıştırarak veya bir VM, yönetilen küme veya App Engine üzerinde çalışıyor olabilecek daha uzun süreli bir uygulamayı çağırarak bir olaya yanıt vermeniz gerektiğinde iyi çalışır.

**Depolama**

Genel bulutlar, çok çeşitli uygulama gereksinimleri için yararlı olan birkaç tür depolama hizmeti sunar. Bu türler şunları içerir:

* Nesne depolama
* Dosya depolama
* Blok depolama
* Önbellekler

Bulut hizmetlerinin kurumsal kullanıcıları genellikle bu hizmetlerin bir kombinasyonunu kullanır.

**Nesne Depolama**

Nesne depolama, depolama kullanımını nesneler veya blob'lar açısından yöneten bir sistemdir. Genellikle bu nesneler dosyalardır, ancak dosyaların geleneksel bir dosya sisteminde depolanmadığını unutmamak önemlidir. Nesneler, "bucket" adı verilen gruplar halinde düzenlenir. Her nesneye genellikle bir URL aracılığıyla ayrı ayrı erişilebilir.

Nesne depolama, bir sunucuya bağlı disklerin veya katı hal sürücülerinin (SSD'ler) boyutuyla sınırlı değildir. Nesneler, bir diskte ne kadar boş alan olduğuna bakılmaksızın yüklenebilir. Erişilebilirliği ve dayanıklılığı artırmak için nesnelerin birden çok kopyası depolanır. Bazı durumlarda, bir bölgeye erişilemez hale gelse bile erişilebilirliği sağlamak için nesne kopyaları farklı bölgelerde depolanabilir.

Nesne depolamanın bir diğer avantajı da sunucusuz olmasıdır. VM oluşturmaya ve onlara depolama birimleri bağlamaya gerek yoktur. Google Cloud'un Cloud Storage adlı nesne depolama hizmetine, Google Cloud'da çalışan sunuculardan ve internet erişimi olan diğer cihazlardan erişilebilir.

**Dosya Depolama**

Dosya depolama hizmetleri, dosyalar için hiyerarşik bir depolama sistemi sunar. Dosya sistemi depolaması, ağ üzerinden paylaşılan dosya sistemleri sağlar. Google Cloud'da, Ağ Dosya Sistemi (NFS) depolama sistemini temel alan Cloud Filestore adlı bir dosya depolama hizmeti bulunur.

Dosya depolama, dosyalara işletim sistemi benzeri dosya erişimi gerektiren uygulamalar için uygundur. Dosya depolama sistemi, dosya sistemini belirli sanal makinelerden ayırır. Dosya sistemi, dizinleri ve dosyaları, bu dosyalara erişebilecek sanal makinelerden veya uygulamalardan bağımsız olarak var olur.

**Blok Depolama**

Blok depolama, verileri düzenlemek için sabit boyutlu bir veri yapısı olan bloğu kullanır. Blok depolama, genellikle sanal makinelere bağlı geçici ve kalıcı disklerde kullanılır. Bir blok depolama sistemi ile, blok depolamanın üzerine dosya sistemleri kurabilir veya doğrudan bloklara erişen uygulamalar çalıştırabilirsiniz. Bazı ilişkisel veritabanları, dosya sistemleri üzerinden çalışmak yerine doğrudan bloklara erişecek şekilde tasarlanabilir.

Linux dosya sistemlerinde, 4 KB yaygın bir blok boyutudur. İlişkisel veritabanları genellikle doğrudan bloklara yazarlar, ancak genellikle 8 KB veya daha büyük boyutlar kullanırlar.

Blok depolama, Google Cloud'daki sanal makinelere bağlı disklerde mevcuttur. Blok depolama kalıcı veya geçici olabilir. Kalıcı bir disk, bir sanal sunucudan ayrılsa veya bağlı olduğu sanal sunucu kapansa bile var olmaya ve veri depolamaya devam eder. Geçici diskler yalnızca bir sanal makine çalıştığı sürece var olur ve veri depolar. Sanal makine kapatıldığında geçici diskler silinir. Verilerin bir sanal makinenin yaşam döngüsünden bağımsız olarak bir blok depolama aygıtında var olmasını istediğinizde kalıcı diskler kullanılır. Bu diskler, bir VM'nin yaşam döngüsünden bağımsız olarak erişilebilir olmasını istediğiniz verilere sahip olduğunuzda ve hızlı işletim sistemi ve dosya sistemi düzeyinde erişimi desteklediğinde iyi seçeneklerdir.

Nesne depolama da verileri bir VM'nin yaşam döngüsünden bağımsız tutar, ancak işletim sistemi veya dosya sistemi düzeyinde erişimi desteklemez; nesnelere erişmek için HTTP gibi daha yüksek düzeyli protokoller kullanmanız gerekir. Nesne depolamadan veri almak, blok depolamadan veri almaktan daha uzun sürer. Uygulama ihtiyaçlarınızı karşılamak için nesne depolama ve blok depolamanın bir kombinasyonuna ihtiyacınız olabilir. Nesne depolama, gerektiğinde kalıcı diske kopyalanan büyük veri hacimlerini depolayabilir. Bu kombinasyon, gerektiğinde büyük depolama hacimlerinin yanı sıra işletim sistemi ve dosya sistemi tabanlı erişimin avantajını sunar.

**Önbellekler**

Önbellekler, verilere hızlı erişimi sağlayan bellek içi veri depolarıdır. Veri almanın aldığı süreye gecikme (latency) denir. Bellek içi depoların gecikmesi milisaniyenin altında olacak şekilde tasarlanmıştır. Bir karşılaştırma yapmak gerekirse, işte bazı diğer gecikmeler:

|  |  |
| --- | --- |
| Ana belleğe erişim 100 nanosaniye veya 0,1 mikrosaniye sürer.  Bir SSD'den rastgele 4 KB okumak 150 mikrosaniye sürer.  Bellekten sıralı olarak 1 MB okumak 250 mikrosaniye sürer.  Bir SSD'den sıralı olarak 1 MB okumak 1.000 mikrosaniye veya 1 milisaniye sürer.  Bir diskten sıralı olarak 1 MB okumak 20.000 mikrosaniye veya 20 milisaniye sürer. | Referans için bazı çeviriler:  1.000 nanosaniye 1 mikrosaniyeye eşittir.  1.000 mikrosaniye 1 milisaniyeye eşittir.  1.000 milisaniye 1 saniyeye eşittir. |

Bu ve diğer faydalı zamanlama verileri, Jonas Bonér'in "Her Programcının Bilmesi Gereken Gecikme Sayıları" adlı makalesinde (https://gist.github.com/jboner/2841832) bulunabilir.

1 MB veri okuma örneğini inceleyelim. Verileriniz bellek içi bir önbellekte depolanıyorsa, verileri 250 mikrosaniyede veya 0,25 milisaniyede alabilirsiniz. Aynı veriler bir SSD'de depolanıyorsa, 1 milisaniyede yani dört kat daha uzun sürer. Aynı verileri bir sabit disk sürücüsünden (HDD) alırsanız, bellek içi önbellekten okumaya göre 80 kat daha uzun olan 20 milisaniye beklemeniz beklenebilir.

Önbellekler, uygulamanızda okuma gecikmesini minimumda tutmanız gerektiğinde oldukça yardımcıdır. Elbette, hızlı erişim sürelerini kim sevmez ki? Neden her zaman verilerimizi önbelleklerde saklamıyoruz? Bunun üç nedeni vardır:

Bellek, SSD veya HDD depolamadan daha pahalıdır. Çoğu durumda, SSD'ler veya HDD'ler üzerindeki kalıcı blok depolama kadar bellek içi depolamaya sahip olmak pratik değildir.

Önbellekler geçicidir; güç kesildiğinde veya işletim sistemi yeniden başlatıldığında önbellekte depolanan verileri kaybedersiniz. Hızlı erişim için verileri bir önbellekte saklayabilirsiniz, ancak asla verileri tutan tek veri deposu olarak kullanılmamalıdır. Verilerin her zaman en son ve en doğru sürümünü içeren bir "doğruluk sistemi" veya veri deposunu korumak için bir tür kalıcı depolama kullanılmalıdır.

Önbellekler, doğruluk sistemiyle senkronizasyonunu kaybedebilir. Bu, doğruluk sistemi güncellendiğinde ancak yeni veriler önbelleğe yazılmadığında olabilir. Bu olduğunda, önbelleğe bağlı bir uygulamanın önbellekteki verilerin geçersiz olduğunu tespit etmesi zor olabilir. Bir önbellek kullanmaya karar verirseniz, önbellek ve doğruluk sistemi arasındaki tutarlılık gereksinimlerinizi karşılayan bir önbellek güncelleme stratejisi tasarladığınızdan emin olun. Bu o kadar zorlu bir tasarım problemidir ki, Phil Karlton'ın ünlü sözüyle anılmıştır: "Bilgisayar biliminde çözülmesi zor sadece iki şey vardır: önbellek geçersiz kılma ve isimlendirme." (Bilgisayar bilimi mizahının bu nadir örneği üzerine çeşitli yorumlar için https://martinfowler.com/bliki/TwoHardThings.html adresine bakabilirsiniz.)

Tablo Gerçek Dünya Örneği

|  |
| --- |
| **Veritabanı Sorgu Yanıt Süresini İyileştirme**  Kullanıcılar web uygulamalarının son derece duyarlı olmasını bekler. Bir sayfanın yüklenmesi 2 ila 3 saniyeden fazla sürerse, kullanıcı deneyimi olumsuz etkilenebilir. Bir sayfanın içeriğini, müşteri kimliğine göre hesap bilgilerini aramak gibi bir veritabanı sorgusunun sonuçlarını kullanarak oluşturmak yaygındır. Veritabanına bir sorgu yapıldığında, veritabanı motoru genellikle diskte bulunan verileri arar. Veritabanına ne kadar çok kullanıcı sorgu gönderirse, veritabanının hizmet vermesi gereken sorgu sayısı da o kadar artar. Veritabanları, yanıtlanması gereken ancak henüz işlenemeyen sorgular için bir kuyruk tutar çünkü veritabanı diğer sorgularla meşguldür. Bu, web uygulamasının veritabanının sorgu sonuçlarını döndürmesini beklemesi gerekeceğinden daha uzun gecikme yanıt süresine neden olabilir.  Gecikmeyi azaltmanın bir yolu, verileri okumak için gereken süreyi azaltmaktır. Bazı durumlarda, sabit disk sürücülerini daha hızlı SSD sürücülerle değiştirmek yardımcı olur. Ancak, sorgu hacmi SSD'lerle bile sorgu kuyruğunun uzun olmasına yetecek kadar yüksekse, bir diğer seçenek bir önbellek kullanmaktır.  Sorgu sonuçları alındığında, önbellekte saklanır. Bu bilgilere bir dahaki sefere ihtiyaç duyulduğunda, veritabanı yerine önbellekten alınır. Bu gecikmeyi azaltabilir çünkü veriler diskten daha hızlı olan bellekten alınır. Ayrıca veritabanına yapılan sorgu sayısını da azaltır, böylece önbellekte veri arayarak yanıtlanamayan sorgular işlenmeden önce sorgu kuyruğunda uzun süre beklemek zorunda kalmaz.  Bu yaklaşım, sorgu sonuçlarını önbellekte saklamak ve veritabanını sorgulamadan önce önbelleği veri açısından kontrol etmek için uygulama kodunda değişiklik yapılmasını gerektirir |

**Ağ Kurulumu**

Bulutta çalışırken, bulut kaynaklarınız arasındaki ve muhtemelen şirket içi sistemlerinizle olan ağ bağlantısıyla uğraşmanız gerekecektir. Bulut ortamınızda birden çok sanal makine çalıştırdığınızda, bir noktada IP adreslerini yönetmeniz gerekecektir. Ortamınızdaki ağ üzerinden erişilebilen her cihazın veya hizmetin bir IP adresine ihtiyacı olacaktır. Aslında, Google Cloud içindeki cihazlar hem dahili hem de harici adreslere sahip olabilir. Dahili adreslere yalnızca dahili ağınızdaki hizmetler erişebilir. Dahili Google Cloud ağınız, bir sanal özel bulut (VPC) olarak tanımlanır. Harici adreslere ise internetten erişilebilir. Harici IP adresleri statik veya geçici olabilir. Statik adresler, uzun süreler boyunca bir cihaza atanır. Geçici harici IP adresleri sanal makinelere bağlanır ve sanal makine durdurulduğunda serbest bırakılır.

IP adreslerini belirtmenin yanı sıra, genellikle VPC'nizdeki alt ağlara ve sanal makinelere erişimi kontrol etmek için güvenlik duvarı kuralları tanımlamanız gerekecektir. Örneğin, erişimini kısıtlamak istediğiniz bir veritabanı sunucunuz olabilir, böylece yalnızca bir uygulama sunucusu veritabanını sorgulayabilir. Bir güvenlik duvarı kuralı, uygulama sunucusunun veya uygulama kümesinin önündeki yük dengeleyicinin IP adresine gelen ve giden trafiği sınırlayacak şekilde yapılandırılabilir.

Şirket içi bir veri merkezi ile VPC'niz arasında veri ve ağ erişimini paylaşmanız gerekebilir. Bunu, farklı ağları birbirine bağlamak için genel bir terim olan çeşitli eşleme türlerinden birini kullanarak yapabilirsiniz. Google Cloud, VPN'ler, Bağlantılar Arası (Interconnects), Paylaşılan VPC (Shared VPC), VPC ağ eşlemesi ve Doğrudan veya Taşıyıcı eşlemesi dahil olmak üzere çeşitli eşleme türleri sunar.

**Özel Amaçlı Hizmetler**

Çoğu genel bulut sağlayıcısı, uygulamaların yapı taşları olarak veya veri işleme iş akışının bir parçası olarak kullanılabilecek özel amaçlı hizmetler sunar. Özel amaçlı hizmetlerin ortak özellikleri şunlardır:

* Sunucusuzdurlar; sunucuları veya kümeleri yapılandırmanız gerekmez.
* Metin çevirme veya görüntü analizi gibi belirli bir işlevi sağlarlar.
* Hizmetin işlevselliğine erişmek için bir uygulama programlama arayüzü (API) sağlarlar.
* Diğer bulut hizmetlerinde olduğu gibi, hizmeti kullanımınıza göre ücretlendirilirsiniz.

İşte Google Cloud'daki bazı özel amaçlı hizmetler:

* AutoML, bir makine öğrenimi hizmeti
* Cloud Natural Language, metin analizi için bir hizmet
* Speech-to-Text, konuşulan dili metne dönüştürme hizmeti
* Recommendations AI, kişiselleştirilmiş ürün önerileri

Özel amaçlı hizmetler, gelişmiş bilgi işlem yeteneklerini kapsar ve doğal dil işleme ve makine öğrenimi gibi alanlarda uzman olmayan geliştiricilerin erişimine sunar. Google Cloud'a daha fazla özel amaçlı hizmetin eklenmesini bekleyebilirsiniz.

# **Bulut Bilişim ve Veri Merkezi Bilişimi**

Bulutta sanal makineler çalıştırmanın, kendi veri merkezinizde çalıştırmaktan pek de farklı görünmese de, BT ortamlarını bulutta işletmek ile şirket içi veya ortak konumlandırılmış bir veri merkezinde işletmek arasında önemli farklılıklar vardır.

**Kaynaklara Sahip Olmak Yerine Kiralamak**

Kurumsal veri merkezleri sunucular, disk dizileri ve ağ ekipmanlarıyla doludur. Bu ekipman genellikle şirket tarafından uzun süreler boyunca sahip olunur veya kiralanır. Bu model, şirketlerin ekipman satın almak için önemli miktarda parayı peşin ödemesini veya ekipman için uzun vadeli bir kiralama sözleşmesi yapmasını gerektirir. Bu yaklaşım, bir kuruluşun uzun bir süre boyunca ihtiyaç duyacağı sunucu ve diğer ekipman sayısını doğru bir şekilde tahmin edebildiğinde ve bu ekipmanı tutarlı bir şekilde kullanabildiğinde iyi işler. Bu model, şirketlerin ortalama iş yükünden önemli ölçüde daha yüksek olan en yoğun kapasiteyi planlaması gerektiğinde o kadar iyi çalışmaz. Örneğin, bir perakendeci ortalama olarak 20 sunucudan oluşan bir küme gerektiren bir yüke sahip olabilir, ancak tatil sezonunda iş yükü 80 sunucuya ihtiyaç duyulacak noktaya kadar artar. Şirket, en yoğun kapasiteyi karşılayacak kaynaklara sahip olmak için 80 sunucu satın alabilir ve yılın büyük bir bölümünde 60 sunucuyu boşta bırakabilir. Alternatif olarak, daha az sunucu satın alabilir veya kiralayabilir ve bilgi işlem kaynakları talebe ayak uyduramayınca meydana gelecek iş kaybına katlanabilir. Her ikisi de çekici bir seçenek değildir. Genel bulutlar, bilgi işlem kapasitesinin kısa vadeli kiralanması şeklinde bir alternatif sunar. Örneğin, perakendeci, en yoğun dönemlerde şirket içi sunucularına ek olarak bulutta sanal makineler çalıştırabilir. Bu, perakendeciye ihtiyaç duyduğu anda ihtiyaç duyduğu sunuculara, ihtiyaç duyulmadığında ödeme yapmak zorunda kalmadan erişim sağlar.Bulutta sunucu çalıştırmanın birim maliyeti, veri merkezindeki eşdeğer bir sunucuyu çalıştırmanın maliyetinden daha yüksek olabilir, ancak şirket içi ve bulutta kısa vadeli sunucu karışımının toplam maliyeti, en yoğun kapasite için satın alma veya kiralama ve kaynakları boşta bırakma maliyetinden yine de önemli ölçüde daha az olabilir.

**Kullandıkça Öde Modeli**

Bulut bilişimin kısa vadeli kiralama modeliyle ilişkili olan, kullandıkça öde modelidir. Bulutta bir sanal sunucu çalıştırdığınızda, genellikle 1 dakika gibi minimum bir süre için ve sonrasında saniye başına ödeme yaparsınız. Saniye başına birim maliyet, sunucunun özelliklerine göre değişir. Daha fazla CPU ve belleğe sahip sunucular, daha az CPU ve daha az belleğe sahip sunuculardan daha pahalı olacaktır.

Bulut mühendislerinin, bulut sağlayıcılarının fiyatlandırma modelini anlaması önemlidir. Kullanımınızı izlemiyorsanız, sunucular ve depolama için büyük bir fatura oluşturmak kolaydır. Aslında, bazı bulut müşterileri, uygulamaları bulutta çalıştırmanın şirket içinde çalıştırmaktan daha pahalı olabileceğini görmektedir.

**Esnek Kaynak Tahsisi**

Şirket içi ve genel bulut bilişimi arasındaki bir diğer önemli fark, kısa sürede bilgi işlem ve depolama kaynaklarını ekleme ve çıkarma yeteneğidir. Bulutta, dakikalar içinde 20 sunucuyu başlatabilirsiniz. Şirket içi bir veri merkezinde, ek donanım sağlanması gerekiyorsa aynı şeyi yapmak günler veya haftalar sürebilir.

Bulut sağlayıcıları, veri merkezlerini kapsamlı bilgi işlem, depolama ve ağ kaynaklarıyla tasarlar. Bu kaynakları müşterilere verimli bir şekilde kiralayarak yatırımlarını optimize ederler. Müşteri kullanım alışkanlıkları hakkında yeterli veriyle, müşteri talebini karşılamak için ihtiyaç duydukları kapasiteyi tahmin edebilirler. Çok sayıda müşterileri olduğundan, herhangi bir müşterinin talebindeki değişiklik, kaynaklarının genel kullanımı üzerinde çok az etkiye sahiptir.Kapsamlı kaynaklar ve müşteriler arasında kaynakları hızlı bir şekilde kaydırma yeteneği, genel bulut sağlayıcılarının daha küçük veri merkezlerinde yapılabileceğinden daha verimli bir şekilde esnek kaynak tahsisi sunmasını sağlar.

**Özel Amaçlı Hizmetler**

Özel amaçlı hizmetler, doğaları gereği yaygın olarak anlaşılmaz. Birçok geliştirici kullanıcı arayüzleri geliştirmeyi veya bir veritabanını sorgulamayı bilir, ancak daha azı doğal dil işleme veya makine öğrenimi detaylarına maruz kalmıştır. Büyük işletmelerin veri bilimi ve makine görüşü gibi alanlarda şirket içi uzmanlık geliştirmek için finansal kaynakları olabilir, ancak birçok diğerinin yoktur.Bulut sağlayıcıları, özel amaçlı hizmetler sunarak, gelişmiş yetenekleri daha geniş bir geliştirici kitlesine sunmaktadır. Büyük miktarlarda donanıma yatırım yapmak gibi, genel bulut satıcıları da özel amaçlı hizmetlere yatırım yapabilir ve bu hizmetler çok sayıda müşteri tarafından kullanıldığı için maliyetlerini karşılayıp kar elde edebilirler.

**Özet**

Google Cloud, bilgi işlem, depolama, ağ ve özel amaçlı hizmetler için çeşitli hizmetler sunar. Bilgi işlem hizmetleri sanal makineleri ve Kubernetes kümelerini içerirken, depolama hizmetleri nesne ve dosya depolamayı önbellekleme ile birlikte destekler. Ağ hizmetleri, sanal özel bulutları ve VPN'ler, Bağlantılar Arası (Interconnects), Paylaşılan VPC (Shared VPC), VPC ağ eşlemesi ve Doğrudan veya Taşıyıcı eşlemesi dahil olmak üzere diğer hizmetleri sağlar. Özel amaçlı hizmetler arasında makine öğrenimi, Konuşmadan Metne ve öneri hizmetleri bulunur. Bulut bilişimin şirket içi bilişime göre çeşitli avantajları arasında altyapıya sahip olmak yerine kiralamak, kullandıkça öde modeli, esnek kaynak tahsisi ve özel amaçlı hizmetlerden faydalanmak yer almaktadır.

**Sınavın Püf Noktaları**

Bulut bilişim kaynaklarının sunulma yöntemlerini kavrayın. Bilgi işlem kaynakları, sizin yönettiğiniz bireysel sanal makineler (VM'ler) veya sanal makine kümeleri şeklinde tahsis edilebilir. Ayrıca, bir Kubernetes kümesini yönetmenin operasyonel yükünü azaltan yönetilen Kubernetes kümelerinden de yararlanabilirsiniz. Sunucusuz bilgi işlem seçenekleri ise kullanıcıları sunucu yönetimi zahmetinden kurtarır. Bunun yerine, geliştiriciler kodlarını bulut sağlayıcısı tarafından yönetilen konteynerleştirilmiş bir ortamda veya kısa süreli çalışan kod için özel olarak tasarlanmış bir bilgi işlem platformunda çalıştırırlar. Geliştiriciler ve DevOps uzmanları, kendi sunucularını ve kümelerini yönettiklerinde kaynaklar üzerinde en üst düzeyde kontrole sahip olurlar. Yönetilen hizmetler ve sunucusuz seçenekler, bilgi işlem ortamı üzerinde doğrudan kontrole ihtiyaç duymadığınız ve bilgi işlem kaynaklarını yönetme zorunluluğunun ortadan kalkmasının size daha fazla değer katacağı senaryolarda ideal tercihlerdir. Bulut depolamanın farklı biçimlerini ve bunların uygun kullanım alanlarını tekrar edin. Dört temel depolama kategorisi bulunmaktadır: nesne, dosya, blok ve bellek içi önbellekler. Nesne depolama, görüntüler veya veri kümeleri gibi nesnelerin yüksek güvenilirlikte ve dayanıklılıkta saklanması için özel olarak tasarlanmıştır. Nesne depolamanın dosya sistemi tabanlı depolama sistemlerine kıyasla işlevselliği daha sınırlıdır. Dosya sistemi tabanlı depolama ise dosyalar için hiyerarşik bir dizin yapısı sunar ve yaygın işletim sistemi ve dosya sistemi işlevlerini destekler. Dosya sistemi hizmetleri, birden çok sunucu tarafından erişilebilen ağ üzerinden erişilebilir dosya sistemleri sağlar. Blok depolama, verilerin diskler üzerinde saklanması için kullanılır. Dosya sistemleri ve veritabanları, blok depolama sistemlerinden faydalanır. Blok depolama, SSD'ler ve HDD'ler gibi kalıcı depolama aygıtlarıyla birlikte kullanılır. Önbellekler, veri alımındaki gecikmeyi en aza indirmek amacıyla kullanılan bellek içi veri depolarıdır. Kalıcı depolama sağlamazlar ve kesinlikle bir "doğruluk sistemi" olarak kabul edilmemelidirler.

Şirket içi ve bulut ortamlarında bir BT altyapısı çalıştırmanın temel farklılıklarına dikkat edin. Bir BT ortamını bulutta çalıştırmak, kaynakların kısa süreli kiralanması, kullandıkça öde modeli, esnek kaynak tahsisi ve özel amaçlı hizmetlerden yararlanma imkanı gibi çeşitli avantajlar sunar. Orta seviye bir sunucunun dakika başına maliyeti gibi bulut kaynaklarının birim maliyeti, şirket içi çözümlere kıyasla daha yüksek olabilir. Bulut sağlayıcınızın maliyet modelini anlamanız bu nedenle kritik öneme sahiptir; böylece iş yükünüzün bulut ve şirket içi kaynaklar arasında en verimli şekilde nasıl dağıtılacağına dair bilinçli kararlar verebilirsiniz.

Bölüm Değerlendirme Soruları

1. Aşağıdakilerden hangisi Google Cloud'da bir bilgi işlem kaynağı seçeneğidir?

A. Önbellek (Cache)

B. Sanal makine (VM)

C. Blok (Block)

D. Alt ağ (Subnet)

2. Bir bulut sağlayıcısı tarafından yönetilen bir küme kullanırsanız, aşağıdakilerden hangisi bulut sağlayıcısı tarafından sizin için yönetilir?

A. İzleme (Monitoring)

B. Ağ Kurulumu (Networking)

C. Bazı güvenlik yönetimi görevleri (Some security management tasks)

D. Yukarıdakilerin hepsi (All of the above)

3. Dosya işleme ve bir web sitesinin arka ucunu çalıştırmak için sunucusuz bilgi işlemeye ihtiyacınız var; Google Cloud'dan hangi iki ürünü seçebilirsiniz?

A. Kubernetes Engine ve Compute Engine

B. Cloud Run ve Cloud Functions

C. Cloud Functions ve Compute Engine

D. Cloud Functions ve Kubernetes Engine

4. Kullanıcıların büyük veri dosyalarını bir veri analitiği iş akışı tarafından analiz edilmek üzere yüklemesine olanak tanıyan bir web uygulaması için bir depolama sistemi tasarlamanız istendi. Dosyalar yüksek erişilebilirliğe sahip bir depolama sisteminde saklanmalıdır. Dosya sistemi işlevselliği gerekli değildir. Google Cloud'da hangi depolama sistemi kullanılmalıdır?

A. Blok depolama (Block storage)

B. Nesne depolama (Object storage)

C. Önbellek (Cache)

D. Ağ Dosya Sistemi (Network File System)

5. Tüm blok depolama sistemleri hangi blok boyutunu kullanır?

A. 4 KB

B. 8 KB

C. 16 KB

D. Blok boyutu değişebilir (Block size can vary)

6. Bir sanal özel bulutta ağ güvenliği kurmanız istendi. Şirketiniz birden çok alt ağa sahip olmak ve alt ağlar arasındaki trafiği sınırlamak istiyor. Alt ağlar arasındaki trafik akışını kontrol etmek için hangi ağ güvenliği kontrolünü kullanırsınız?

A. Kimlik erişim yönetimi (Identity access management)

B. Yönlendirici (Router)

C. Güvenlik duvarı (Firewall)

D. IP adresi tablosu (IP address table)

7. Tablolu verileri kullanarak nesneleri nasıl sınıflandıracağını öğrenen bir makine öğrenimi hizmeti oluşturduğunuzda, bilgi işlem kaynaklarını yönetmek için ne tür sunucular seçmelisiniz?

A. Sanal makineler (VM'ler).

B. VM kümeleri.

C. Sunucu yok; sunucusuz olan özel amaçlı hizmetleri kullanmalısınız.

D. Yalnızca Linux çalıştıran VM'ler.

8. Üç ila beş yıl gibi uzun süreler boyunca sunucu kullanmaya taahhüt etmek gibi, sunuculara yatırım yapmak ne zaman iyi sonuç verir?

A. Bir şirket yeni kurulurken

B. Bir şirket uzun bir süre boyunca sunucu ihtiyacını doğru bir şekilde tahmin edebildiğinde

C. Bir şirketin sabit bir BT bütçesi olduğunda

D. Bir şirketin değişken bir BT bütçesi olduğunda

9. Şirketiniz Kuzey Amerika'da bulunuyor ve faturaları toplu olarak işlemek için bir sanal sunucu çalıştıracak. Dakika başına birim maliyetini hangi faktör belirler?

A. Sanal makinenin (VM) çalıştırıldığı günün saati

B. Sunucunun özellikleri

C. Çalıştırdığınız uygulama

D. Yukarıdakilerin hiçbiri

10. Yakın gelecekte satış verilerini analiz etmek ve ürün talebini tahmin etmek için AutoML'yi kullanmayı planlıyorsunuz. Saat başına 1.000 ila 2.500 ürün arasında analiz yapmayı planlıyorsunuz. En yüksek talebi karşılamak için kaç VM ayırmalısınız?

A. 1

B. 10

C. 25

D. Hiçbiri; AutoML sunucusuz bir hizmettir.

11. Bir uygulamayı desteklemek için bir dizi hizmet çalıştırmanız gerekiyor. Aşağıdakilerden hangisi iyi bir dağıtım modelidir?

A. Büyük, tek bir VM üzerinde çalıştırın.

B. Yönetilen bir kümede konteynerler kullanın.

C. İki büyük VM kullanın, birini salt okunur yapın.

D. Tüm hizmetler için küçük bir VM kullanın ve CPU kullanımı yüzde 90'ı aştığında VM'nin boyutunu artırın.

12. Bir VM oluşturdunuz. Aşağıdaki sistem yönetimi işlemlerinden hangilerini üzerinde gerçekleştirmenize izin verilir?

A. Dosya sistemini yapılandırın.

B. İşletim sistemi yazılımını güncelleyin (yama uygulayın).

C. Dosya ve dizin izinlerini değiştirin.

D. Yukarıdakilerin hepsi.

13. Cloud Filestore hangi dosya sistemi teknolojisini temel alır?

A. Ağ Dosya Sistemi (NFS)

B. XFS

C. EXT4

D. ReiserFS

14. Google Cloud'da kaynak oluştururken, bu kaynaklar her zaman neyin parçasıdır?

A. Sanal özel bulut

B. Alt alan adı

C. Küme

D. Yukarıdakilerin hiçbiri

15. Bir uygulama için veri depolamanız gerekiyor ve bir önbellek kullanıyorsunuz. Önbellek veri alımını nasıl etkileyecektir?

A. Önbellek, istemci tarafı JavaScript'in yürütülmesini iyileştirir.

B. Önbellek, güç kesilse bile veri depolamaya devam ederek kullanılabilirliği artıracaktır.

C. Önbellekler, doğruluk sistemiyle senkronizasyonunu kaybedebilir.

D. Önbellek kullanmak gecikmeyi azaltacaktır, çünkü önbellekten almak SSD'lerden veya HDD'lerden almaktan daha hızlıdır.

16. Bulut sağlayıcıları neden esnek kaynak tahsisi sunabilir?

A. Bulut sağlayıcıları, düşük öncelikli müşterilerden kaynak alıp yüksek öncelikli müşterilere verebilir.

B. Kapsamlı kaynaklar ve müşteriler arasında kaynakları hızlı bir şekilde kaydırma yeteneği, genel bulut sağlayıcılarının esnek kaynak tahsisini daha küçük veri merkezlerinde yapılabileceğinden daha verimli bir şekilde sunmasını sağlar.

C. Ne kadar çok kaynak kullanırsanız o kadar çok ücret alırlar.

D. Sunamazlar.

17. Aşağıdakilerden hangisi Google Cloud'daki özel amaçlı hizmetlerin bir özelliği değildir?

A. Sunucusuzdurlar; sunucu veya küme yapılandırmanız gerekmez.

B. Metin çevirme veya görüntü analizi gibi belirli bir işlevi sağlarlar.

C. Kullanıcı tarafından izleme gerektirirler.

D. Hizmetin işlevselliğine erişmek için bir API sağlarlar.

18. Müşterinizin işlemleri, dosyaların belirli bölümlerine rastgele erişime izin veren bir VM'ye bağlı bir sürücüye erişmelidir. Bağlı sürücü ne tür bir depolama sağlar?

A. Nesne depolama

B. Blok depolama

C. NoSQL depolama

D. SQL depolama

19. Bir web uygulamasını desteklemek için yeni bir ilişkisel veritabanı dağıtıyorsunuz. Veritabanının veri dosyalarını depolamak için hangi tür depolama sistemini kullanırsınız?

A. Nesne depolama

B. Veri depolama

C. Blok depolama

D. Önbellek

20. Bir kullanıcı minimum kurulum gerektiren hizmetleri tercih ediyor; neden Cloud Storage, Cloud Run ve Cloud Functions'ı önerirsiniz?

A. Yalnızca zamana göre ücretlendirilirler.

B. Sunucusuzdurlar.

C. Kullanıcının VM yapılandırmasını gerektirirler.

D. Yalnızca Go dilinde yazılmış uygulamaları çalıştırabilirler.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Soru | A | B | C | D |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. B. Doğru cevap B'dir. Google Cloud'da bilgi işlem kaynakları satın almak için temel bir birim sanal makinedir (VM). A seçeneği yanlıştır; önbellek düşük gecikmeli bir depolama sistemidir. C seçeneği yanlıştır; blok, kalıcı disklerde bir depolama birimidir. D seçeneği yanlıştır; alt ağ bir ağ soyutlamasıdır.
2. D. Doğru cevap D'dir. Yönetilen kümeler kullanılırken, bulut sağlayıcısı kümedeki sunucuların (düğümler olarak da bilinir) sağlığını izleyecek; kümedeki düğümler arasında ağ bağlantısını kuracak ve güvenlik duvarı ile diğer güvenlik kontrollerini yapılandıracaktır.
3. B. Doğru cevap B'dir. Cloud Run, konteyner çalıştırmak için sunucusuz bir platformdur ve Cloud Functions, olaylara yanıt olarak kısa süreli çalışan işlevleri yürütmek için bir hizmettir. Kubernetes Engine yönetilen bir küme hizmetidir ve hem Kubernetes Engine hem de Compute Engine sunucu yapılandırmanızı gerektirir. Ne Compute Engine ne de Kubernetes sunucusuz bir seçenek değildir.
4. B. Doğru cevap B'dir. Cloud Storage gibi Nesne depolama, depolayabileceğiniz veri miktarında sınır olmaksızın yedekli olarak depolanmış nesneler sağlar, bu da B seçeneğini doğru kılar. Dosya sistemi işlevselliği gerekmediğinden, D seçeneği iyi bir seçenek değildir. Blok depolama kullanılabilir, ancak yüksek kullanılabilirlik sağlamak için kendi replikasyonunuzu yönetmeniz gerekirdi ve nesne depolamadan daha maliyetli olurdu. Önbellekler geçici, bellek içi depolamadır ve yüksek kullanılabilirlikli, kalıcı depolama sistemleri değildir.
5. D. Doğru cevap D'dir. Bir blok depolama sistemindeki blok boyutları değişebilir. Blok boyutu bir dosya sistemi oluşturulduğunda belirlenir. Linux'ta, 4 KB blok boyutları yaygın olarak kullanılır.
6. C. Doğru cevap C'dir. Google Cloud'daki güvenlik duvarları, bir ağa veya alt ağa giren ve çıkan trafik akışını sınırlayan yazılım tanımlı ağ kontrolleridir. Yönlendiriciler, trafiği ağdaki uygun hedeflere taşımak için kullanılır. Kimlik erişim yönetimi, kullanıcıları kimlik doğrulamak ve yetkilendirmek için kullanılır; alt ağlar arasındaki ağ kontrolleriyle ilgili değildir. IP adresi tabloları bir güvenlik kontrolü değildir.
7. C. C seçeneği doğrudur çünkü Google Cloud'daki AutoML gibi özel amaçlı hizmetler sunucusuzdur. Hizmetler tarafından kullanılan bilgi işlem kaynaklarını Google yönetir. Bir kullanıcının sunucu ayırmasına veya yönetmesine gerek yoktur.
8. B. B seçeneği doğrudur; sunuculara yatırım yapmak, bir kuruluşun uzun bir süre boyunca ihtiyaç duyacağı sunucu ve diğer ekipman sayısını doğru bir şekilde tahmin edebildiğinde ve bu ekipmanı tutarlı bir şekilde kullanabildiğinde iyi sonuç verir. Yeni kurulan şirketler, üç ila beş yıl içindeki beklenen ihtiyaçları yönlendirebilecek geçmişe sahip yerleşik işletmeler değildir. Bütçenin sabit veya değişken olması önemli değildir; sunuculara yatırım yapmak sunucu kapasitesi talebine dayanmalıdır.
9. B. Sanal sunucu sayısı, bellek miktarı ve VM'yi çalıştırdığınız bölge gibi sunucunun özellikleri maliyeti etkiler, bu nedenle B seçeneği doğrudur. Günün saati bir faktör değildir, aynı şekilde VM üzerinde çalıştırdığınız uygulama türü de değildir.
10. D. AutoML, Google Cloud'un özel amaçlı hizmetlerinden biridir. Hizmet kullanıcılarının hizmeti kullanmak için herhangi bir VM yapılandırmasına gerek yoktur. Bölüm 1: Google Cloud'a Genel Bakış 475
11. B. Konteynerler, bir kümenin kaynaklarını verimli bir şekilde kullanmak için en fazla esnekliği sağlar ve orkestrasyon platformları operasyonel yükü azaltır, bu da B seçeneğini doğru kılar. Tek bir VM üzerinde çalıştırmak önerilmez çünkü sunucu arızalanırsa tüm hizmetler kesilir. Birinin salt okunur olduğu iki VM kullanmak yararlı değildir. Salt okunur sunucular bazen veritabanlarıyla kullanılır, ancak soruda veritabanlarından bahsedilmemiştir. Küçük bir VM kullanmak ve iş yüküne ayak uyduramadığında yükseltmek, kullanıcılara düşük kaliteli hizmet sunar ve kaçınılmalıdır.
12. D. Doğru cevap D'dir. Bir VM oluşturulduktan sonra, tüm işlemler bir sistem yöneticisi tarafından gerçekleştirilebilir.
13. A seçeneği doğrudur; Cloud Filestore, dağıtılmış bir dosya yönetim sistemi olan Ağ Dosya Sistemi'ni (NFS) temel alır. Diğer seçenekler Linux tarafından desteklenen dosya sistemleridir.
14. Kaynak oluşturduğunuzda, bunlar bir VPC içinde oluşturulur. Kaynaklar VPC'ye eklenir ve açıkça yapılandırmadığınız sürece VPC dışından erişilemez. Alt alan adı web alanlarıyla ilgilidir ve Google Cloud kaynaklarının organizasyonuyla ilgili değildir. Kubernetes kümeleri gibi kümeler ağınızda olabilir, ancak tüm kaynaklar mutlaka bir kümede olmak zorunda değildir.
15. D. Doğru cevap D'dir. Önbellekler belleği kullanır ve bu da onları veri okuma için en hızlı depolama türü yapar. Önbellekler, dağıtılmış sistemlerin arka ucundaki veri depolarıdır, istemcilerin değil. Bir önbelleğin istemci tarafı JavaScript yürütmesi üzerinde hiçbir etkisi olmaz. Güç kesilirse önbellekler veri kaybedebilir ve verilerin yeniden yüklenmesi gerekir. Doğruluk sistemi güncellenebilir ancak önbellek güncellenmeyebilir, bu nedenle önbellekler doğruluk sistemiyle senkronizasyonunu kaybedebilir. Önbellekler, SSD'lerden ve HDD'lerden daha hızlı okuma sürelerine sahiptir.
16. B seçeneği doğrudur; bulut sağlayıcıları büyük kapasiteye sahiptir ve bu kaynakları farklı müşterilere hızlı bir şekilde tahsis edebilirler. Farklı müşteri ve iş yükleri karışımıyla, kaynak tahsisini optimize edebilirler. A seçeneği yanlıştır; bulut sağlayıcıları, öncelikli örnekler hariç, bir müşteriden kaynak alıp başka bir müşteriye vermezler. C seçeneği yanlıştır; bulut sağlayıcıları genellikle artan kullanım için indirim sunarlar.
17. C seçeneği doğrudur. Özel amaçlı hizmetler Google tarafından izlenir, bu nedenle kullanıcıların bunları izlemesine gerek yoktur. Özel amaçlı hizmetler belirli bir bilgi işlem işlevselliği sağlar ancak kullanıcının herhangi bir kaynak yapılandırmasını gerektirmez. Ayrıca API'ler sağlarlar.
18. B. Doğru cevap B'dir. Bağlı sürücüler blok depolama aygıtlarıdır. Cloud Storage bir nesne depolama hizmetidir ve doğrudan bir VM'ye bağlanmaz. NoSQL bir veritabanı türüdür, bir depolama sistemi değildir. SQL depolama diye bir şey yoktur; SQL, ilişkisel veritabanlarında kullanılan bir sorgulama dilidir.
19. Doğru cevap C'dir. Veritabanları blok aygıtlarında kalıcı depolama gerektirir. Nesne depolama, veri bloğu veya dosya sistemi depolaması sağlamaz. Veri depolama bir depolama sistemi türü değildir. Önbellekler genellikle okuma performansını artırmak için veritabanlarıyla birlikte kullanılır, ancak geçicidirler ve veri dosyalarını kalıcı olarak depolamak için uygun değildirler.
20. Doğru cevap B'dir. Üç hizmet de sunucusuzdur, bu nedenle kullanıcının VM yapılandırmasına gerek yoktur. Cloud Storage, depolanan verinin süresine ve boyutuna göre ücretlendirilir. Cloud Run ve Cloud Functions yalnızca Go diliyle sınırlı değildir.