LitmusChaos Kaos Aracı

Fizibilite Raporu

İÇİNDEKİLER

LitmusChaos Kaos Aracı Fizibilite Raporu

1. Giriş

Bu fizibilite raporu, LitmusChaos Kaos aracının potansiyel bir ürün veya hizmet olarak piyasaya sürülmesi için gerekli olan analizleri sunmaktadır. Litmus Kaos aracı, çeşitli işletmelerin karmaşık ve belirsiz durumlarla başa çıkmasına yardımcı olacak bir yazılım platformu olarak tasarlanmıştır. Bu rapor, pazar araştırması, teknik değerlendirme, mali analiz ve risklerin incelenmesi gibi çeşitli alanları içerecektir.

1.1 Kaos Mühendisliği Nedir?

Kaos mühendisliği (Chaos Engineering), dağıtılmış bilgi işletim sisteminin beklenmedik kesintilere dayanabilmesini sağlamak için test etme sürecidir. Rastgele ve beklenmeyen davranış ve tutumlara odaklanan kaos teorisindeki kavramlara dayanır. Bir başka ifadeyle, kaos mühendisliğinin amacı, kontrollü deneyler yaparak rastgele ve öngörülemeyen davranışları ortaya çıkararak bir sistemdeki zayıflığı tespit etmektedir.

1.2 Neden Kaos Mühendisliği Araçları Kullanılmalı?

Kaos mühendisliği araçları, sistemlere güven oluşturmak için kullanılan geleneksel test yöntemlerine nispeten yeni bir yaklaşımdır. Yazılım platformları kaçınılmaz olarak başarısız olacaktır ve bu nedenle zayıf noktaları tespit etmek ve iş operasyonlarını olumsuz etkilemeden önce bunları düzeltmek kritik öneme sahiptir.

Amazon, Netflix ve Microsoft gibi önde gelen teknoloji kuruluşları, iç sistematik davranışları ve kusurları daha iyi anlamak için kaos mühendisliğini kullanıyor. Bu yaklaşımın ilkeleri, sistem mimarilerinin çeşitli hipotezler ve performansa dayalı ölçümler yoluyla test edilmesi fikrine dayanmaktadır. Kaos mühendisliği araçları, varsayımların ve başarılı kaos deneylerinin uygulanması yoluyla altyapı arızalarının veya yanıt vermeyen sistemlerin ortaya çıkarılması için bir yol haritası sağlayabilir.

Kaos mühendisliği, aşağıdaki adımların her birini içeren bir dizi genel yönergeyi izler:

Kararlı durum hipotezi oluşturma: Ortaya çıkabilecek potansiyel sistem sorunlarını düşünün. Arıza enjeksiyon kaos test protokollerini ayarlayın ve çeşitli potansiyel sonuçları tahmin edin.

Gerçek dünya senaryolarını simüle edin: Sistemlerin farklı değişkenlere nasıl tepki vereceğini belirleyecek bir dizi test oluşturun. Çeşitli koşulları ve faktörleri test etmek için bir deney grubu kullanın.

Sistem metriklerini gözden geçirin: Sistem performansı ve metrikleriyle ilgili sistem sonuçlarını inceleyin. Hipotezlere göre başarısızlık oranlarını belirleyin ve yinelenen sorunları düzeltmek ve düzeltmek için ileriye dönük bir yol bulun.

Gerektiğinde değişiklikleri uygulayın: Kaos deneylerinin sonuçlanmasının ardından, en iyi eylem planının ne olduğunu tespit edebilmelisiniz. Sorunları gidermeye çalışın ve sistemler çok az veya hiç hata olmadan çalışana kadar işlemi tekrarlayın.

Kaos deneylerini otomatikleştirin: Sisteminizin arıza moduna karşı dayanıklı olduğu doğrulandıktan sonraki adım, ortamda meydana gelen dinamik sistem yapılandırma değişiklikleri yoluyla sürekli doğrulamayı sağlamak için kaos deneyleri oluşturmak ve bunları yazılım teslim hattınızda otomatikleştirmektir. Deneme başarısız olursa, bu konuda bilgilendirilebilir veya başarısızlığa yol açan bir değişikliğin geri alınmasını otomatikleştirebilirsiniz. Sisteminiz, korunduğunuz bilinen bir arıza modundan dolayı kesinti yaşamayacaktır.

1.3 LitmusChaos Kaos Aracı

LitmusChaos, ekiplerin kaos testlerini kontrollü bir şekilde başlatarak altyapılardaki zayıflıkları ve potansiyel kesintileri belirlemesine olanak tanıyan açık kaynaklı bir Kaos Mühendisliği platformudur. Geliştiriciler ve SRE'ler, modern Kaos Mühendisliği ilkelerine ve topluluk işbirliğine dayalı, kullanımı kolay olduğundan LitmusChaos ile Kaos Mühendisliği uygulayabilir. %100 açık kaynak ve bir CNCF projesidir. LitmusChaos, kaosu oluşturmak, yönetmek ve izlemek için bulutta yerel bir yaklaşım benimsiyor. Platformun kendisi bir dizi mikro hizmet olarak çalışıyor ve kaos amacının yanı sıra kararlı durum hipotezini tanımlamak için Kubernetes özel kaynaklarını (CR'ler) kullanıyor.

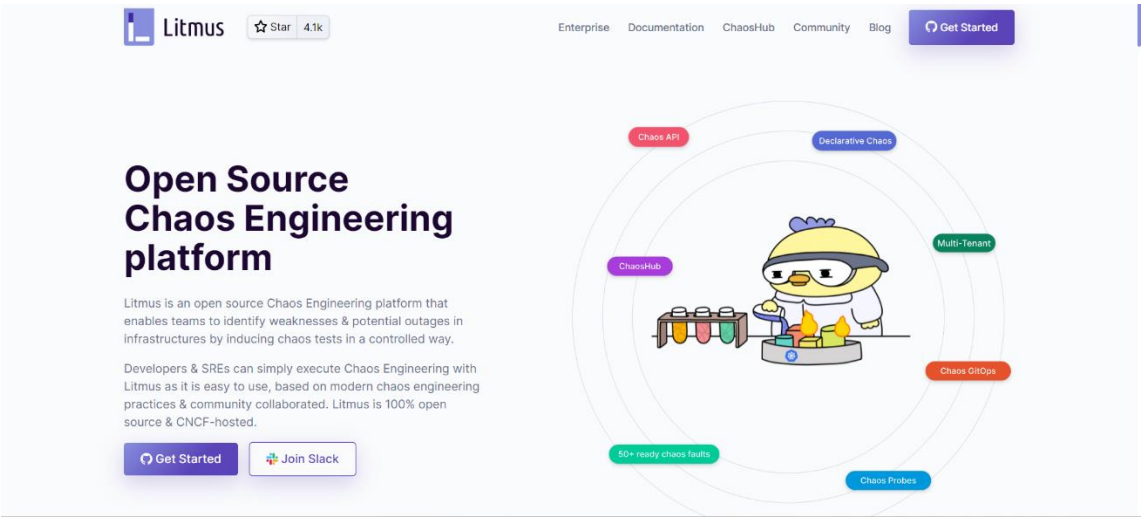
2. Pazar Araştırması

İşletmeler, çeşitli nedenlerden dolayı giderek artan bir şekilde geleneksel dağıtım yöntemleri yerine bulut tabanlı dağıtımlara (yani Kubernetes tabanlı olanlara) yöneliyor; bunlardan biri dağıtım hızını artırma ihtiyacıdır. Site güvenilirliği mühendislerinin (SRE'ler) ve geliştirme ekiplerinin şu anda karşılaştığı zorluk, bulutta yerel sistemlerin geleneksel dağıtımlardan daha fazla şekilde başarısız olabilmesidir.

Planlanmayan kesintilerin ticari açıdan, marka açısından ve itibar açısından önemli etkileri olabilir. Planlanmamış kesinti sürelerinin maliyetleri ve sistem düzeyindeki karmaşıklıktaki bu artış, bulutta yerel sistemleri test etme yöntemimizi geliştirme konusunda artan bir ihtiyaç yarattı. Kaos mühendisliği araçlarından olan LitmusChaos Kaos aracı, zayıf noktaları ortaya çıkarmak için sistem düzeyinde yazılım testinin gerçekleştirildiği mekanizmayı sağlar ve ekiplerin daha güvenilir sistemler sunmasına yardımcı olur.

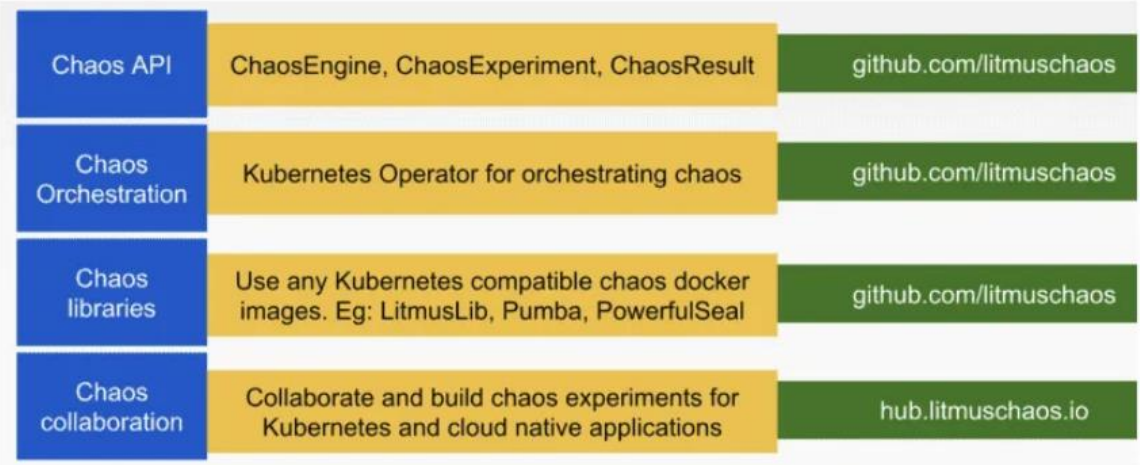
3. Teknik Değerlendirme

LitmusChaos Kaos aracının kullanıcı arayüzü:



3.1 LitmusChaos Aracı Bileşenleri

LitmusChaos aracı 4 bileşenden oluşuyor.



3.2 LitmusChaos Aracı API’leri

LitmusChaos mevcutta 3 API sağlamakta; **ChaosEngine, ChaosExperiment** ve **ChaosResult.**

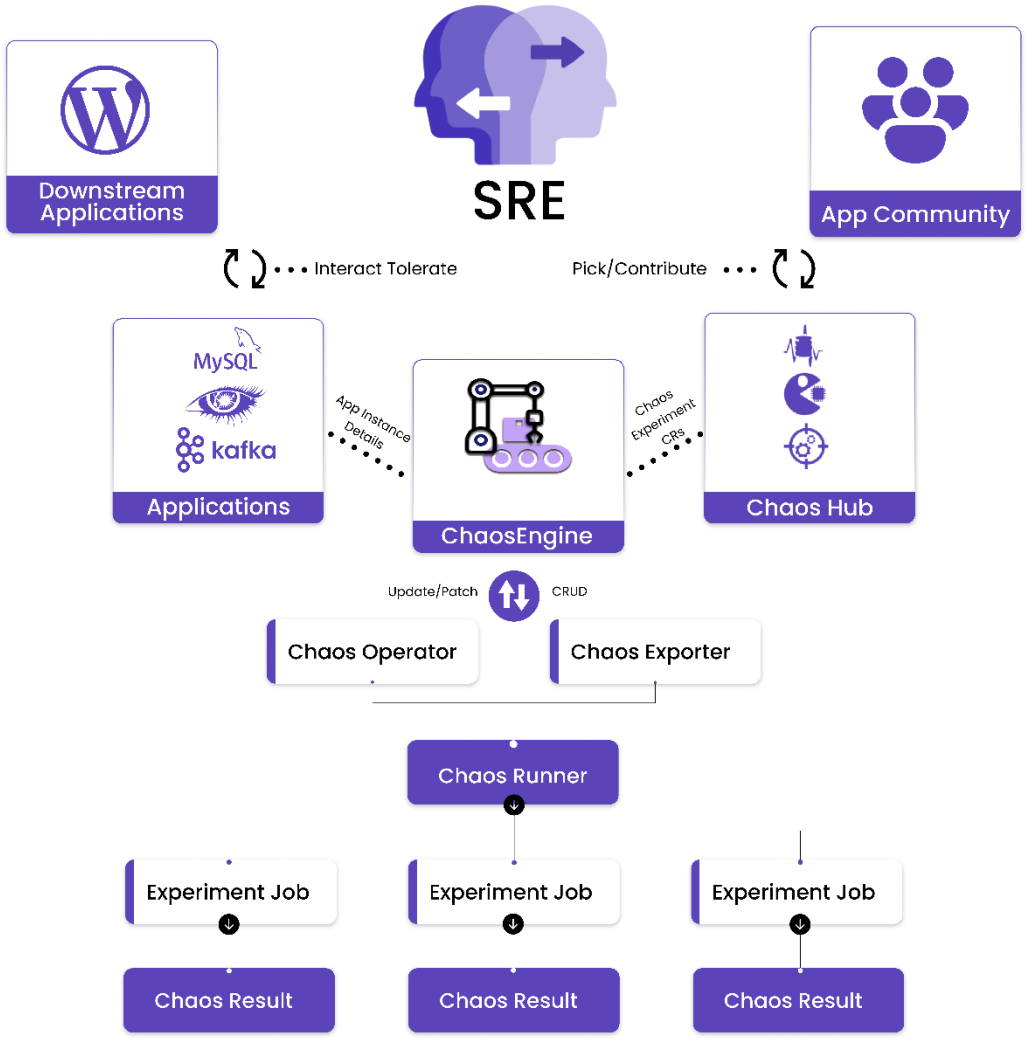
ChaosExperiment : Belirli bir hatanın yapılandırma parametrelerini gruplandıran bir kaynak. ChaosExperiment CR'leri esasen, hatayı gerçekleştiren kitaplığı tanımlayan, onu çalıştırmak için gereken izinleri ve birlikte çalışacağı varsayılanları belirten kurulabilir şablonlardır. Litmus, ChaosExperiment aracılığıyla, hata enjeksiyonunu gerçekleştirmek için herhangi bir üçüncü taraf aletin entegre edilmesine (isteğe bağlı) yardımcı olan BYOC'yi (bring-your-own-chaos) destekler.

ChaosEngine : Bir Kubernetes uygulaması iş yükünü/hizmetini, düğümünü veya bir altyapı bileşenini ChaosExperiment tarafından açıklanan bir hataya bağlayan bir kaynak. Ayrıca çalıştırma özelliklerini ayarlama ve 'probes' kullanarak kararlı durum doğrulama kısıtlamalarını belirleme seçenekleri de sağlar. ChaosEngine, koşucular aracılığıyla onu uzlaştıran (deneyin yürütülmesini tetikleyen) Kaos Operatörü tarafından izlenir.

ChaosExperiment ve ChaosEngine CR'leri, bir veya daha fazla deneyi istenen sırayla bir araya getirebilen bir İş Akışı nesnesinin içine yerleştirilmiştir.

ChaosResult : Deneme çalıştırmasının sonuçlarını tutacak bir kaynak. Her bir doğrulama kısıtlamasının başarısı, hatanın geri alma/geri alma durumu ve ayrıca bir karar hakkında ayrıntılı bilgi sağlar. Kaos ihracatçısı sonuçları okur ve bilgileri prometheus ölçümleri olarak ortaya çıkarır. ChaosResults özellikle otomatik çalıştırmalar sırasında kullanışlıdır.

ChaosExperiment CR'leri hub.litmuschaos.io'da barındırılır . Uygulama geliştiricilerinin veya satıcılarının, kullanıcılarının üretimdeki uygulamaların dayanıklılığını artırmak için kullanabilmeleri için kaos deneylerini paylaştığı merkezi bir merkezdir.



4.Risklerin Değerlendirilmesi

LitmousChaos aracının avantajları ve dezavantajları şunlardır:

Avantajları:

- Çeşitli deneyleri içeren merkezi depo
- Tekrarlanan sistem sağlık kontrolleri
- Otomatik hata tespiti ve dayanıklılık puanları
- Hem kaos hem de fonksiyonel testleri gerçekleştirme yeteneği.
- Kullanıcıların test paketlerini çalıştırmasına, günlük yakalama gerçekleştirmesine ve raporlar oluşturmasına olanak tanır.
- Bir deneme öncesinde, sırasında ve deney tamamlandıktan sonra uygulamanın durumunu izleme yeteneği.

Dezavantajları:

- LitmusChaos ile başlamak kullanıcının geçmişine bağlı olarak zor olabilir.
- Karmaşık yönetim görevleri, her ad alanı için hizmet hesaplarının ve ek açıklamaların ayarlanmasını gerektirir.
- İzinlerin yönetilmesi ve izlenmesi zor olabilir.

4.1 Diğer Kaos Araçlarına Göre Avantaj ve Dezavantajları

Diğer kaos mühendisliği araçlarına göre avantajları ve dezavantajları şu şekilde sıralanabilir:

Avantajlar:

Kubernetes Odaklı Olması: LitmusChaos, özellikle Kubernetes ortamları için tasarlanmıştır ve Kubernetes kaynakları üzerinde çalışmak için özel olarak geliştirilmiştir. Bu, Kubernetes tabanlı altyapılar için daha uygun ve etkili bir çözüm sağlar.

Genişletilebilirlik: LitmusChaos, modüler bir yapıya sahiptir ve özelleştirilebilir eylemler ve senaryolar eklemek için kolayca genişletilebilir. Bu, kullanıcıların ihtiyaçlarına ve ortamlarına uyacak şekilde aracı özelleştirmelerine olanak tanır.

Topluluk Desteği: LitmusChaos, geniş bir topluluk tarafından desteklenmektedir. Bu, kullanıcıların sorunlarını çözmek için kaynaklara ve desteklere daha kolay erişim sağlar.

Dokümantasyon ve Örnekler: LitmusChaos'un kapsamlı dökümantasyonu ve örnek senaryoları, kullanıcıların başlamasını kolaylaştırır ve aracı etkili bir şekilde kullanmalarına yardımcı olur.

Dezavantajlar:

Yenilikçi Olarak Geliştirme: LitmusChaos, diğer bazı kaos mühendisliği araçlarına kıyasla daha yenidir ve bu nedenle bazı olgunluk eksiklikleri veya eksik özellikler olabilir.

Belirli Senaryoların Eksikliği: Diğer bazı kaos araçlarına göre, LitmusChaos'un bazı spesifik senaryolar veya eylemler için eksiklikleri olabilir. Bu, bazı kullanıcıların ihtiyaçlarını tam olarak karşılamayabilir.

Performans ve Kararlılık: Herhangi bir yazılım gibi, LitmusChaos'un performans ve kararlılık sorunları olabilir. Büyük ve karmaşık Kubernetes ortamlarında kullanıldığında bu sorunlar daha belirgin olabilir.

Kurulum ve Yapılandırma Karmaşıklığı: LitmusChaos'un kurulumu ve yapılandırılması, kullanıcılar için karmaşık olabilir. Özellikle yeni başlayanlar için bu süreç zorlayıcı olabilir.

5.Sonuç

LitmusChaos, docker konteynerlerinin test edilmesinden belirli Pod'lara kadar çeşitli deneyleri kolaylaştıran Kubernetes'e özgü bir araçtır. Prometheus aracılığıyla çeşitli izleme yeteneklerine sahip çok yönlü bir araç olan LitmusChaos faydalıdır ancak başlamadan önce önemli düzeyde bilgi gerektirir.

6. Kaynakça

- https://www.harness.io/blog/chaos-engineering-tools
- https://medium.com/t%C3%BCrk-telekom-bulut-teknolojileri/chaos-engineering-litmuschaos-f8744c062b09
- https://github.com/litmuschaos/litmus?tab=readme-ov-file
- https://litmuschaos.io/
- https://medium.com/t%C3%BCrk-telekom-bulut-teknolojileri/kaos-m%C3%BChendisli%C4%9Fi-nedir-2a7a9e11a273