

Okuma: OpenShift'te Build Tetikleyicileri ile Yapıların Otomasyonu

Tahmini Süre: 20 dakika

Hedefler

Bu okumayı tamamladıktan sonra şunları yapabileceksiniz:

- OpenShift'teki build tetikleyicilerinin kavramını ve önemini anlamak.
- OpenShift'te mevcut olan farklı build tetikleyici türlerini tanımlamak ve açıklamak.
- Webhook tetikleyicilerinin nasıl çalıştığını ve ana özelliklerini açıklamak.
- Görüntü değişikliği tetikleyicilerinin sürecini ve faydalarnı tanımlamak.
- Konfigürasyon değişikliği tetikleyicilerinin işlevselliğini ve avantajlarını anlamak.

Yapı Tetiklerine Giriş

OpenShift'te, yapı tetikleri, yapı sürecini otomatikleştirmek için gereklidir ve uygulamaların sürekli olarak güncellenmesini ve verimli bir şekilde inşa edilmesini sağlar. Bu okumada, OpenShift'te mevcut olan farklı yapı tetiklerini keşfedeceğiz, her bir tetikleyiciyi detaylı bir şekilde açıklayacağız ve bunların akıcı bir yapı sürecine nasıl katkıda bulunduğu ele alacağız.

Build Tetikleyicilerini Anlamak

OpenShift'teki build tetikleyicileri, belirli olaylar veya koşullar temelinde bir build sürecini otomatik olarak başlatan mekanizmalardır. Bu tetikleyicileri kullanarak, geliştiriciler uygulamaları güncelleme ve dağıtım sürecini otomatikleştirip basitleştirebilir, bu da daha hızlı geliştirme döngüleri ve daha az manuel müdahale ile sonuçlanır.

Build Tetikleyici Türleri

OpenShift, ortamınızdaki farklı olaylara veya değişikliklere yanıt vermek üzere tasarlanmış çeşitli build tetikleyici türleri sunar. Ana build tetikleyici türleri şunlardır:

- Webhook Tetikleyicileri
- Görüntü Değişikliği Tetikleyicileri
- Yapılardırma Değişikliği Tetikleyicileri

Her türlü ayrıntılı olarak inceleyelim:

1. Webhook Tetikleyicileri

Webhook tetikleyicileri, HTTP istekleri aracılığıyla derlemelerin başlatılmasına olanak tanıyan güçlü bir özellikleir. Bu tetikleyiciler, belirli depo olaylarına dayalı olarak derlemeleri otomatikleştirmek için GitHub gibi dış sistemlerle entegrasyon sağlamak amacıyla yaygın olarak kullanılır.

Webhook Tetikleyicileri Nasıl Çalışır

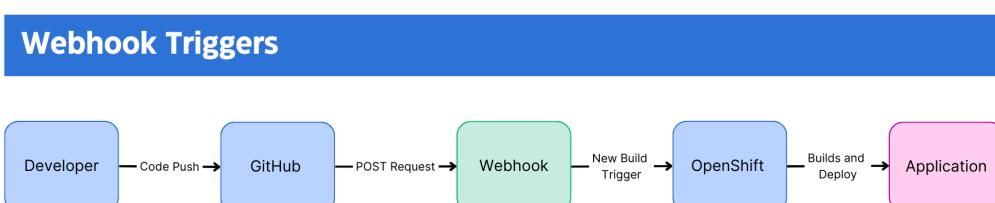
Bir webhook tetikleyicisi yapılandırıldığında, HTTP POST isteklerini dinleyen OpenShift'te bir uç nokta kurar. Örneğin, GitHub, bir depoya yeni bir commit gönderildiğinde, bir pull request birleştirildiğinde veya diğer belirlenen olaylar meydana geldiğinde bu uç noktaya bir istek gönderecek şekilde yapılandırılabilir. Bu istek, OpenShift'te build sürecini tetikler ve en son kod değişikliklerinin otomatik olarak derlenip dağıtılmmasını sağlar.

Ana Özellikler

- Git Depoları ile Entegrasyon:** GitHub, GitLab ve Bitbucket gibi popüler depolardan tetikleyicileri destekler.
- Olay Tabanlı:** Yapıları, taahhütler, birleşirmeler veya etiket oluşturma gibi çeşitli olaylara dayalı olarak başlatır.
- Esnneklik:** Hem genel hem de GitHub'a özgü web kancalarını destekleyerek farklı kullanım durumları için çok yönlü hale getirir.

Örnek İş Akışı

Aşağıda, bir GitHub webhook tetikleyicisinin nasıl çalıştığını gösteren temel bir iş akışı diyagramı bulunmaktadır:



İş Akışı Açıklaması:

- Bir geliştirici, GitHub deposuna yeni kod gönderir.
- GitHub, OpenShift webhook uç noktasına bir POST isteği gönderir.
- Webhook, OpenShift'te yeni bir derleme tetikler.

4. OpenShift, uygulamayı derler ve dağıtımını günceller.

2. GörSEL Değişiklik Tetikleyicileri

GörSEL Değişiklik tetikleyicileri, yeni bir konteyner görselinin versiyonu mevcut olduğunda otomatik olarak inşaları başlatır. Bu tür bir tetikleyici, en son bağımlılıklar veya güvenlik yamaları ile güncel uygulamaların sürdürülmesi için özellikle faydalıdır.

Görüntü Değişikliği Tetikleyicileri Nasıl Çalışır

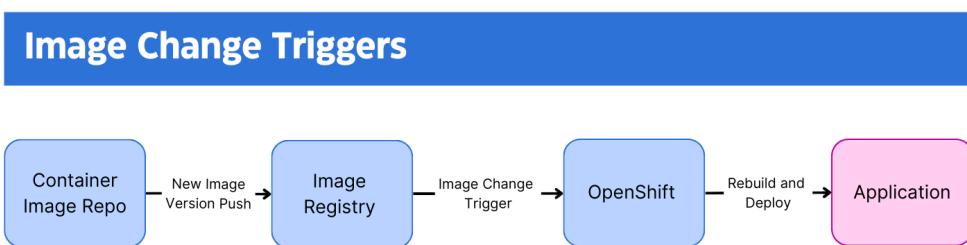
Bir görüntü değişikliği tetikleyicisi kurulduğunda, belirli bir konteyner görüntüsünü güncellemeler için izler. Örneğin, uygulamanız bir Node.js temel görüntüsüne dayanyorsa, bu temel görüntüdeki güncellemeleri tespit etmek için bir görüntü değişikliği tetikleyicisi yapılandırılabilir. Görüntünün yeni bir sürümü mevcut olduğunda, tetikleyici, güncellenmiş görüntüyü uygulamanıza dahil etmek için bir yapı sürecini başlatır.

Ana Özellikler

- Otomatik Bağımlılık Yönetimi:** Uygulamaların en son temel görüntüler veya kütüphanelerle otomatik olarak yeniden inşa edilmesini sağlar.
- Güvenlik ve Bakım:** Güncellenmiş temel görüntülerle inşa başlatabilir安全性と保守性を確保する。
- Sürekli Güncellemeler:** Uygulamaları bağımlı görüntülerdeki en son değişikliklerle güncel tutar.

Örnek İş Akışı

Bir görüntü değişikliği tetikleyicisinin nasıl çalıştığını gösteren basit bir iş akışı diyagramı:



İş Akışı Açıklaması:

- Temel görüntünün (örneğin, Node.js) yeni bir versiyonu görüntü kayıt defterine itilir.
- Görüntü değişikliği tetikleyicisi güncellemeye alılgılar.
- OpenShift, güncellenmiş temel görüntüyü kullanarak yeni bir derleme süreci başlatır.
- Uygulama, en son görüntü ile yeniden derlenir ve yeniden dağıtılr.

3. Yapılandırma Değişikliği Tetikleyicileri

Yapılandırma değişikliği tetikleyicileri, yeni bir BuildConfig kaynağı oluşturulduğunda veya mevcut bir tanım değiştirildiğinde yapıları başlatır. Bu tetikleyiciler, yapılandırma değişikliklerinin otomatik olarak yapıların yansıtılmasını sağlar; örneğin, kaynak kodu havuzlarındaki güncellemeler veya yapı stratejilerindeki değişiklikler gibi.

Yapılandırma Değişikliği Tetikleyicileri Nasıl Çalışır

Yapılandırma değişikliği tetikleyicileri, OpenShift'teki BuildConfig kaynaklarını izler. Yeni bir BuildConfig oluşturulduğunda veya mevcut bir tane güncellendiğinde, tetikleyici otomatik olarak yeni bir yapı başlatır. Tetikleyiciler, yapılandırmadaki herhangi bir değişikliğin uygulanmasına hemen uygulanmasını sağlar ve uygulamanın en son yapılandırma ayarlarıyla senkronize kalmasını garanti eder.

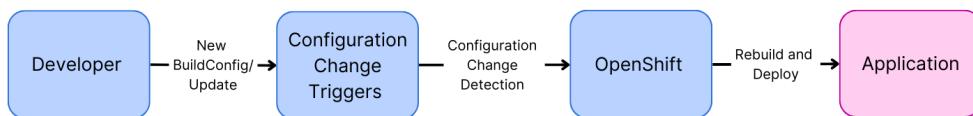
Anahtar Özellikler

- Otomatik Yapı Güncellemeleri:** Yapıların en son yapılandırma değişikliklerini yansıtmasını sağlar, manuel müdahale olmadan.
- Basitleştirilmiş Yapılandırma Yönetimi:** Yapılandırma yönetimini ve uygulanmasını verimli bir şekilde sağlar.
- Esneklik:** Kaynak depolarındaki güncellemeler, yapı stratejileri veya çıktı ayarları gibi geniş bir yapılandırma değişikliği yelpazesinin yapıları tetiklemesine olanak tanır.

Örnek İş Akısı

Aşağıda bir yapılandırma değişikliği tetikleyicisinin nasıl çalıştığını gösteren temel bir iş akışı diyagramı bulunmaktadır:

Configuration Change Triggers



İş Aķışı Açıklaması:

1. Yeni bir BuildConfig kaynağı oluşturulur veya mevcut olan güncellenir.
2. Yapılandırma değişikliği tetikleyicisi değişikliği algılar.
3. OpenShift, güncellenmiş BuildConfig'e dayalı olarak yeni bir yapı süreci başlatır.
4. Uygulama, yeni yapılandırma ile yeniden inşa edilir ve yeniden dağıtılr.

Sonuç

OpenShift'teki yapı tetikleyicileri, yapı ve dağıtım sürecini otomatikleştirmek için sağlam bir mekanizma sağlar. Webhook, görüntü değişikliği ve yapılandırma değişikliği tetikleyicilerini kullanarak, uygulamalarınızın en son değişiklikler ve olaylar temelinde sürekli olarak oluşturulmasını ve güncellenesini sağlayabilirsiniz. Bu otomasyon, daha verimli geliştirme döngüleri, güncellemelere veya sorunlara daha hızlı yanıt süreleri ve azaltılmış manuel yük ile sonuçlanır. Böylece, nihayetinde daha düzenli ve güvenilir bir geliştirme iş akışına katkıda bulunur.

Yazar:

Rajashree Patil



Skills Network