

**Sözlük: OpenShift Temelleri**

Terim	Tanım
<b>A/B testi</b>	Yeni özellikleri ön uç uygulamalarında test etmek için kullanılan bir stratejidir. Uygulamanın A ve B olarak adlandırılan iki versiyonunu değerlendirerek, hangisinin kontrol edilen ortamda daha iyi performans gösterdiğini belirlemek için kullanılır. Uygulamaların iki versiyonu, özellikler açısından farklılık gösterir ve farklı kullanıcı gruplarına hitap eder. Kullanıcılardan alınan etkileşim ve geri bildirimlere dayanarak, küresel olarak üretime alınabilecek bir uygulama versiyonu seçebilirsiniz.
<b>Build</b>	Girdi kaynaklarının bir sonuç nesnesine dönüştürülmesi süreci.
<b>BuildConfig</b>	Bir build'in izlemesi gereken süreci tanımlayan OpenShift'e özgü bir nesne. Build süreci, girdi kaynaklarını ve build stratejisini kullanır. BuildConfig, bir plan ve build, o planın bir örneğidir.
<b>Canary Dağıtımları</b>	Uygulamanın yeni versiyonunu, kullanıcı sayısını kademeli olarak artırarak dağıtmayı amaçlar. Canary dağıtım stratejisi, yeni uygulama versiyonunu test etmek için gerçek kullanıcıları kullanır. Sonuç olarak, hatalar ve sorunlar, uygulamanın yeni versiyonu tüm kullanıcılar için küresel olarak dağıtılmadan önce tespit edilip düzeltilebilir.
<b>Circuit breaking</b>	Bir mikroservisteki hataların diğer mikroservislere yayılmasını önleyen bir yöntem.
<b>Yapılandırma Değişikliği</b>	Yeni bir BuildConfig kaynağı oluşturulduğunda yeni bir build'in çalışmasını tetikleyen bir olay.
<b>Kontrol Düzlemi</b>	Kontrol düzlemi, istenen yapılandırmayı ve hizmetlerin görünümünü alır ve ortam değişikçe dinamik olarak proxy sunucularını programlar ve günceller.
<b>Özel build stratejisi</b>	Kendi oluşturucu görüntünüzü tanımlayıp oluşturmanızı gerektirir.
<b>Özel oluşturucu görüntüleri</b>	Girdileri beklenen çıktıya dönüştürmek için gereken mantığı içeren normal Docker görüntüleridir.
<b>CRD'ler</b>	Tam bir özel sonucu inşa etmeden Kubernetes API sunucunuza eklemek için bir kaynağı tanımlayan özel kod.
<b>Özel denetleyiciler</b>	Özel kaynakların (CRD'ler) gerçek durumunu istenen durumu ile uzlaştırır.
<b>Veri düzlemi</b>	Hizmetler arasındaki iletişim veri düzlemi tarafından yönetilir. Bir hizmet ağı yoksa, ağ, akan trafiğin türünü, kaynağını ve varış noktasını tanımlayamaz ve gerekli kararları veremez.
<b>Uygulanabilirlik (Kontrol)</b>	Istio, tüm filosu boyunca politikaları uygulayarak kontrol sağlar ve kaynakların tüketiciler arasında adil bir şekilde dağıtılmasını garanti eder.
<b>Envoy proxy</b>	Tüm ağ trafiği, hizmet ağı tarafından kullanılan ve yapılandırmaya bağlı olarak birçok özelliği mümkün kılan Envoy adlı bir proxy tarafından etkilenir veya yakalanır.
<b>İnsan operatörleri</b>	Kontrol ettikleri sistemleri anlarlar. Hizmetleri nasıl dağıtacaklarını ve sorunları nasıl tanıyıp düzelteceklerini bilirler.
<b>Görüntü Değişikliği</b>	Yeni veya güncellenmiş bir görüntü mevcut olduğunda, konteynerleştirilmiş bir uygulamanın yeniden inşa edilmesini tetikleyen bir olaydır. Örneğin, bir uygulama Node.js tabanlı bir görüntü kullanılarak oluşturulmuşsa, o görüntü güvenlik düzeltmeleri yayımlandıkça ve diğer güncellemeler gerçekleştiğinde güncellenecektir.
<b>ImageStream</b>	OpenShift içinde konteyner görüntülerini referans almak için bir soyutlamadır. Her görüntü, onu tanımlayan bir ID veya digest içerir. ImageStreams, görüntü verisi içermez; bunun yerine görüntü digest'lerine işaret eder.
<b>ImageStream Etiketi</b>	Bir kayıt defterindeki belirli bir görüntüye işaret eden bir ImageStream'deki işareti tanımlar.
<b>Istio</b>	Kubernetes ile sıkça kullanılan platformdan bağımsız ve popüler bir hizmet ağı platformudur. Trafik ve API çağrılarının hizmetler arasında akışını akılcıca kontrol eder, çeşitli testler gerçekleştirir ve ağ hizmetlerini yönetmenin karmaşıklığını azaltır. Istio, kimlik doğrulama, yetkilendirme ve şifreleme yoluyla hizmetleri güvence altına alır. Istio, tüm filosu boyunca uygulanabilecek politikaları tanımlayarak kontrol sağlar. Istio ile ağınızdaki trafik akışını gözlemleyebilir, çağrı akışlarını, bağımlılıkları izleyebilir ve gecikme, trafik, hatalar ve doyma gibi hizmet iletişim metriklerini görüntüleyebilirsiniz.
<b>Adam Ortada Saldırıları</b>	Adam ortada (MiTM) saldırısı, saldırganın iki taraf arasında gizlice mesajları yakalayıp ilettiği bir siber saldırı türüdür. Saldırı, saldırganın tüm konuşmayı kontrol ettiği bir dinleme türüdür.
<b>Gözlemlenebilirlik</b>	Ağınızdaki trafik akışını gözlemlemenize, çağrı akışlarını ve bağımlılıkları izleyebilmenize, gecikme ve hatalar gibi metrikleri görüntülemenize yardımcı olur.
<b>OpenShift</b>	Bir hibrit bulut, kurumsal Kubernetes uygulamasıdır.
<b>OpenShift CI/CD süreci</b>	Yeni kod değişikliklerini otomatik olarak depoya birleştirir, inşa eder, test eder, onaylar ve farklı ortamlara yeni bir versiyon dağıtır.
<b>Operatörler</b>	Küme görevlerini otomatikleştirir ve Kubernetes API'sini genişletmek için özel bir denetleyici olarak hareket eder.
<b>Operatör Çerçevesi</b>	Verimli bir müşteri deneyimi sunmak için bir dizi araç ve yeteneklerdir. Sadece kod yazmakla kalmaz; Operatörlerin test edilmesi, dağıtılması ve güncellenmesi de kritik öneme sahiptir.
<b>OperatorHub</b>	Web konsolu, küme yöneticilerinin kümelerine kurmak için Operatörleri bulmalarını sağlar. Red Hat Operatörleri, Red Hat ile ortaklaşa çalışan bağımsız hizmet sağlayıcılardan Sertifikalı Operatörler, açık kaynak topluluğundan gelen ancak Red Hat tarafından resmi olarak desteklenmeyen Topluluk Operatörleri ve kullanıcılar tarafından tanımlanan özel Operatörler dahil olmak üzere birçok farklı türde Operatör sağlar.
<b>Operatör Yaşam Döngüsü Yöneticisi</b>	(veya OLM) Bir kümedeki Operatörlerin kurulumunu, yükseltmesini ve rol tabanlı erişim kontrolünü (veya RBAC) kontrol eder.

<b>Operatör olgunluk modeli</b>	Genel ikinci gün Operasyon faaliyetleri için olgunluk aşamalarını tanımlar ve Temel Kurulumdan Otonom Pilota kadar uzanır.
<b>Operatör Deseni</b>	Bir Denetleyiciyi bir veya daha fazla özel kaynakla bağlayan bir sistem tasarımıdır.
<b>Operatör Kaydı</b>	CRD'leri, küme hizmeti sürümlerini (CSV'ler) ve paketler ve kanallar için Operatör meta verilerini depolar. Operatör katalog verilerini OLM'ye sağlamak için Kubernetes veya OpenShift kümelerinde çalışır.
<b>Operatör SDK</b>	(Helm, Go ve Ansible'ı içerir) Yazarların, Kubernetes API karmaşıklığına dair bilgi gerektirmeden Operatörlerini oluşturmaya, test etmesine ve paketlemesine yardımcı olur.
<b>postCommit</b>	Bölüm, isteğe bağlı bir build kancası tanımlar.
<b>Tekrar Denemeler</b>	Başarısız bir isteği başka bir hizmete otomatik olarak yeniden denemek için kullanılan bir mekanizmadır.
<b>runPolicy</b>	Bir yapılandırmadan oluşturulan build'lerin nasıl çalışması gerektiğini kontrol eden alandır. Değerler, varsayılan Seri (sıralı) ve eşzamanlı gibi seçenekleri içerir.
<b>Hizmet Aracısı</b>	Yükseltmeler, devreye alma veya ölçeklendirme gibi ardışık günlerin işlemlerini gerçekleştiremeyen kısa süreli bir işlem sağlar.
<b>Hizmet Ağı</b>	Hizmetler arası iletişimi güvenli ve güvenilir hale getirmek için özel bir katmandır. Trafik akışını kontrol etmek için trafik yönetimi sağlar, hizmetler arasındaki trafiği şifrelemek için güvenlik sağlar ve hizmet davranışını gözlemlenebilir olanağı sunar; böylece uygulamaları sorun giderebilir ve optimize edebilirsiniz.
<b>Yazılım operatörleri</b>	İnsan operatörlerinin bilgilerini yakalamaya ve aynı süreçleri otomatikleştirmeye çalışır.
<b>Source-to-Image</b>	Tekrar üretilebilir konteyner görüntüleri oluşturmak için bir araçtır. Ayrıca S2i olarak kısaltılır, uygulama kaynak kodunu bir konteyner görüntüsüne enjekte ederek çalışmaya hazır bir görüntü üretir.
<b>Kaynak stratejisi</b>	Build'i yürütmek için kullanılan stratejiyi gösteren bölümdür; örneğin, Bir Kaynak, Docker veya Özel strateji gibi.
<b>Kaynak türü</b>	Ana girdi türünü belirler; bir Git deposu, bir satır içi Dockerfile veya ikili yükler gibi.
<b>Webhook</b>	OpenShift Container Platform API uç noktasına bir istek gönderen bir tetikleyicidir. Genellikle bu bir GitHub webhook'u olacaktır, ancak aynı zamanda genel bir webhook da olabilir. Bir GitHub webhook'u kullanılıyorsa, GitHub, belirli bir dalda yeni bir taahhüt olduğunda veya bir çekme isteği birleştirildiğinde veya birçok başka durumda OpenShift'e isteği gönderebilir. Webhook'lar, yeni kod geliştirildiğinde otomatik olarak derlemelerin gerçekleştirilmesi için geliştirme akışlarını otomatikleştirmenin harika bir yoludur.



# Skills Network