



**T. C.
RECEP TAYYİP ERDOĞAN ÜNİVERSİTESİ
MİMARLIK VE MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**İzleme ve Kayıtlama
Sistemleri**

Grup Üyeleri

Alper ŞAHİN – 201401060

Hüseyin YAŞAR – 201401014

Rıdvan KARASUBAŞI – 201401030

Sema ŞAHİN – 201401062

Yıldıray KARASUBAŞI – 201401031

Dersin Sorumlusu

Dr. Mehmet SEVRİ

2024

Önsöz

Teknolojinin hızla ilerlediği bu çağda, sistemlerin sürekli olarak izlenmesi ve kayıtlarının tutulması, verimliliğin ve güvenliğin korunmasında hayati bir rol oynamaktadır.

Bu kitap, IT profesyonelleri, sistem yöneticileri ve teknoloji meraklıları için değerli bir kaynak olma potansiyeline sahiptir. Sistem izleme ve kayıtlarının neden bu kadar önemli olduğunu ve nasıl doğru bir şekilde uygulanacağını öğrenmek isteyen herkes için bu kitap bir rehberdir.

İçindekiler

1. Sistem kayıtlama (logging) nedir?	6
1.1. Sistem Kayıtları (Log) Neden Önemlidir?	6
1.2. Kayıt Dosyaları Nereden Geliyor?	9
1.3. Kayıt (Log) Türleri	9
1.4. Linux Sistem Kayıtları.....	11
1.5. Linux Sistem Kayıt (Log) Dosyaları	12
1.6. Windows Sistem Kayıtları (Log).....	17
1.7. Windows Kayıt (Log) Kategorileri.....	17
1.8. Windows Sistem Kayıtları (Log) Önem Düzeyi.....	19
1.9. Windows Sistem Kayıtları (Log) Öğeleri.....	21
1.10. Windows Olay Görüntüleyicisi (Event Viewer) Nasıl Açılır?	22
2. Monitoring nedir?.....	24
2.1. Monitoring Yapmanın Başlıca Faydaları Nelerdir?.....	24
2.2. Prometheus Tarihçesi	24
2.3. Prometheus Nedir? Ne İçin Kullanılır?	25
2.4. Prometheus'un Kullanım Alanları Nelerdir?	26
2.5. Prometheus Metrik Türleri	27
2.6. Uygulama Metriklerini Almazsak Ne Olur?.....	28
2.7. Prometheus Kurulumu	29
2.8. Putty ile Sanal Sunucuya Bağlanma	33

2.9.	Sanal Sunucuya Prometheus Kurma.....	36
2.10.	Sanal Makineye Node_Exporter Kurma.....	41
2.11.	Farklı Bir Sanal Makineye Node_Exporter Kurma	44
2.12.	Premetheus Exporter Nedir?.....	47
2.13.	Prometheus Mimarisi.....	49
2.14.	Prometheus Verileri Nasıl Analiz Eder	55
2.15.	Prometheus Rules Nedir?	56
2.16.	Grafana Nedir?	58
2.17.	Grafana'nın Özellikleri Nelerdir?.....	58
2.18.	Docker Nedir?.....	60
2.19.	Grafana ve Docker Kurulumu	61
2.20.	Alert Manager Nedir?	70
2.21.	Alert Manager Kurulumu	70
2.22.	Alert Manager İle E-Posta Bildirimi Almak.....	75
8.	ELK Stack Nedir?	79
8.1.	ElasticSearch'e Giriş	79
8.2.	ElasticSearch Bileşenleri	79
8.3.	ElasticSearch Kurulumu	81
8.4.	Kibana ve Kibana Kurulumu	83
8.5.	Logstash Nedir ve Logstash Kurulumu?	85
8.6.	Beats	89
8.7.	Filebeat	89

8.8.	Winlogbeat.....	89
8.9.	Metricbeat.....	90
2.10.	Packetbeat.....	90
2.11.	Auditbeat	90
2.12.	Heartbeat.....	90
3.	Uygulama	91
3.1.	Senaryo	91
3.2.	Winlogbeat Kurulumu:	91

1. SİSTEM KAYITLAMA (LOGGING) NEDİR?

Sistem kayıtları, sistemlerde oluşan her türlü olayın kaydedilmesi ve belgelenmesi işlemidir. Bu, sistemde meydana gelen herhangi bir etkinliğin veya hata, uyarı, bilgi, ya da herhangi bir kullanıcı etkileşimi gibi olayların detaylı bir şekilde kaydedilmesini içerir. Örneğin, bir kullanıcının web sitesine erişimini düşünelim. Bu erişim olayı, sistem kayıtları tarafından kaydedilecek bir olaydır. Kayıt (Log) dosyasında, kullanıcının IP adresi, erişim zamanı, erişilen sayfa veya kaynak, tarayıcı bilgileri gibi bilgiler kaydedilir. Bu kayıtlar, kullanıcıların siteye nasıl eriştiğini, hangi sayfaları ziyaret ettiklerini ve hangi tarayıcıları kullandıklarını anlamak için değerli bir kaynak sağlar.

Günümüzde sistem kayıtlarının tutulması artık devletler tarafından zorunlu hale getirilmektedir. Ayrıca 5651 Sayılı Kanun ile ISO 27001 Bilgi Güvenliği Yönetim Sistemi Standardı ve KVKK (Kişisel Verileri Koruma Kanunu), kayıt (log) tutulmasını zorunlu hale getirmiştir.

1.1. Sistem Kayıtları (Log) Neden Önemlidir?

Sistem kayıtlama, bir bilgisayar sistemindeki faaliyetleri ve olayları kaydetme sürecidir. Bu süreç, çeşitli amaçlar için önemlidir ve bir dizi nedenle zorunlu hale gelmiştir. Sistem kayıtları, sistemin performansını izleme, sorun giderme, güvenlik analizi, kapasite planlama ve uyumluluk gibi alanlarda kritik bir rol oynar. İşte bunun nedenleri ise şu şekildedir.

- 1. Güvenlik Analizi:** Güvenlik analizi için sistem kayıtları oldukça kritiktir çünkü bir sistemdeki güvenlik tehditlerini belirlemek ve bunlara yanıt vermek için gereken bilgiyi sağlarlar. Örneğin, kullanıcıların sisteme başarılı veya başarısız

oturum açma denemelerini kaydeder. Bir kullanıcının birden fazla kez hatalı parola girişi yaptığını veya yetkisiz bir kullanıcı tarafından yapılan oturum açma denemelerini gösterebilir. Bu tür etkinlikler, şüpheli sistem etkinliklerinin belirlmesine olanak tanır.

2. **Performans İzleme:** Kayıtlar, sistem bileşenlerinin (CPU, bellek, disk, ağ vb.) performansını izlemek için önemli veriler sağlar. Bu veriler, sistem kaynaklarının nasıl kullanıldığını, performans sorunlarının nereden kaynaklandığını ve iyileştirmelerin nerede yapılması gerektiğini anlamak için kullanılır. Kayıtların düzenli olarak analiz edilmesi, tipik çalışma kalıplarını ve davranışları belirlememize ve sistemdeki herhangi bir anormalliği tanımlamamıza olanak tanır. Örneğin, CPU kullanımında anormal bir artış, bir uygulamanın performans sorunlarına veya bir saldırı girişimine işaret edebilir.
3. **Denetim ve Uyumluluk:** İşletmeler için denetim ve uyumluluk, kritik öneme sahip bir konudur. Birçok sektörde, işletmelerin belirli yasal düzenlemelere uygun olmaları gerekmektedir. Bu düzenlemeler genellikle güvenlik, veri koruma ve kişisel gizlilik gibi konuları kapsar. Sistem kayıtları, kullanıcı ve sistem etkinliklerini ayrıntılı olarak kaydettiği için, işletmelerin bu tür düzenlemelere uyum sağlamalarına yardımcı olur. Örneğin, GDPR (General Data Protection Regulation) gibi düzenlemeler, kişisel verilerin nasıl işlendiğini ve korunduğunu belirler. Kayıtlar, bu tür uyumluluk gereksinimlerini karşılamak için önemli bir gerektir.
4. **Kaynak Yönetimi:** Sistem kayıtları CPU, bellek, disk ve ağ kullanımını gösterir. Yöneticiler bu kayıtları analiz ederek yoğun kaynak kullanan işlemleri bulabilir ve kapasiteyi planlayabilir. Bu, belirli bir dönemdeki kaynak kullanım

eğilimlerini analiz ederek, gelecekteki büyüme veya artan talepler için gerekli kaynakları tahmin etmelerine olanak tanır. Örneğin, bir uygulamanın kullanımı artıyorsa veya yeni bir iş süreci devreye alınıyorsa, bu durumlar kayıtlardaki verilere dayanarak kapasite artışı gerektiğini belirlemek için kullanılabilir.

5. **Sorun Giderme ve Tanımlama:** Sistem kayıtlama sayesinde sistemdeki sorunları giderebilir veya tanımlayabiliriz. Sistem kayıtlama, sistemde meydana gelen hataları, uyarıları ve bilgilendirme mesajlarını kaydeder. Bu bilgiler, sistem yöneticilerine sorunların kaynağını bulmaları ve çözmeleri için kritik ipuçları sağlar. Örneğin, bir sunucunun aniden çökmesi durumunda, sistem kayıtları çeşitli hataları ve olayları kaydedebilir. Bu kayıtlar incelendiğinde, çökmenin nedeninin disk bozulması, aşırı bellek kullanımı veya bir yazılım hatası gibi bir sorundan kaynaklandığı belirlenebilir.
6. **Uzun Vadeli İyileştirme Fırsatları:** Kayıtlar, sistemdeki uzun vadeli iyileştirme fırsatlarını belirleme ve sistem yönetim stratejilerini geliştirme konusunda önemli bir rol oynar. Örneğin, uzun vadeli trendlerin analizi, sistem altyapısında kapasite artırımı veya donanım güncellemeleri gibi uzun vadeli iyileştirme gereksinimlerini belirlemeye yardımcı olabilir. Ayrıca, sistem kayıtlarının tarihsel verileri, sistemdeki güvenlik ihlalleri veya performans sorunlarını tanımlamak ve gelecekte benzer sorunların tekrarlanmasını önlemek için kullanılabilir.
7. **Uygulama Kayıtları:** Uygulama kayıtları, yazılımın çalışma sürecinde meydana gelen hataları ve istisnaları kaydeder. Bu kayıtlar, geliştiricilere ve destek ekiplerine, uygulamanın beklenmeyen davranışlarını ve hata koşullarını belirleme ve

anlama imkânı sağlar. Örneğin, bir kullanıcı bir işlemi gerçekleştirmeye çalışırken karşılaştığı bir hata durumunda uygulama kayıtları, ilgili hata mesajını ve bunun nedenini kaydederek geliştiricilere bu hatanın kökenini tespit etme ve düzeltme konusunda rehberlik eder.

1.2. Kayıt Dosyaları Nereden Geliyor?

- Ağ hizmetleri
- Uygulamalar
- Veri tabanları
- Sunucular
- Güvenlik duvarları
- Uç noktalar
- IoT Cihazları
- Ağlar



Şekil 1. Kayıt kaynakları.

1.3. Kayıt (Log) Türleri

Her sistem farklı türde veriler üretir ve her bileşen bu verileri kendi kayıtlarında toplar. Bu nedenle, aşağıdakiler de dâhil olmak üzere birçok kayıt türü mevcuttur.

1. **Olay Kayıtları (Event Log):** İşletim sistemlerindeki olay kayıtları, bilgisayarın çalışması sırasında meydana gelen çeşitli olayları kaydetmek için kullanılır. Oturum açma

denemeleri, program yükleme ve güncelleme gibi bu olaylara ilişkin bilgileri kaydeden bir kayıttır.

2. **Sunucu Kayıtları:** Belirli bir zaman diliminde belirli bir sunucuyla ilgili etkinliklerin kaydını içerir. Bu kayıtlar, sunucunun performansını izlemek, sorun giderme yapmak, güvenlik olaylarını incelemek ve diğer yönetim görevlerini yerine getirmek için kullanılır.
3. **Sistem Kaydı:** Sistem kaydı işletim sistemi tarafından oluşturulan başlangıç mesajlarını, sistem değişikliklerini, beklenmedik kapanmaları, hataları, uyarıları ve diğer önemli süreçleri içerir. Bu kayıtlar, Windows, Linux ve macOS gibi çeşitli işletim sistemlerinde bulunur.
4. **Yetkilendirme Kayıtları:** Yetkilendirme kayıtları, belirli bir kullanıcının veya sistemin bir kaynağa erişim izinlerini belirlemek için kullanılır. Bu kayıtlar, kimin hangi kaynağa erişim izinlerine sahip olduğunu izler ve izin değişikliklerini kaydeder. Örneğin, bir kullanıcının bir dosyaya yazma izni verilmesi veya bir uygulamanın belirli bir veri tabanına erişim izni alması gibi durumlar yetkilendirme kayıtlarına kaydedilir.
5. **Değişiklik Kayıtları:** Değişiklik kayıtları, bir uygulama veya dosyada yapılan değişikliklerin kronolojik bir listesini içerir. Bu kayıtlar, dosyaların veya uygulamaların hangi kullanıcılar veya sistemler tarafından değiştirildiğini, ne zaman değiştirildiğini ve yapılan değişikliklerin ne olduğunu belirlemek için kullanılır.
6. **Kullanılabilirlik Kayıtları:** Kullanılabilirlik kayıtları, bir sistem veya ağdaki sistem performansını, çalışma süresini ve kullanılabilirliğini izlemek için kullanılır. Bu kayıtlar, sistemin ne kadar süreyle çalıştığını, hangi süreçlerin ne

kadar süreyle çalıştığını, sistem kesintilerini ve performans sorunlarını kaydeder. Örneğin, bir bulut hizmeti sağlayıcısı düşünelim. Bu sağlayıcı, müşterilere sunucu barındırma hizmeti sağlar ve müşterilerin sistemlerinin sürekli çalışmasını sağlamak için yüksek düzeyde kullanılabilirlik ve performans sağlamak zorundadır. Bu sağlayıcı, sistemlerinin çalışma süresini izlemek ve performans sorunlarını tespit etmek için kullanılabilirlik kayıtlarını kullanır.

7. **Kaynak Kayıtları:** Kaynak kayıtları, bir ağ veya sistemdeki bağlantı sorunları, kapasite sınırları ve kaynak kullanımı hakkında bilgi sağlar. Bu kayıtlar, ağ altyapısındaki performansı izlemek, aşırı yüklenmeleri tespit etmek, kaynak tükenmesi durumlarını belirlemek ve sorun giderme yapmak için kullanılır.
8. **Tehdit Kayıtları:** Tehdit kayıtları, önceden tanımlanmış güvenlik profilleriyle eşleşen sistem, dosya veya uygulama trafiği hakkında bilgi içerir. Örneğin, bir tehdit kaydı, bir ağdaki kötü amaçlı yazılım girişimlerini, bilgisayar korsanlığı faaliyetlerini veya yetkisiz erişim denemelerini kaydedebilir.

1.4. Linux Sistem Kayıtları

Linux da sistem kayıtları önemlidir. Linux'ta herhangi bir problem ile karşılaştığımızda ilk yapmamız gereken şeylerden biri kayıt dosyalarını analiz etmek olmalıdır. Linux ortamlarında oluşturulan kayıt dosyaları 4 farklı kategoriye ayrılır. Bunlar;

- Uygulama Kayıtları
- Olay Kayıtları
- Servis Kayıtları

- Sistem Kayıtları.

Linux da kayıt dosyaları /var/log dizininde bulunur. Bu dizin sistem yöneticisi tarafından değiştirilebilir ve kayıtlar farklı dizinde tutulabilir.

1.5. Linux Sistem Kayıt (Log) Dosyaları

- **cd /var/log**
- **ls**

Bu komutlar ile /var/log dizinine girip ardından dizinde bulunan kayıt dosyalarını listeleyebiliriz.

```
alper@ubuntu: /var/log
File Edit View Search Terminal Help
alper@ubuntu:~$ cd ..
alper@ubuntu:/home$ cd ..
alper@ubuntu:/$ cd var/log
alper@ubuntu:/var/log$ ls
alternatives.log          gdm3                  tallylog
alternatives.log.1        gpu-manager.log       ubuntu-advantage.log
alternatives.log.2.gz     gpu-manager-switch.log ubuntu-advantage.log.1
alternatives.log.3.gz     hp                    ubuntu-advantage.log.2.gz
appport.log               installer             ubuntu-advantage.log.3.gz
appport.log.1             journal               unattended-upgrades
appport.log.2.gz          kern.log              vmware
appport.log.3.gz          kern.log.1            vmware-network.1.log
appport.log.4.gz          kern.log.2.gz         vmware-network.2.log
apt                       kern.log.3.gz         vmware-network.3.log
auth.log                  kern.log.4.gz         vmware-network.4.log
auth.log.1                lastlog               vmware-network.5.log
auth.log.2.gz             mail.log              vmware-network.6.log
auth.log.3.gz             mail.log.1            vmware-network.7.log
auth.log.4.gz             mail.log.2.gz         vmware-network.8.log
bootstrap.log             postgresql             vmware-network.9.log
btftp                     speech-dispatcher     vmware-network.log
btftp.1                   syslog                vmware-vmsvc-root.1.log
cups                      syslog.1              vmware-vmsvc-root.2.log
dist-upgrade              syslog.2.gz           vmware-vmsvc-root.3.log
dpkg.log                  syslog.3.gz           vmware-vmsvc-root.log
dpkg.log.1                syslog.4.gz           vmware-vntoolsd-root.log
dpkg.log.2.gz             syslog.5.gz           wtmp
dpkg.log.3.gz             syslog.6.gz           wtmp.1
faillog                   syslog.7.gz
fontconfig.log            sysstat
```

Şekil 2. /var/log dizini.

➤ **/var/log/syslog**

Bu dosya, Linux sistemlerinde genel olayların kaydedildiği bir kayıt dosyasıdır. Sistemde meydana gelen çeşitli olaylar, hata mesajları ve uyarılar bu dosyaya kaydedilir. Örneğin, sistemin başlatılması, ağ bağlantıları, hizmetlerin başlatılması ve kapanması gibi olaylar bu dosyada izlenebilir. İçeriği görüntülemek için **cat /var/log/syslog** komutu kullanılır.

```
kiper@ubuntu:/var/log$ cat syslog
Apr 30 20:36:21 ubuntu rsyslogd: [origin software="rsyslogd" swVersion="8.32.0" x-pid="986" x-info="http://www.rsyslog.com"] rsyslogd was HUPed
Apr 30 20:36:27 ubuntu anacron[982]: Job 'cron.daily' terminated
Apr 30 20:36:29 ubuntu dbus-daemon[1003]: [system] Activating via systemd: service name='org.freedesktop.timedate1' unit='dbus-org.freedesktop.timedate1.service' request_id=1
Apr 30 20:36:29 ubuntu systemd[1]: Starting Time & Date Service...
Apr 30 20:36:29 ubuntu dbus-daemon[1003]: [system] Successfully activated service 'org.freedesktop.timedate1'
Apr 30 20:36:29 ubuntu systemd[1]: Started Time & Date Service.
Apr 30 20:36:30 ubuntu snapd[1033]: storehelpers.gn:791: cannot refresh: snap has no updates available: 'bare', 'core20', 'gnome-3-34-1804', 'gnome-3-38-2004', 'gnome-system-monitor', 'gtk-common-themes', 'snapd'
Apr 30 20:36:35 ubuntu systemd[1]: Reloading.
Apr 30 20:36:35 ubuntu systemd[1]: Mounting Mount unit for core18, revision 2823...
Apr 30 20:36:35 ubuntu systemd[1]: Mounted Mount unit for core18, revision 2823.
Apr 30 20:36:35 ubuntu /usr/lib/gdm3/gdm-x-session[2784]: (!) vmware(0): New layout.
Apr 30 20:36:35 ubuntu /usr/lib/gdm3/gdm-x-session[2784]: (!) vmware(0): 0: 0 0 800 767
Apr 30 20:36:36 ubuntu gsd-color[3668]: unable to get EDID for xrandr-Virtual1: unable to get EDID for output
Apr 30 20:36:36 ubuntu systemd[1]: Reloading.
Apr 30 20:36:36 ubuntu gsd-color[3668]: unable to get EDID for xrandr-Virtual1: unable to get EDID for output
Apr 30 20:36:36 ubuntu gsd-color[3668]: unable to get EDID for xrandr-Virtual1: unable to get EDID for output
Apr 30 20:36:40 ubuntu systemd[1]: Reloading.
Apr 30 20:36:49 ubuntu systemd[1]: Mounting Mount unit for core22, revision 1380...
Apr 30 20:36:49 ubuntu systemd[1]: Mounted Mount unit for core22, revision 1380.
Apr 30 20:36:49 ubuntu systemd[1]: Reloading.
Apr 30 20:36:52 ubuntu systemd[1]: Reloading.
Apr 30 20:36:52 ubuntu systemd[1]: Mounting Mount unit for gnome-characters, revision 797...
Apr 30 20:36:52 ubuntu systemd[1]: Mounted Mount unit for gnome-characters, revision 797.
Apr 30 20:36:52 ubuntu kernel: [ 343.876277] kauditd_printk_skb: 27 callbacks suppressed
Apr 30 20:36:52 ubuntu kernel: [ 343.876278] audit: type=1400 audit(1714498252.993:39): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="/snapd.pid-4035" comm="apparmor_parser"
Apr 30 20:36:53 ubuntu kernel: [ 343.891726] audit: type=1400 audit(1714498253.009:40): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="/snapd/mount-namespace-capture-helper" pid=4035 comm="apparmor_parser"
Apr 30 20:36:53 ubuntu kernel: [ 344.688669] audit: type=1400 audit(1714498253.725:41): apparmor="STATUS" operation="profile_replace" profile="unconfined" name="snapd-comm" comm="apparmor_parser"
```

Şekil 3. /var/log/syslog dizini.

➤ **/var/log/faillog**

Bu dosya, başarısız oturum açma girişimleri hakkında kayıt içerir. Özellikle güvenlikle ilgili olarak, kullanıcı adı ve şifre kırma gibi kaba kuvvet saldırılarına karşı koruma sağlar. İçeriğini görüntülemek için **cat /var/log/faillog** komutu kullanılır.

➤ **/var/log/messages**

Bu dosya, sistemin genel kayıtlarını tutar. Donanım ve yazılımla ilgili yaşanan problemler bu dosyaya kaydedilir. Özellikle Ubuntu tabanlı sistemlerde /var/log/syslog olarak da adlandırılır. İçeriği görüntülemek için `cat /var/log/messages` komutu kullanılır.

➤ **/var/log/cron**

Bu dosya, cron işlemlerinin kaydedildiği bir kayıt dosyasıdır. Cron servisinde tanımlanan işlemler ve hatalar bu dosyaya kaydedilir. İçeriği görüntülemek için `cat /var/log/cron` komutu kullanılır.

Not: Cron, Unix ve Unix benzeri işletim sistemlerinde zamanlanmış görevlerin otomatik olarak çalıştırılmasını sağlayan bir zamanlama aracıdır. Genellikle veri tabanı yedeklemeleri, günlük rapor oluşturma ve sistem bakımı gibi tekrarlanan işlerin otomatikleştirilmesi için kullanılır.

➤ **/var/log/auth.log**

“auth.log” authentication (kimlik doğrulama) kelimesinden gelir. Bu sebeple bu dosya, kimlik doğrulama işlemlerinin kaydedildiği bir kayıt dosyasıdır. Kullanıcıların oturum açma girişimleri ve yetkilendirme hataları bu dosyada görüntülenir. Örneğin kullanıcılardan birisi giriş yapamazsa, bu işlem “auth.log” dosyasına kaydedilir. Bu dosyayı okumak için `cat /var/log/auth.log` komutu kullanılır.

```
alper@ubuntu: /var/log
File Edit View Search Terminal Help
May 1 11:43:05 ubuntu systemd-logind[978]: System is rebooting.
May 1 11:43:05 ubuntu dbus-daemon[1003]: [system] Rejected send message, 1 matched rules; type="error", sender=":1.83" (uid=1000 pid=2941 comm="/usr/bin/pulseaudio --start --log-target=syslog " label="unconfined") interface="(unset)" member="(unset)" error name="org.bluez.MediaEndpoint1.Error.NotImplemented" requested_reply="0" destination=":1.7" (uid=0 pid=1025 comm="/usr/lib/bluetooth/bluetoothd " label="unconfined")
May 1 11:43:05 ubuntu dbus-daemon[1003]: message repeated 3 times: [ [system] Rejected send message, 1 matched rules; type="error", sender=":1.83" (uid=1000 pid=2941 comm="/usr/bin/pulseaudio --start --log-target=syslog " label="unconfined") interface="(unset)" member="(unset)" error name="org.bluez.MediaEndpoint1.Error.NotImplemented" requested_reply="0" destination=":1.7" (uid=0 pid=1025 comm="/usr/lib/bluetooth/bluetoothd " label="unconfined")]
May 1 11:43:05 ubuntu dbus-daemon[1003]: [system] Rejected send message, 1 matched rules; type="error", sender=":1.83" (uid=1000 pid=2941 comm="/usr/bin/pulseaudio --start --log-target=syslog " label="unconfined") interface="(unset)" member="(unset)" error name="org.freedesktop.DBus.Error.UnknownMethod" requested_reply="0" destination=":1.7" (uid=0 pid=1025 comm="/usr/lib/bluetooth/bluetoothd " label="unconfined")
May 1 11:43:05 ubuntu systemd: pam_unix(systemd-user:session): session closed for user alper
May 1 11:43:36 ubuntu systemd-logind[965]: New seat seat0.
May 1 11:43:36 ubuntu systemd-logind[965]: Watching system buttons on /dev/input/event0 (Power Button)
May 1 11:43:36 ubuntu systemd-logind[965]: Watching system buttons on /dev/input/event1 (AT Translated Set 2 keyboard)
May 1 11:43:38 ubuntu gdm-launch-environment]: pam_unix(gdm-launch-environment:
```

Şekil 4. /var/log/auth.log dizini.

➤ /var/log/secure

Bu dosya, Red Hat ve CentOS gibi sistemlerde kullanılan /var/log/auth.log dosyasının benzeridir. Yetkilendirme sistemi ile ilgili tüm mesajları kaydeder ve güvenlikle ilgili olayları içerir. İçeriği görüntülemek için `cat /var/log/secure` komutu kullanılır.

➤ **var/log/boot.log**

Bu dosya, sistem açılışında meydana gelen olayları kaydeder. Uygun olmayan kapanmalar, önyükleme hataları ve beklenmedik yeniden başlatmalar bu dosyada görüntülenir. İçeriği görüntülemek için `cat /var/log/boot.log` komutu kullanılır.

➤ **var/log/dmesg**

Bu dosya, donanım sorunlarını teşhis etmek için kullanılır. Özellikle donanım algılama sorunları ve sürücü hataları bu dosyada kaydedilir. İçeriği görüntülemek için `cat /var/log/dmesg` komutu kullanılır.

➤ **/var/log/yum.log**

Bu dosya, Red Hat ve CentOS gibi sistemlerde kullanılan bir kayıt dosyasıdır. Yum paket yöneticisi kullanılarak yapılan işlemleri ve yüklenen yazılımların kaydını içerir. Bu dosyanın içeriğini okumak için `cat /var/log/yum.log` komutu kullanılır.

➤ **/var/log/maillog veya mail.log**

Bu dosya, e-posta işlemleriyle ilgili kayıtları tutar. Gönderilen ve alınan e-postaların bilgilerini içerir. İçeriği görüntülemek için `cat /var/log/maillog` veya `cat /var/log/mail.log` komutları kullanılır. Bu dosya, farklı e-posta sunucuları veya servisler kurulduğunda değişiklik gösterebilir.


```
alper@ubuntu:/var/log$ ls * | egrep "mail"
ls: cannot open directory 'gdm3': Permission denied
ls: cannot open directory 'speech-dispatcher': Permission denied
mail.log
mail.log.1
mail.log.2.gz
alper@ubuntu:/var/log$ cat mail.log
May  1 11:43:40 ubuntu postfix/postfix-script[1482]: starting the Postfix mail s
ystem
May  1 11:43:40 ubuntu postfix/master[1484]: daemon started -- version 3.3.0, co
nfiguration /etc/postfix
alper@ubuntu:/var/log$
```

Şekil 5. /var/log/mail.log dizini.

1.6. Windows Sistem Kayıtları (Log)

Windows sistem kayıtları, Windows işletim sistemi tarafından tutulan ayrıntılı kayıtlardır ve sistem, güvenlik ve uygulamalarla ilgili olayları içerir. Bu kayıtlar, sistemi izlemek, güvenlik açıklarını tespit etmek ve gelecekteki sorunları öngörmek amacıyla kullanılabilir. Microsoft, Windows olay kayıtlarını ilk olarak Windows Vista ve Windows Server 2008 sürümünde sundu. Bu, Windows'un sonraki tüm sürümlerine dâhil edildi

1.7. Windows Kayıt (Log) Kategorileri

Windows da kayıtlar 5'e ayrılır bunlar;

- **Uygulama:** Windows'ta uygulamalarda meydana gelen olayları içerir ve "Uygulama" başlığı altında Olay Görüntüleyicisi'ne (Event Viewer) kaydedilir. Örneğin, Paint uygulamasını açarken uygulamanın, oluşacak sorun doğrultusunda kapanması.

- **Güvenlik:** Sistemin güvenliğini içeren olayları içerir ve “Güvenlik” başlığı altında Olay Görüntüleyicisi ’ne kaydedilir. Örneğin, başarısız oturum açma denemesi.
- **Kurulum:** Güncellemeler ve yüklemeler gibi olayları içerir. “Kurulum” başlığı altında Olay Görüntüleyicisi ’ne kaydedilir.
- **Sistem:** Windows sistemi ve bileşenleriyle ilgili olayları içerir. Bu kayıt, bilgisayarın genel performansı, sürücülerin yüklenmesi, donanım sorunları ve diğer sistem düzeyindeki olayları kaydeder.
- **İletilen Olaylar:** Aynı ağdaki diğer bilgisayarlardan iletilen olay kayıtlarını içerir. Bu, merkezi bir yerden birden fazla bilgisayarın olaylarını izlemek veya toplamak için kullanılır. Örneğin, başka bir bilgisayarın sistem kaydında kaydedilen bir ağ bağlantı hatası. Bu, iletilen olaylar kaydına "Etkinlik Kimliği: 12345" gibi bir kimlikle kaydedilebilir.

Windows Logs			
Name	Type	Number of Events	Size
Application	Administrative	24,715	20.00 MB
Security	Administrative	37,686	20.00 MB
Setup	Operational	265	1.00 MB
System	Administrative	41,020	20.00 MB
Forwarded Events	Operational	0	0 Bytes

Şekil 6. Windows kayıtları.

1.8. Windows Sistem Kayıtları (Log) Önem Düzeyi

Sistem kayıtları önem düzeyleri, kayıtların ciddiyetini ve önemini belirtmek için aşağıdaki gibi sınıflandırılmıştır.

- **Bilgi (Information):** Bilgi, bir olayın herhangi bir sorun olmadan gerçekleştiği anlamına gelir. Genellikle Windows'ta kayıtların çoğu bilgi olaylarından oluşur.
- **Ayrıntılı (Verbose):** Belirli bir olayla ilgili ilerleme veya başarı mesajlarını belirtir. Bu tür olaylar genellikle sistem veya uygulama durumunu bildiren ek bilgiler içerir. Örneğin, bir yedekleme işleminin tamamlanması veya bir sistem güncellemesinin başarılı bir şekilde yüklenmesi gibi.
- **Uyarı (Warning):** Sistem yöneticilerinin dikkat etmesi gereken olası bir sorunu vurgular. Bu tür olaylar, potansiyel bir sorunun belirtileri veya önlemlerin alınması gereken durumları içerir. Örneğin, bir disk bölümünün dolmak üzere olması veya bir uygulamanın beklenmedik şekilde kapanması gibi.
- **Hata (Error):** Sistemde veya hizmette anında müdahale gerektirmeyen sorunları açıklar. Bu tür olaylar, bir hizmetin başarısız olması, bir uygulamanın çökmesi veya bir donanım bileşeninde bir arızanın belirlenmesi gibi sorunları içerir.

- **Kritik (Critical):** Bir uygulamadaki veya sistemdeki acil müdahale gerektiren önemli bir sorunu belirtir. Bu tür olaylar, sistem çökmesi, kritik bir hizmetin başarısız olması veya güvenlik açığının tespit edilmesi gibi ciddi sorunları içerir. Bu tür olaylar genellikle derhal ele alınması gereken acil durumları belirtir.
- **Başarı Denetimi (Success Audit):** Sadece “Güvenlik” kategorisinde bulunur. Denetlenen güvenlik erişiminin başarılı bir şekilde gerçekleştirildiğini belirtir. Bu tür olaylar, genellikle kullanıcı oturum açma girişimlerinin başarılı olduğunu gösterir.
- **Arıza Denetimi (Failure Audit):** Sadece “Güvenlik” kategorisinde bulunur. Denetlenen güvenlik erişiminin başarısız olduğunu belirtir. Bu tür olaylar, genellikle güvenlik duvarı tarafından engellenen veya izinsiz erişim girişimlerini içerir.

1.9. Windows Sistem Kayıtları (Log) Öğeleri

System Number of events: 41,020				
Level	Date and Time	Source	Event ID	Task Category
Information	01/05/2024 16:07:10	Kernel-Pow...	521	(220)
Information	01/05/2024 16:07:04	Kernel-Pow...	521	(220)
Information	01/05/2024 15:58:32	Service Co...	7045	None
Information	01/05/2024 15:58:32	WindowsU...	19	Windows Update Agent

Event 521, Kernel-Power

General Details

Active battery count change.

Log Name: System

Source: Kernel-Power

Event ID: 521

Level: Information

User: SYSTEM

OpCode: Info

More Information: [Event Log Online Help](#)

Logged: 01/05/2024 16:07:10

Task Category: (220)

Keywords: (1024),(4)

Computer: Ovina

Şekil 7. Windows kayıt öğesi.

Windows Olay Görüntüleyicisi kayıtlarının daha iyi anlaşılabilmesi için belli bir standartta saklar. Bu standardın sınıflandırılması aşağıdaki gibidir:

- **Seviye:** Olayın önem düzeyini belirtir. Farklı seviyeler bilgi, hata, uyarı, ayrıntılı ve kritik şeklinde kategorize edilir.
- **Tarih/Zaman:** Bu, olayın ne zaman meydana geldiğini belirtmek için kullanılır.
- **Kaynak:** Olayın sebep olduğu yazılımın veya uygulamanın adını verir.

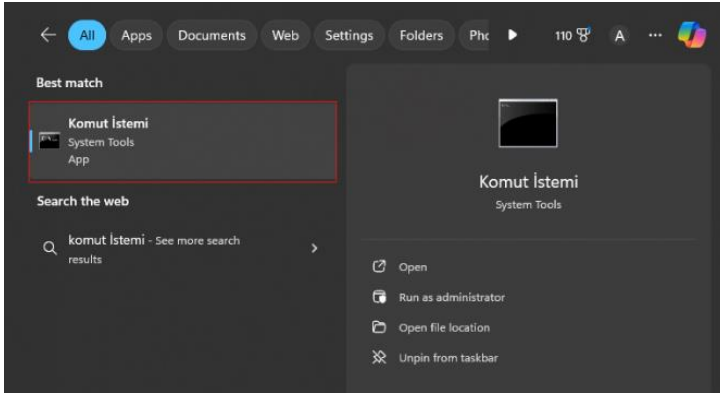
- **Etkinlik ID:** Windows'un olaya atadığı benzersiz ID'dir. Bu sayede olay, basit ve benzersiz bir şekilde tanımlanabilir.
- **Görev Kategorisi:** Kaydedilen olayın türünü tanımlar. Bu, olayın ne tür bir etkinlikle ilişkili olduğunu belirtir. Örneğin, "Sistem Başlatma", "Ağ Bağlantısı" gibi.
- **Kayıt (Log) Adı:** Farklı bileşenlerden gelen olayların kaydedileceği kayıtların adı. Genellikle sistem, güvenlik ve uygulama kayıtları olarak adlandırılırlar. Örneğin, "Sistem", "Güvenlik" ve "Uygulama" gibi.
- **Kullanıcı:** Olayın meydana geldiği sırada oturum açmış olan kullanıcının adı. Bu, belirli bir kullanıcıyla ilişkili olayları belirlemeye yardımcı olabilir.
- **Bilgisayar:** Olayın kaydedildiği bilgisayarın adı. Büyük bir ağ ortamında, birden fazla bilgisayar olabilir, bu nedenle olayın hangi bilgisayardan kaydedildiğini belirtmek önemlidir.

1.10. Windows Olay Görüntüleyicisi (Event Viewer) Nasıl Açılır?

Windows'ta Olay Görüntüleyicisini görüntülemek için birden fazla yöntem bulunmaktadır. Bu kısımda sadece komut isteminden nasıl açılacağından bahsedilmektedir.

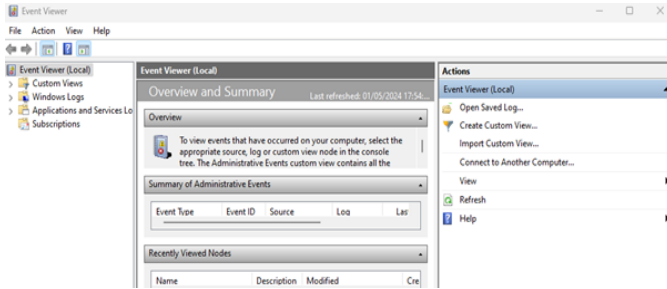
Komut İstemi

1. Windows arama kısmına komut istemi yazarak karşınıza gelen araca tıklayın.



Şekil 8. Windows arama penceresi.

2. Ardından karşınıza gelen komut istemi ekranına “Eventvwr” yazıp klavyenizden enter tuşuna tıklayın.



Şekil 9. Event viewer.

2. MONITORING NEDİR?

Bir networkteki bütün bileşenlerin performansını sürekli olarak takip etme sürecini temsil eder. Sistem ve ağ izleme, olası sorunları önceden tespit ederek anında müdahale etmenizi sağlar ve böylece sistem ve ağ yapısının düzgün çalışmasını kontrol altında tutmanıza yardımcı olur.

2.1. Monitoring Yapmanın Başlıca Faydaları Nelerdir?

Performans İyileştirme

Monitoring, yavaşlamaları ve sorunlu alanları bularak performans verilerini toplar, böylece sistem yöneticilerinin ağ performansını optimize etmesine ve olası problemlere hızlıca müdahale etmesine olanak tanır. Ayrıca, ileride oluşabilecek problemlerin önlenmesine yardımcı olur.

Güvenlik ve Tehdit Tespiti

Ağ izleme sistemlerinin en önemli faydalarından birisi, kötü niyetli eylemleri tespit etmektir. Bu sistemler, ağ tehditlerini, izinsiz indirmelerden şifre değişikliklerine kadar geniş bir yelpazede izleyebilir ve olası durumları hızlı bir şekilde tespit ederek olağandışı davranışları belirleyebilir. Bu sayede güvenlik açıkları minimize edilebilir ve potansiyel riskler önceden engellenebilir.

2.2. Prometheus Tarihçesi

Prometheus, 2012 senesinde SoundCloud isimli müzik platformunda Matt T. Proud, Julius Volz ve Björn Rabenstein ile geliştirilen SoundCloud'un karışık yapısını detaylı izleyebilmek için

yapılan projedir. İlerleyen tarihlerde açık kaynak şekilde sunulmuş ve geniş topluluk tarafından ilgi görmüştür. 2015 senesinde Cloud Native Computing Foundation tarafınca Kabul görmüş. Günümüzde Prometheus monitoring çokça büyük şirketler ve organizasyon tarafından tercih edilen güvenilir ve ölçeklenebilir izleme çözümüdür.

2.3.Prometheus Nedir? Ne İçin Kullanılır?

Prometheus, başlangıçta SoundCloud tarafından geliştirilen açık kaynaklı, time series monitoring ve alarm yönetim aracıdır. Kendine has sorgulama dili vardır (PromQL). Metrik adı, label(lar) ve bunlara karşılık gelen değerden oluşan bir metrik modeline sahiptir. Genellikle metrikleri belirli zaman aralıklarında kaynaklardan okumak üzere ayarlanır.

Pushgateway özelliği ile kaynaklar tarafından metriklerin Prometheus servera gönderilmesi de sağlanabilir. Metrikleri, grafana gibi birçok monitoring paneli üzerinde kullanılabilir. Topluluk tarafından kabul görmüş ve yaygın olarak kullanılmaktadır.

Prometheus; CPU kullanımı, bellek kullanımı, ağ trafiği ve disk kullanımı gibi sistem performansına ilişkin metrikleri takip ederek kullanıcıya sistemlerinin iç işleyişini ve performansını daha iyi anlamalarını sağlar. Bu tür metrikler, mevcut veya gelecekte oluşabilecek sorunları erken tespit etmeye ve daha sorunsuz bir sistem süreci yürütmeye yarar. Prometheus ayrıca sistem yöneticileri veya operasyon ekiplerinin, belirli sistemlerdeki performans sorunlarını belirlemelerini, kapasite planlaması yapmalarını ve gelecekteki olası sorunları tahmin etmelerini sağlayan metrik veriler sunar.

Prometheus'un uyarı sistemine dayanan işlevselliği, veri doğrulanan eşik değerler ile uyarılar verilir ve operasyon ekipleri de bu sorunlara daha hızlı müdahale eder. Prometheus'un metrik toplama ve depolama sunumları da ayrıca kullanılarak sistem hata oranının azaltmak daha

hızlı ve daha güvenilir bir sistem yapısına ulaşabilmek mümkün hale gelir. Sonuç olarak Prometheus, modern dağıtılmış sistemlerin izlenmesine ve kontrol edilmesine kesinlikle vazgeçilmez bir araçtır. Böylece, sistem herkes için daha güvenilir, daha yüksek performanslı ve daha ölçeklenebilir olur.

2.4.Prometheus'un Kullanım Alanları Nelerdir?

Sistem Performans İzleme

Prometheus, sunucuları, ağ altyapısını ve diğer sistem bileşenlerini izleyerek olası sorunları tespit etmenizi sağlar. Bu sayede sistemlerinizdeki yavaşlamaları veya hataları hızlıca teşhis edebilir ve çözebilirsiniz.

Uygulama Performans İzleme

Uygulama geliştirirken Prometheus, uygulamanızdaki hataları, performans düşüşlerini ve diğer sorunları izlemenizi sağlar. Bu şekilde uygulamanızın daha verimli çalışmasını sağlayabilirsiniz.

Bulut Tabanlı Sistemlerin Yönetimi

Prometheus, özellikle Kubernetes gibi bulut tabanlı teknolojilerle kullanıldığında, sistemlerin daha iyi yönetilmesine yardımcı olur ve dağıtımların izlenmesini sağlar.

Kapasite Planlama

Prometheus, sistem ve uygulama izleme kapasitesiyle kaynakların kullanımını optimize etmenize yardımcı olur. Bu sayede, kaynakları verimli bir şekilde planlayabilir ve maliyetleri düşürebilirsiniz.

Olay İzleme ve Alarm Yönetimi

Prometheus, olayları izleyerek kullanıcı tanımlı koşullara göre alarmlar oluşturmanıza imkân tanır. Bu özellikle kritik sorunları hızla tespit etmenize ve çözüm süreçlerini hızlandırmanıza yardımcı olur.

Prometheus, finans, sağlık hizmetleri, e-ticaret, telekomünikasyon ve hizmet sağlayıcı gibi birçok sektörde performans izleme ve alarm yönetimi için yaygın olarak kullanılmaktadır.

2.5. Prometheus Metrik Türleri

Prometheus, bir zaman serisi veritabanı ve metrik toplama aracıdır. Prometheus, uygulama ve sistemlerin performansını ve durumunu izlemek için kullanılır. Prometheus, ölçüm, toplama ve sorgulama için kullanılan bir dizi standart metrik içerir. Bu metrik türleri, sistemlerin farklı yönlerini izlemek ve anlamınıza yardımcı olur.

Bazı temel Prometheus metrikleri ve ne işe yaradıkları;

Sayaçlar (Counters)

Sayaçlar yalnızca artabilir veya sıfırlanabilir metriklerdir. Genellikle istek sayısı, hata sayısı veya aktarılan bayt miktarı gibi sürekli artan veya azalan değerleri izlemek için kullanılırlar. Örneğin, bir web sunucusunun aldığı toplam istek sayısını veya bir servisin işlediği toplam hata sayısını izlemek için sayaçlar kullanılabilir.

Göstergeler (Gauges)

Göstergeler, her yöne ilerleyebilen metriklerdir. Genellikle anlık durum olarak kullanılan mevcut bellek kullanımı, CPU kullanımı veya eşzamanlı istemci sayısı yer almaktadır. Örneğin, bir sunucunun anlık CPU kullanımını veya hafıza kullanımını izlemek için göstergeler kullanılabilir.

Histogramlar

Histogramlar, bir metriğin dağılımını izlemek için kullanılabilen bir başka araçtır. Genellikle istek gecikmelerinin yüzdelik dilimlerini veya yanıt boyutlarının dağılımını görmek için yaygın olarak kullanılırlar. Örneğin, bir web uygulamasının aldığı isteklerin yüzde 87'lik dilimdeki gecikme süresini veya yanıt boyutlarının dağılımını izlemek için histogramlar kullanılabilir.

Özetler (Summaries)

Özetler, histogramları andıran ancak minimum, maksimum ve ortalama gibi özet istatistikleri de içeren formda olabilir. İstek gecikmeleri, yanıt boyutları veya hata sayıları gibi metriklerin detaylı analizini yapmamızı sağlar. Örneğin, bir servisin gecikme süresinin ortalaması

2.6. Uygulama Metriklerini Almazsak Ne Olur?

Görünürlük, projelerin başarısı için temel bir faktördür. Sorun giderme ve beklenen performansı sağlamak, görünürlük olmadan zor olabilir. Bu nedenle, SLA'lar (Hizmet Seviyesi Anlaşmaları) ve SLO'lar (Hizmet Seviyesi Hedefleri), performans ve kullanılabilirlik beklentilerini tanımlayarak herkesin aynı sayfada olmasını sağlar. SRE (Site Güvenilirliği Mühendisliği) ekipleri, net SLA'lar ve SLO'lar ile sorunları daha hızlı tanımlayabilir, çözebilir ve önleyebilir. Prometheus metrikleri, sistemlerimizin iç işleyişi hakkında değerli bir görünürlük sağlar. İstek sayısı, istek gecikmesi, bellek kullanımı ve CPU kullanımı gibi faktörleri izlememizi sağlar. Bu bilgiler, sorunları teşhis etmek, gidermek ve sistem performansını optimize etmek için kritik öneme sahiptir. Prometheus metriklerini toplamamak, sistemlerimizin ne olup bittiğine dair eksik kalmasına neden olur. Sorunları tespit edemez ve düzeltici önlemler alamayız. Bu durum, kesintilere, performans

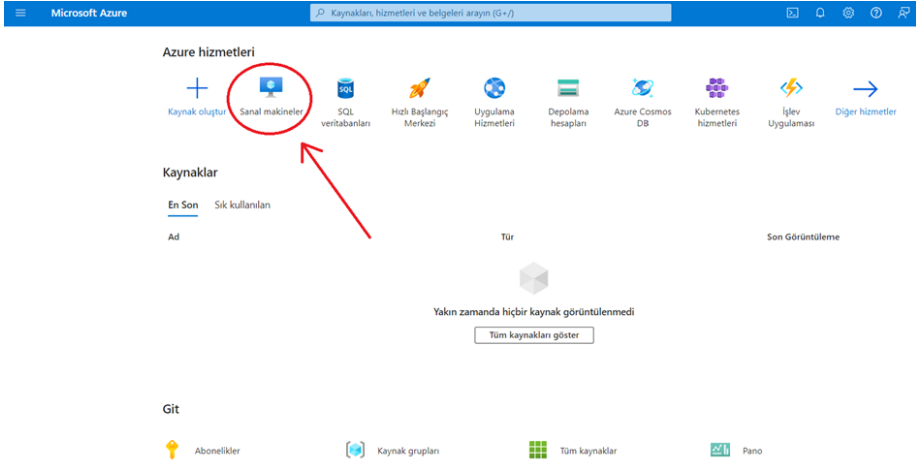
düşüşlerine ve diğer sorunlara yol açabilir. Örneğin, uygulamamızı yük testine tabi tutmak istesek bile, hizmetlerin nasıl davrandığını veya kaynakları nasıl kullandığını anlamadan başarılı bir test yapamayız. Bu nedenle, sistemlerimizi izlemek için Prometheus metriklerini toplamak önemlidir. Bu, sistemlerimizin sorunsuz çalışmasını sağlamak için gereken görünürlüğü sağlayacaktır.

2.7. Prometheus Kurulumu

Microsoft Azure'dan Sanal Sunucu Kurma

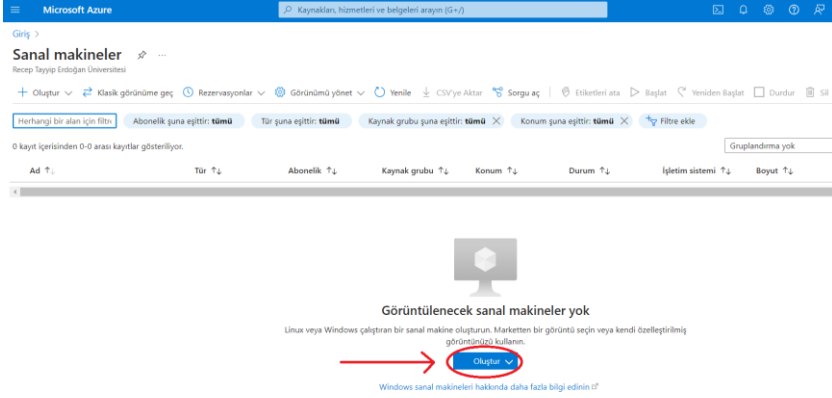
NOT: Sanal sunucu kurmamızın sebebi, local'de çalıştırınca kullanılabileceğimiz kaynaklar sınırlı oluyor. Sanal sunucuda bu durum yok.

1. Bir Microsoft Azure hesabı oluşturun ve hesabınıza giriş yapın. Ardından Azure ana sayfasına gidin. Yukarıda bulunan “Azure hizmetleri” sekmesinden “Sanal Makineler” kısmına tıklayın.



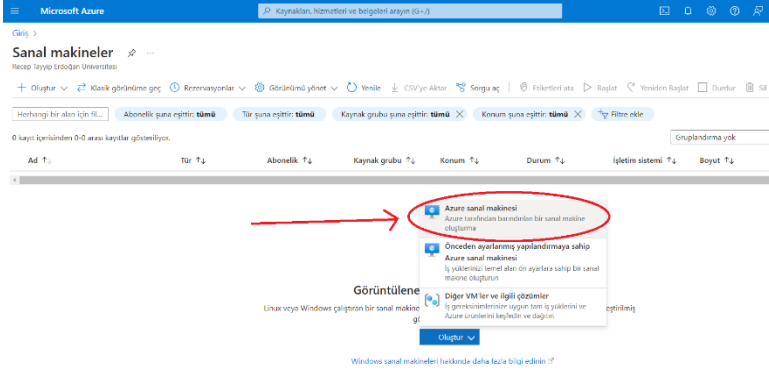
Şekil 10. Azure hesabı oluşturmak için 1. adım

2. Açılan sayfanın alt kısmında bulunan “Oluştur” butonuna tıklayın.



Şekil 11. Azure hesabı oluşturmak için 2. adım

3. Karşımıza seçenekler çıkacak. Buradan “Azure sanal makinesi” yazan kısma tıklayın.



Şekil 12. Azure hesabı oluşturmak için 3. adım

4. Sanal makine ayarlarını yapmamız gerekiyor. Alt kısımdan sanal makinenize bir isim verin.

Microsoft Azure

Kaynakları, hizmetleri ve belgeleri arayın (G+I)

Giriş > Sanal makineler >

Sanal makine oluştur

Temel Ayarlar Diskler Ağ İletişimi Yönetim İzleme Gelişmiş Etiketler Gözden geçir + oluştur

Linux veya Windows'ta çalışan bir sanal makine oluşturun. Azure Market'ten bir görüntü seçin veya kendi özel görüntünüzü kullanın. Temel sekmesini ve daha sonra Gözden geçir + oluştur sekmesini tamamlayarak varsayılan parametrelerle bir sanal makine sağlayın veya tam özelleştirme için bütün sekmeleri gözden geçirin. [Daha fazla bilgi edinin](#)

i Bu abonelik, belirli bölgelerde belirli boyutlardaki VM'leri dağıtmak için uygun olmayabilir.

Proje ayrıntıları

Dağıtılan kaynakları ve maliyetleri yönetmek için bir abonelik seçin. Tüm kaynakları düzenlemek ve yönetmek için klasörler gibi kaynak grupları kullanın.

Abonelik * ⓘ Azure for Students

Kaynak grubu * ⓘ (Yeni) prometheus_group
[Yeni oluştur](#)

Örnek ayrıntıları

Sanal makine adı * ⓘ prometheus ✓

Bölge * ⓘ (US) East US

Şekil 13. Azure hesabı oluşturmak için 4. adım

5. Sayfayı aşağıya doğru kaydırın ve “Yönetici hesabı” kısmını bulun. Burada “Kimlik doğrulama türü”nü Parola olarak seçin ve gerekli alanları doldurun.

Microsoft Azure

Kaynakları, hizmetleri ve belgeleri arayın (G+/)

Giriş > Sanal makineler >

Sanal makine oluşturun

Hazırda beklemeyi etkinleştir (önizleme) ☐

Yönetici hesabı

Kimlik doğrulama türü ☐ SSH ortak anahtarı ☒ Parola

Kullanıcı Adı *

Parola *

Şekil 14. Azure hesabı için 5. adım

6. En altta bulunan “Gözden geçir + oluşturun” butonuna tıklayın.

☒ Parola

Kullanıcı Adı *

Parola *

Parolayı onaylayın *

Gelen bağlantı noktası kuralları

Hangi sanal makine ağ bağlantı noktalarının genel Internet'ten erişilebilir olduğunu seçin. Ağ sekmesinde daha sınırlı veya ayrıntılı ağ erişimi belirtebilirsiniz.

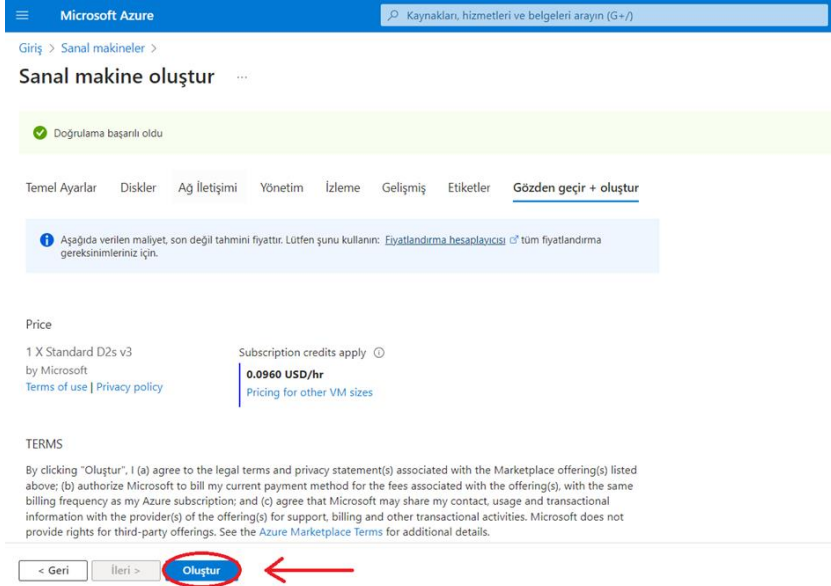
Genel gelen bağlantı noktaları * ☐ Yok ☒ Seçili bağlantı noktalarına izin ver

Gelen bağlantı noktaları seçin *

< Geri Sonraki: Diskler > **Gözden geçir + oluşturun**

Şekil 15. Azure hesabı oluşturmak için 6. adım

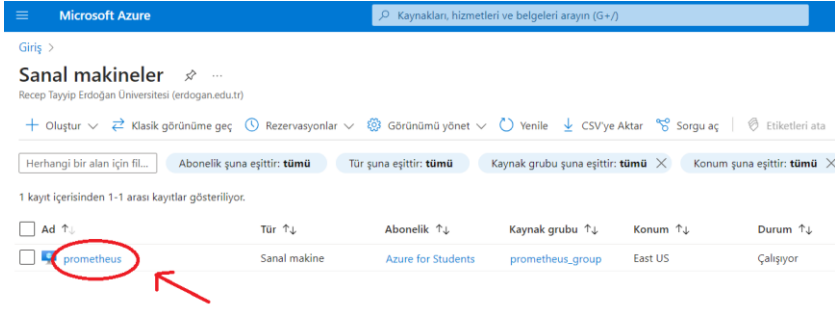
7. Son olarak sanal makine ayarlarınızı gözden geçirin ve aşağıda bulunan “Oluştur” butonuna tıklayınız.



Şekil 16. Azure hesabı oluşturmak için son adım.

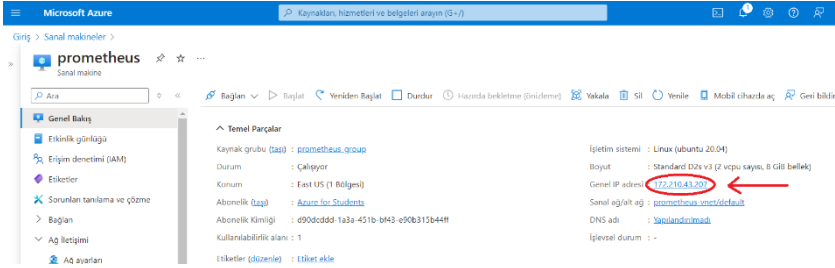
2.8. Putty ile Sanal Sunucuya Bağlanma

1. Azure ana sayfasında bulunan “Sanal makineler” kısmına tıklayın ve açılan ekrandan sanal makinenizi seçin.



Şekil 17. Sanal sunucuya bağlanma.

2. Karşınıza sanal makineniz ile ilgili bilgiler gelecektir. Bu kısımdan “Genel IP adresi” ni bulun ve kopyalayın.

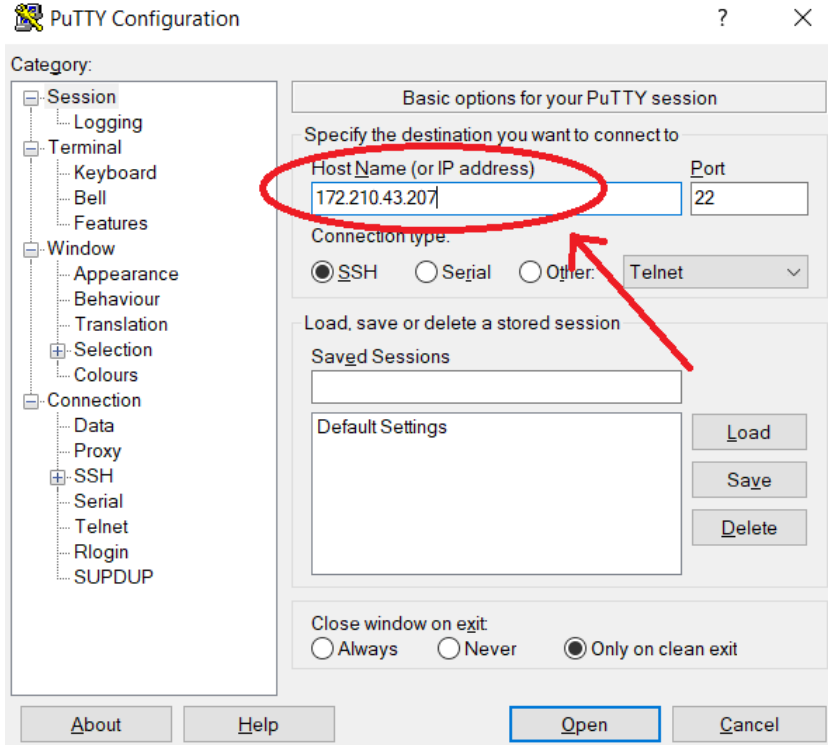


Şekil 18. Sanal sunucuya bağlanmak için IP adresi alma..

3. PuTTY ana sayfasından PuTTY yazılımını indirin ve bilgisayarınıza kurun.
4. PuTTY yazılımını açın ve “Host Name (or IP address)” kutucuğuna sanal makinenizin IP adresini yapıştırın. Ardından aşağıda bulunan “Open” butonuna tıklayın.

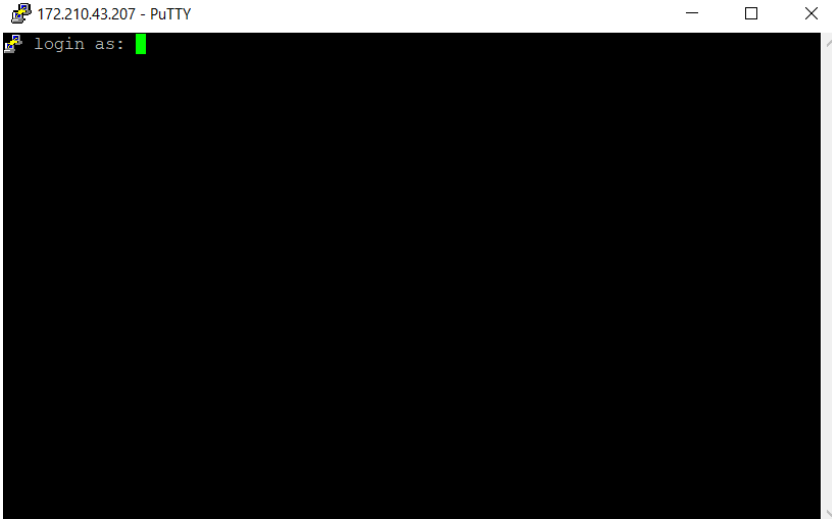
NOT: 22 numaralı port, TCP/IP ağlarında kullanılan bir port numarasıdır ve genellikle SSH (Secure Shell) ile ilişkilendirilir. SSH, ağ üzerin-

den güvenli bir şekilde iletişim kurmak için kullanılan bir protokoldür. Özellikle, uzak sunuculara güvenli bir şekilde erişim sağlamak için sıkça kullanılır.



Şekil 19. PuTTY arayüzü

5. Karşınıza bir CLI (command line interface) açılacak. Sizden kullanıcı adınızı ve şifrenizi girmenizi isteyecek.



Şekil 20. PuTTY terminal ekranı.

6. Kullanıcı adınızı ve şifrenizi girdikten sonra sanal sunucu ile bağlantı kurulacaktır.

2.9. Sanal Sunucuya Prometheus Kurma

1. PuTTY ile sanal sunucumuza bağlandık. Şimdi prometheus'u sanal sunucunuza yüklemelisiniz. Bunu yapabilmek için ilk önce "prometheus.io" sayfasına gidin ve "DOWNLOAD" kısmını açın. Sanal sunucumuzda Linux (Ubuntu) kurulu olduğu için linux indirme linkine sağ tıklayın ve "Bağlantı adresini kopyala" kısmına basın.

prometheus

The Prometheus monitoring system and time series database. [prometheus/prometheus](https://prometheus.io/)

2.52.0-rc.1 / 2024-05-03 Pre-release Release notes				
File name	OS	Arch	Size	SHA256 Checksum
prometheus-2.52.0-rc.1.darwin-amd64.tar.gz	darwin	amd64	100.33 MiB	084c02a2c246c0c246a228820b25a0f2a6770f150570ea30f74366411ac0
prometheus-2.52.0-rc.1.linux-amd64.tar.gz	linux	amd64	99.81 MiB	4629052f6677934936cc4ad2038c208756ee01479e08203904ec04fe37761
prometheus-2.52.0-rc.1.windows-amd64.zip	windows	amd64	102.02 MiB	168708282c3452a205a6e0d50c4635ad07c040f1c0a4dbcf1f3c43e576707692

Şekil 21. Prometheus arayüzü.

2. Prometheus’u indirmek için “wget” komutundan yararlanacağız. Sanal makinenizin CLI’ını açın ve wget yazdıktan sonra kopyaladığınız bağlantı adresini yapıştırın.

NOT: “wget” komutu, Unix/Linux işletim sistemlerinde kullanılan bir komuttur ve bir URL’den dosya indirmek için kullanılır.

```
ridvanks@prometheus: ~  
ridvanks@prometheus:~$ wget https://github.com/prometheus/prometheus/releases/download/v2.52.0-rc.1/prometheus-2.52.0-rc.1.linux-amd64.tar.gz
```

Şekil 22. wget komutu.

3. İndirilen dosya zipli. Bunu dışarı çıkarmak için “tar” komutundan yararlanacağız. “tar” yazdıktan sonra indirdiğiniz dosyanın adını girin ve komutu çalıştırın.

```
ridvanks@prometheus: ~  
ridvanks@prometheus:~$ tar zxvf prometheus-2.52.0-rc.1.linux-amd64.tar.gz
```

Şekil 23. zip dosyasından çıkartma.

4. Dosyayı çıkardıktan sonra “cd” komutu ile içerisine girin. Prometheus 9090 portunu kullanır. Prometheus’u çalıştırmak için bu portu açmamız gerekiyor. İlk olarak Azure CLI yükleyeceğiz fakat ubuntu’da Azure CLI yok. Azure CLI’ı yüklemek için aşağı-

daki adımlar izlenir. Bu adımlarda yazan komutları sırası ile sanal makinenizin CLI’ında çalıştırın.

4.1. Microsoft’un resmi GPG anahtarını ekleme

```
`curl -sL https://packages.microsoft.com/keys/microsoft.asc | \
  gpg --dearmor | \
  sudo tee /etc/apt/trusted.gpg.d/microsoft.gpg > /dev/null`
```

4.2. Azure CLI deposunu ekleme

```
`AZ_REPO=$(lsb_release -cs)
echo "deb [arch=amd64]
https://packages.microsoft.com/repos/azure-cli/ $AZ_REPO
main" | \
  sudo tee /etc/apt/sources.list.d/azure-cli.list`
```

4.3. Depo bilgilerini güncelleme

```
`sudo apt update`
```

4.4. Azure CLI yükleme

```
`sudo apt install azure-cli`
```

5. Bu işlemlerin ardından Azure hesabımızı CLI’a bağlamamız gerekiyor. Bunun için sanal makinenizin CLI’ına “az login” yazın ve doğrulama kodunu girerek hesabınızı doğrulayın.

```
ridvanks@prometheus:~/prometheus-2.52.0-rc.1.linux-amd64$ az login
To sign in, use a web browser to open the page https://microsoft.com/devicelogin
and enter the code BVFWZH26M to authenticate.
[
  {
    "cloudName": "AzureCloud",
    "homeTenantId": "cc10cfdb-ec44-42c1-a4b0-9df4399bf03e",
    "id": "d90dcddd-1a3a-451b-bf43-e90b315b44ff",
    "isDefault": true,
    "managedByTenants": [],
    "name": "Azure for Students",
    "state": "Enabled",
    "tenantId": "cc10cfdb-ec44-42c1-a4b0-9df4399bf03e",
    "user": {
      "name": "ridvan_karasubasi20@erdogan.edu.tr",
      "type": "user"
    }
  }
]
```

Şekil 24. CLI bağlantısı.

6. Şimdi 9090 portunu açabiliriz. Bunun için “az vm open-port --resource-group grup_adi --name vm_adi --port 9090 --priority 900” komutunu girmeliyiz. Buradaki “grup_adi” yazan kısma ve “vm_adi” yazan kısma sanal sunucumuzun bilgilerinden ulaşabiliriz.

```
ridvanks@prometheus:~/prometheus-2.52.0-rc.1.linux-amd64$ az vm open-port --resource-group pr
ometheus_group --name prometheus --port 9090 --priority 900
```

Şekil 25. Port ayarları

7. Prometheus’u başlatabiliriz.

```
ridvanks@prometheus:~/prometheus-2.52.0-rc.1.linux-amd64$ ./prometheus
```

Şekil 10. Prometheus’u başlatmak için gerekli komut.

8. Eğer “listen tcp 0.0.0.0:9090: bind: address already in use” hatasını veriyorsa prometheus çalışıyor demektir. Prometheus’u kapatmamız lazım. Bunun için prometheus’un pid değerini öğrenip kill etmemiz lazım. Pid değerini öğrenmek için konsola “ps aux | grep prometheus” komutunu girin.

NOT: ``ps aux | grep prometheus`` komutu Unix/Linux işletim sistemlerinde bir arşiv dosyasını (genellikle ``.tar.gz`` uzantılı dosyalar) açmak için kullanılır.

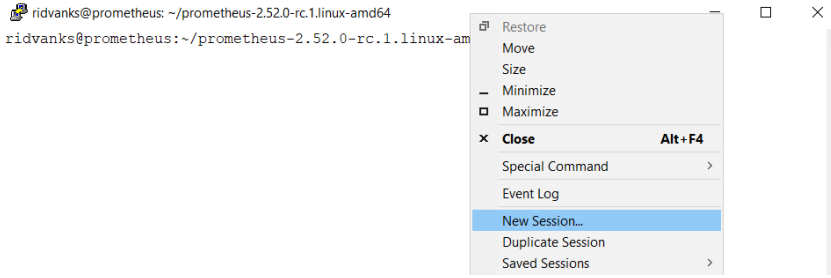
- ``ps``: Sistemde çalışan süreçleri listeler.
- ``a``: Tüm kullanıcıların (sistemi çalıştıran kullanıcı dahil) süreçleri listeler.
- ``u``: Detaylı kullanıcı formatını kullanır. Bu seçenekle birlikte, kullanıcıya ait süreçlerin daha fazla bilgisini gösterir.
- ``x``: Kontrol terminali olmayan süreçleri de listeler. Bu, terminalin dışında çalışan süreçleri de dahil eder.

Pid değerini öğrendikten sonra `“kill -9 pid_degeri”` komutu ile process'i öldürelim.

NOT: ``kill -9 pid_degeri`` komutunda `“-9”` kullanmamızın nedeni, bazı durumlarda process'ler düzgün şekilde sonlandırılmıyor. Bu durumlarda zorla sonlandırmamız gerekiyor. Bunun için ise `“-9”` yazmamız gerekiyor.

2.10. Sanal Makineye Node_Exporter Kurma

1. İlk olarak yeni bir CLI açmalıyız. Bunun için mevcut CLI'a sağ tıklayıp “New Session” kısmına basmalıyız. Sanal sunucumuzun ip adresini girip sanal sunucuya bağlanmalıyız.



Şekil 26. Sanal sunucu bağlantı ayarları.

2. Yeni bir CLI açtık. Şimdi node_exporter’u kurmak için ilk önce “prometheus.io” sayfasına gidin ve “DOWNLOAD” kısmını açın. Açılan sayfayı aşağıya doğru kaydırın ve “node_exporter” kısmını bulun. Sanal sunucumuzda Linux (Ubuntu) kurulu olduğu için linux indirme linkine sağ tıklayın ve “Bağlantı adresini kopyala” kısmına basın.

node_exporter

Exporter for machine metrics [prometheus/node_exporter](#)

1.8.0 / 2024-04-24 [Release notes](#)

File name	OS	Arch	Size	SHA256 Checksum
node_exporter-1.8.0.darwin-amd64.tar.gz	darwin	amd64	4.73 MiB	15ca134a1a08a08a236c4828a973bca508a08155ec13a6dc7979a0a080922
node_exporter-1.8.0.linux-amd64.tar.gz	linux	amd64	10.18 MiB	c384e1d99d518ec40339e0e073c233777e0904a10802d08e0712d0c0f1430

Şekil 27. Node_exporter

3. node_exporter’u kurmak için yine “wget” komutundan yararlanacağız. Yeni açtığınız CLI’ınızı açın ve wget yazdıktan sonra kopyaladığınız bağlantı adresini yapıştırın.

4. İndirilen dosya zipli. Bunu dışarı çıkarmak için “tar” komutundan yararlanacağız. “tar” yazdıktan sonra indirdiğiniz dosyanın adını girin ve komutu çalıştırın.

```
ridvanks@prometheus:~$ tar zxvf node_exporter-1.8.0.linux-amd64.tar.gz
```

Şekil 28. zip dosyasını çıkartma.

5. Dosyayı çıkardıktan sonra “cd” komutu ile içerisine girin. node_exporter 9100 portunu kullanır. node_exporter’u çalıştırmak için bu portu açmamız gerekiyor. Daha önceden Azure CLI yüklediğimiz için tekrar yüklememize gerek yok. O yüzden “az vm open-port --resource-group grup_adi --name vm_adi --port 9100 --priority 899” komutu ile 9100 portunu açabiliriz. Bu komutta eğer priority kısmını 900 olarak girersek 9090 portu ile öncelik sıraları çakışacaktır. bunun önüne geçmek için 899 yazıyoruz. Buradaki “grup_adi” yazan kısma ve “vm_adi” yazan kısma sanal sunucumuzun bilgilerinden ulaşabiliriz.

```
ridvanks@prometheus:~/node_exporter-1.8.0.linux-amd64$ az vm open-port --resource-group prometheus_group --name prometheus --port 9100 --priority 899
```

Şekil 29. Sunucu için konfigürasyonlar

6. Şimdi node_exporter’ı başlatabiliriz. ama önce prometheus.yml dosyasının içerisine 9100 portunu eklememiz gerekiyor. Bunun için prometheus konsoluna geri dönüyoruz ve konsola “vi prometheus.yml” yazıyoruz. “i” tuşuna basarak düzenleme moduna geçiyoruz. “- job_name” den aşağısını kopyalıyoruz ve en alta yapıştırıyoruz. Bunun için klavye imleci en son satırın bir altında olacak şekilde, metnimizi faremizin imleci ile seçiyoruz ve sağ tıklıyoruz. Son hali aşağıdaki gibi olmalıdır.

```
# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this con
  fig.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

  - job_name: "node_exporter"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9100"]
```

Şekil 30. .yml dosyası için port ayarı.

Daha sonra “esc” tuşuna basıyoruz ve ekleme modundan çıkıyoruz. “:” tuşuna basıp “wq!” yazıyoruz.

NOT: “:wq” ve “:wq!” arasındaki fark, :wq kullanırsak dosya yazılabilirse yazılır, yazılamazsa hata verir ama :wq! dosyayı zorla kaydeder.

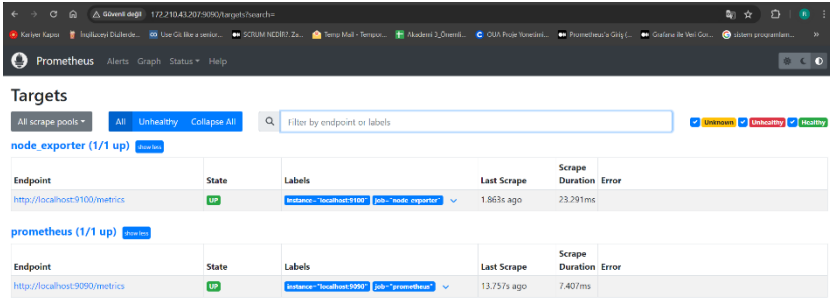
7. Bu işlemleri yaptıktan sonra prometheus ve node_exporter’ı çalıştırın. Bunları çalıştırmak için her birini kendi CLI’larında “./” kullanarak çalıştırabiliriz.

```
ridvanks@prometheus:~/node_exporter-1.8.0.linux-amd64$ ./node_exporter
```

Şekil 31. Prometheus ve node_exporter’in çalıştırılması.

Daha sonra “ip_adresi:9090” adresine gidin. Status’un altında bulunan Targets kısmını kontrol etmeliyiz. Orada node_exporter’u görmemiz lazım.v_m_adi --port 9100 --priority 899” komutu ile 9100 portunu açabiliriz. Bu komutta eğer priority kısmını 900 olarak girsek 9090 portu ile öncelik sıraları

çakışacaktır. bunun önüne geçmek için 899 yazıyoruz. Buradaki “grup_adi” yazan kısma ve “vm_adi” yazan kısma sanal sunucumuzun bilgilerinden ulaşabiliriz.



Şekil 32. Prometheus’a sunucu verilerinin girilmesi.

2.11. Farklı Bir Sanal Makineye Node_Exporter Kurma

1. Daha önce bir sanal makine oluşturmuştuk. Şimdi yine aynı adımları izleyerek yeni bir sanal makine oluşturun. PuTTY ile yeni oluşturduğunuz sanal makinenize bağlanın.
2. Şimdi node_exporter’u kurmak için ilk önce “prometheus.io” sayfasına gidin ve “DOWNLOAD” kısmını açın. Açılan sayfa-yı aşağıya doğru kaydırın ve “node_exporter” kısmını bulun. Sanal sunucumuzda Linux (Ubuntu) kurulu olduğu için linux indirme linkine sağ tıklayın ve “Bağlantı adresini kopyala” kısmına basın

İzleme ve Kayıtlama Sistemleri

node_exporter

Exporter for machine metrics [prometheus/node_exporter](#)

1.8.0 / 2024-04-24 Release notes				
File name	OS	Arch	Size	SHA256 Checksum
node_exporter-1.8.0.darwin-amd64.tar.gz	darwin	amd64	4.73 MiB	15xc134e1a08cd8bc23c6829d971c3ae20f0e08956e1306dc7976eae089922
node_exporter-1.8.0.linux-amd64.tar.gz	linux	amd64	10.18 MiB	c384e5d09d518ec468330e0e073c233f774e0948a18862d089e3f23ac0f3438

Şekil 33. node_exporter’in kurulumu için son ayarlar.

- node_exporter’u kurmak için yine “wget” komutundan yararlanacağız. Yeni açtığınız CLI’nızı açın ve wget yazdıktan sonra kopyaladığınız bağlantı adresini yapıştırın.
- İndirilen dosya zipli. Bunu dışarı çıkarmak için “tar” komutundan yararlanacağız. tar yazdıktan sonra indirdiğiniz dosyanın adını girin ve komutu çalıştırın.

```
ridvanks@prometheus:~$ tar zxvf node_exporter-1.8.0.linux-amd64.tar.gz
```

Şekil 34. zip’ten dosya çıkartma.

- Dosyayı çıkardıktan sonra “cd” komutu ile içerisine girin. node_exporter 9100 portunu kullanır. node_exporter’u çalıştırmak için bu portu açmamız gerekiyor. İlk olarak Azure CLI yükleyeceğiz. Aşağıdaki komutları sırası ile sanal makinenizin CLI’nda çalıştırın

5.1. Microsoft’un resmi GPG anahtarını ekleme:

```
`curl -sL https://packages.microsoft.com/keys/microsoft.asc | \
gpg --dearmor | \
sudo tee /etc/apt/trusted.gpg.d/microsoft.gpg > /dev/null`
```

5.2. Azure CLI deposunu ekleme:

```
`AZ_REPO=$(lsb_release -cs)
```

```
echo "deb [arch=amd64]
https://packages.microsoft.com/repos/azure-cli/ $AZ_REPO
main" | \
sudo tee /etc/apt/sources.list.d/azure-cli.list`
```

5.3. Depo bilgilerini güncelleme:

```
`sudo apt update`
```

5.4. Azure CLI yükleme:

```
`sudo apt install azure-cli`
```

6. Bu işlemlerin ardından Azure hesabımızı CLI'a bağlamamız gerekiyor. Bunun için sanal makinenizin CLI'ına "az login" yazın ve doğrulama kodunu girerek hesabınızı doğrulayın.
7. Şimdi 9100 portunu açabiliriz. Bunun için "az vm open-port --resource-group grup_adi --name vm_adi --port 9100 --priority 900" komutunu kullanmalıyız. Buradaki "grup_adi" yazan kısma ve "vm_adi" yazan kısma sanal sunucumuzun bilgilerinden ulaşabiliriz.

```
ridvanks@node-exporter:~/node_exporter-1.8.0.linux-amd64$ az vm open-port --reso
urce-group node-exporter_group --name node-exporter --port 9100 --priority 900
```

Şekil 35. node_explorer için port ayarı.

8. Şimdi node_exporter'ı başlatabiliriz. ama önce "prometheus.yml" dosyasının içerisine "ip_adresi:9100" portunu eklememiz gerekiyor. Bunun için prometheus'un yüklü olduğu sanal makinenin konsoluna geri dönüyoruz. "vi prometheus.yml" yazıyoruz ve dosyamızı açıyoruz ardından "i" tuşuna basarak düzenleme moduna geçiyoruz. Daha önce eklediğimiz yerin yanına tırnak veya çift tırnak içerisinde "ip_adresi:9100" yazıyoruz. Daha sonra "esc" tuşuna basıyoruz ve ekleme modundan çıkıyoruz. ":" tuşuna basıp ":" yazıyoruz.

```
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]

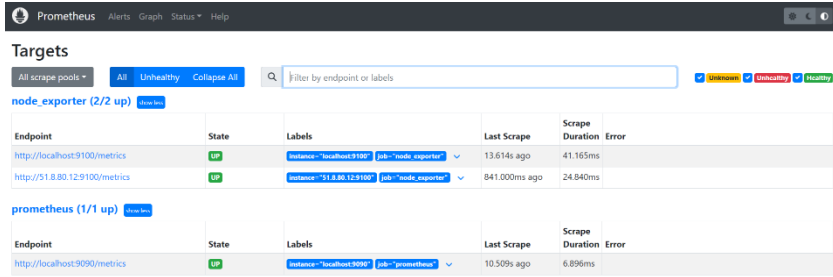
  - job_name: "node_exporter"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9100", "51.8.80.12:9100"]
```

Şekil 36. node_explorer’I çalıştırma.

Bu işlemleri yaptıktan sonra node_exporter’ı çalıştırın. Prometheus’un yüklü olduğu sanal makinenin ip adresini kopyalayın ve “ip_adresi:9090” adresine gidin. Status’un altında bulunan Targets kısmını kontrol edin. Orada yeni node_exporter’u görmemiz lazım.



Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9100/metrics	UP	instance="localhost:9100" job="node_exporter"	13.614s ago	41.165ms	
http://51.8.80.12:9100/metrics	UP	instance="51.8.80.12:9100" job="node_exporter"	841.000ms ago	24.840ms	

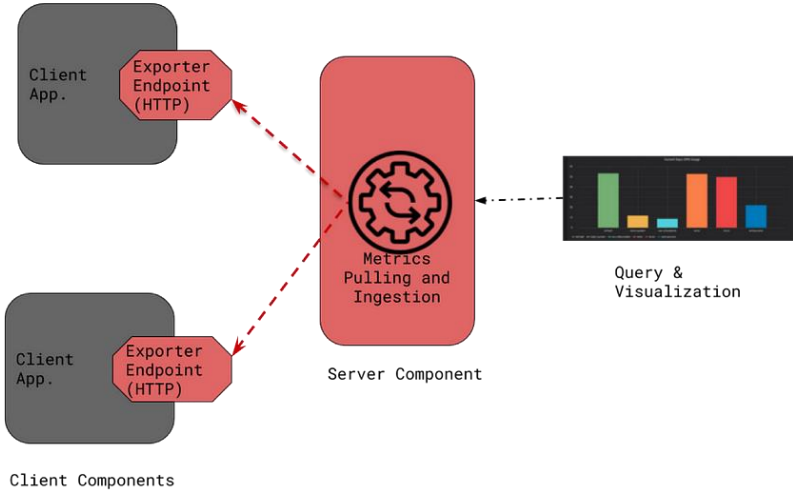
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	10.509s ago	6.896ms	

Şekil 37. Hedeflerin ayarlanması.

2.12. Premetheus Exporter Nedir?

Prometheus exporter ekosisteminin bir parçası olarak kullanılan bir yazılımdır. Bir bilgisayar veya sunucunun sistem metriklerini (örneğin, CPU kullanımı, bellek kullanımı, disk kullanımı, ağ trafiği vb.)

toplamak ve Prometheus tarafından toplanabilir hale getirmek için tasarlanmış bir araçtır. Prometheus dışa aktarıcıları, Prometheus tarafından kullanılan metrikleri toplamak için kullanılan araçlardır. İhracatçılar, metrikleri Prometheus ile uyumlu bir formatta sunarak Prometheus'un bu metrikleri izlemesini ve uyarılar oluşturmalarını sağlar.



Şekil 38. Prometheus exporter şeması.

Prometheus Exporters, uygulamalar ile Prometheus Sunucusu arasında köprü görevi görür. İhracatçılar, uygulamanın kendi metriklerini toplamak için kullandığı protokol üzerinden metrikleri Prometheus'a ileten araçlardır.

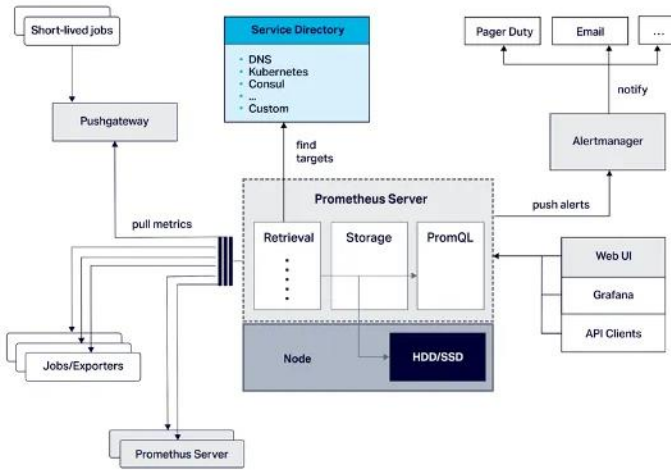
Prometheus Exporter, Prometheus'un izleme yeteneklerini genişletmek için kullanılır. Örneğin, Prometheus bir web sitesinin performansını izlemek için HTTP dışa aktarıcısını, bir veri tabanının

performansını izlemek için MySQL dışı aktarıcısını veya bir uygulamanın performansını izlemek için uygulama dışı aktarıcısını kullanabilir.

Prometheus Exporters, Prometheus'un izleme yeteneklerini genişletmek için güçlü bir araçtır. Kullanımı kolaydır ve çeşitli kaynaklardan alınan metrikleri Prometheus'a aktarabilirler.

2.13. Prometheus Mimarisi

Verileri bir araya getiren, sorgulayan ve depolayan çok farklı yapıların bir arada toplandığı geniş bir mimariye sahiptir.



Şekil 39. Prometheus mimarisi şeması.

1. Prometheus Sunucusu

Üç kısma ayrılır

Veri Toplama Çalışanı

Veri toplama işlevi olan çalışan, belirli hedeflerden metrikleri toplamakla görevlidir. Bu metrikleri toplamak için HTTP istekleri gönderir ve ardından toplanan ölçümleri bir zaman serisi veritabanında saklar.

Zaman Serisi Veritabanı (TSDB)

TSDB, tüm ölçümleri sakladığımız ana veritabanıdır. Prometheus'taki TSDB, zaman serisi ölçümlerinin etkin bir şekilde depolanmasını, geri alınmasını ve yönetilmesini sağlar. Tasarımı ve optimizasyonları, özellikle izleme gereksinimlerini karşılar ve Prometheus'un büyük ölçekli, yüksek performanslı metrik toplama işlemlerini gerçekleştirirken geçmiş verilere hızlı ve güvenilir erişim sağlar.

HTTP Sunucusu

HTTP sunucusu, TSDB'de saklanan verilere erişmemize olanak tanır. Bu verilere erişerek görselleştirmek için HTTP istekleri kullanırız. HTTP sunucusuna bir HTTP isteği yaparak, yerleşik sorgulama dili olan PromQL'i kullanarak sorgular göndeririz. Bu sorguları kullanarak verileri analiz edebilir ve görselleştirebiliriz.

Job/ Exporters

Prometheus, metrikleri "pull" yöntemiyle hedeflenen nodlardan kendisi toplar. Hedefler ("target'lar") "push" işlemi gerçekleştirmezler.

Push Gateway

Bazı metriklerin "pull" yöntemiyle toplanmasına uygun olmadığı durumlarda kullanılır. Bu durumlarda ara katman olarak görev yapar.

Servis Keşfi ("Service Discovery")

İzlenecek hedefleri ve metrikleri belirler. Bunun için iki yöntem vardır: statik bir konfigürasyon dosyası ile ayarlamak veya dinamik olarak eklenecek veya silinecek hedefleri bir veritabanında tutarak Prometheus'u bilgilendirmek.

Prometheus Web UI

Verileri Grafana, API, Prometheus GUI gibi istemcilere sunmak için kullanılır.

Alert Manager

Uyarıları alıcılara iletmekle görevlidir. Örneğin, e-posta, SMS hizmeti, API gibi yöntemlerle uyarıları iletebilir

2. İhracatçılar

İhracatçılar, hedef sistemden metrikleri çıkarmak ve bunları Prometheus veri formatına dönüştürmek için aracı olarak görev yaparlar. Bu metrikleri Prometheus Sunucusu tarafından kazanabilir hale getirmek için bir HTTP uç noktası sağlarlar. Popüler sistemler ve çerçeveler için birçok dışa aktarıcı mevcuttur veya özel ölçümleri ortaya çıkarmak için özelleştirilebilir dışa aktarıcılar geliştirilebilir.

Prometheus'un bazı yerleşik dışa aktarıcıları şunlardır.

Node Exporter (Linux Sunucuları İçin)

Linux sunucularından sistem metriklerini çıkarır ve Prometheus formatına dönüştürür.

Windows Exporter (Windows İçin)

Windows sunuculardan sistem metriklerini çıkarır ve Prometheus tarafından anlaşılabilir formata dönüştürür.

MySQL Exporter

MySQL veritabanından performans ve durum metriklerini çıkarır ve Prometheus için uygun formata getirir.

Apache Exporter

Apache HTTP sunucusundan istatistikleri toplar ve Prometheus veri formatına dönüştürür.

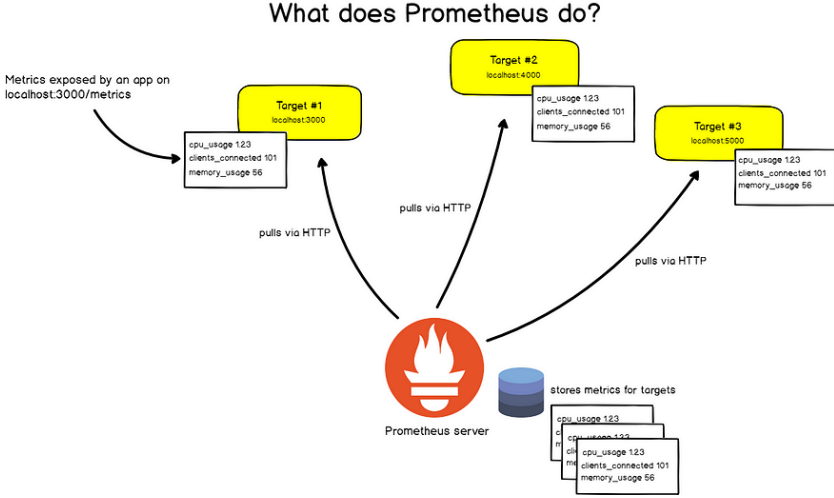
HAProxy Exporter

HAProxy yük dengeleyicisinden istatistikleri toplar ve Prometheus için uygun formata dönüştürür.

Bu dışı aktarıcılar, Prometheus için gerekli olan metrikleri sağlayarak, sistemlerin izlenmesini ve performansın analizini kolaylaştırır.

3. Veriler Nasıl Toplanır ve Saklanır?

Prometheus, verileri toplamak için "Pull modelini" kullanır. Bu yaklaşımda, Prometheus hedefleri olarak belirlenen uygulamalar ve sistemler, Prometheus'a veri göndermez. Bunun yerine, Prometheus hedeflerinden HTTP talepleriyle verileri alır. Bu yöntem, hedefler üzerindeki yükü azaltır ve Prometheus'a daha düşük bir yük getirir.



Şekil 40. Prometheus çalışma mantığı.

Prometheus, büyümeyi izlemek için çeşitli önceden tanımlanmış araçlar sunar. Bu araçlar arasında Node Exporter, Blackbox Exporter, Redis Exporter gibi çeşitli uygulamalar bulunur. Bu uygulamalar, belirli bir cihazın veya sistem bileşeninin ölçümelerini toplar ve Prometheus'un bu verilere HTTP aracılığıyla erişmesini sağlar.

Prometheus, hedefleri tarayarak ölçümleri toplar. Bu işlem, her hedef için ayrı bir kazıma işlemi gerçekleştirir ve ölçümleri zaman serilerine dönüştürür. Bu zaman serileri, etiketi, değeri ve zaman bilgisini içerir.

Prometheus'un hedefleri kazıma işlemi için kullandığı `scrape_interval` (kazıma aralığı) değeri bulunmaktadır. Bu değer, Prometheus'un hedefleri ne sıklıkla kazıdığını belirler. Varsayılan olarak, `scrape_interval` değeri 1 dakikadır ve genellikle yeterli bir aralıktır. Ancak `scrape_interval` değeri, ölçeklenebilirlik ve performansı optimize etmek için ayarlanabilir.

Prometheus'ta veri depolama ve sorgulama için etiketler önemli bir rol oynar. Özellikle zaman serileri, zaman etiketleri ve değerlerle ilişkilendirilen etiketler kullanılarak tanımlanır. Örneğin, bir HTTP istek hızı zaman serisini düşünelim:

Yöntem: GET, POST, PUT, DELETE gibi HTTP istek yöntemleri

Yol: İsteğin yapıldığı URL

Kod: HTTP yanıt kodu (200, 404, 500 vb.)

Bu etiketler, zaman içinde farklı istek yöntemleri, URL'ler ve yanıt kodları için istek hızlarını ayrı ayrı analiz etmenizi sağlar.

Bazı durumlarda, hedeflerin verileri taraması mümkün değildir. Mesela bir eylemin hemen tamamlanması gerektiği zamanlarda verileri push edebilmemizi sağlayacak push Gateway cihazına ihtiyaç duyulur.

Push Gateway, Prometheus'a veri göndermek için kullanılan bir araçtır. Bu araç, Prometheus'a gelen verileri hedeflerden HTTP POST istekleriyle alır ve geçici olarak saklar. Bu sayede, verilerin kaybolması engellenir ve hedefler yanıt veremezse bile veriler korunmuş olur.

Push Gateway'in faydalarından biri, hedeflerin ölçümelerini depolayabilmesidir. Bu, örneğin bir toplu iş veya zamanlanmış görevin sonuçlarını Push Gateway'e göndererek, bu verilerin daha sonra Prometheus tarafından alınabilmesini sağlar.

Genellikle önbellek verileri veya geçici iş sonuçları gibi durumlarda kullanılan Push Gateway, Prometheus entegrasyonunda esneklik sağlar ve veri toplama süreçlerini destekler.

2.14. Prometheus Verileri Nasıl Analiz Eder

PromQL, Prometheus'un zaman serisi verilerini dinamik olarak sorgulamak ve analiz etmek için özel olarak tasarlanmış bir sorgu dilidir. PromQL, kullanıcıların farklı zaman serileri arasında ilişkiler kurmalarını ve karmaşık metrikleri tanımlamalarını sağlar. Bu, kullanıcıların sistemlerinin davranışını daha derinlemesine anlamalarına ve potansiyel sorunları hızlı bir şekilde belirlemelerine yardımcı olur.

PromQL, Prometheus'un güçlü izleme yeteneklerini destekleyen önemli bir özelliktir. Örneğin, aşağıdaki basit bir PromQL sorgusuna bakalım:

```
avg_over_time(node_memory_Active_bytes[2m])/1024/1024/1024
```

Bu sorgu, `node_memory_Active_bytes` metriğinin son 2 dakikalık zaman dilimindeki ortalamasını hesaplar ve bu değeri bayttan gigabayta dönüştürmek için 1024'e bölerek işlem yapar.

Bir diğer örnek sorgumuz şu şekilde olsun

```
Ücret (http_requests_total{job="webserver"}[ 5 m])
```

Yukarıdaki sorguda , "http_requests_total" metrik adı kullanılarak belirli bir web sunucusuna ait zaman serileri "job" etiketiyle

sorgulanacaktır. Daha sonra, bu zaman serileri üzerinden "rate" fonksiyonu kullanılarak 5 dakikalık istek oranı hesaplanacaktır.

Bir örnek sorgu daha verecek olursak şu şekilde olsun

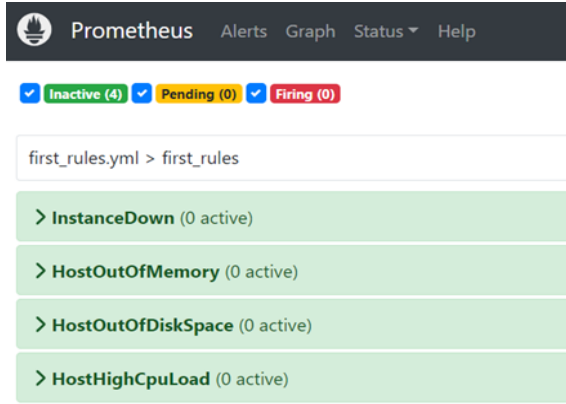
```
http_requests_total{job="web sunucusu"} +
http_requests_total{job="veritabanı sunucusu" }
```

Bu sorguda, "http_requests_total" metrik adı kullanılarak belirli bir web sunucusuna ve veritabanı sunucusuna ait zaman serileri, "job" etiketiyle toplanacaktır.

2.15. Prometheus Rules Nedir?

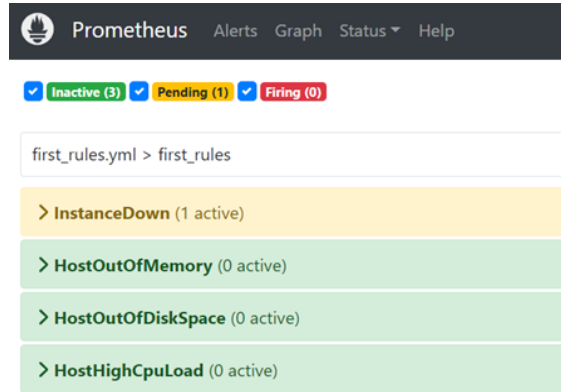
Prometheus'daki kurallar, metrik verilerini analiz etmek, işlemek ve uyarılar oluşturmak için kullanılan özelleştirilmiş kural setleridir. Bu kurallar, belirli bir zaman aralığında metriklerin durumunu izler ve belirli koşullar sağlandığında belirli aksiyonlar alır. Özetle, Prometheus kuralları, metrik verilerini otomatik olarak izlemek ve operasyonel işlemleri otomatize etmek için kullanılan bir mekanizmadır.

Şimdi first_rules adında bir kural dosyası oluşturmamız. Bunun için konsola "vi first_rules.yml" yazıyoruz. Masaüstündeki first_rules.yml dosyasının içeriğini kopyalayıp first_rules.yml dosyasının içerisine kaydediyoruz. Bu pencereyi de kaydedip kapatıyoruz. Ardından prometheus'u çalıştırıp kuralları kontrol edebiliriz. Bunun için "ip_adresi:9090" adresine gidip yukarıdaki kısımdan "Rules" kısmına basmalıyız. "Alerts" kısmına basarak uyarıları da görebiliriz.



Şekil 41. Prometheus'u konfigüre etme.

Eğer herhangi bir node_exporter'ı kapatırsak uyarı verir.



Şekil 42. Prometheus'u konfigüre etme.

E-posta bildirimi almak için **“alertmanager.yml”** dosyasını güncellemeliyiz. Masaüstündeki `email_alertmanager` dosyasının içeriğini buraya yapıştır.

alertmanager’ı başlat. `node_exporter`’lardan herhangi birisini kapat.

E postasını kontrol et.

2.16. Grafana Nedir?

Grafana, açık kaynaklı bir analiz ve görselleştirme platformudur. Kullanıcıların çeşitli kaynaklardan veri toplamasını, analiz etmesini ve anlamasını sağlayarak gerçek zamanlı ve etkileşimli görselleştirmeler oluşturur. Grafana, Prometheus, Elasticsearch, InfluxDB, MySQL, Azure Monitor gibi farklı veri kaynaklarıyla entegrasyon sağlar ve kullanıcıların grafikler, gösterge tabloları ve raporlar aracılığıyla verileri görsel olarak analiz etmesine olanak tanır.

2.17. Grafana’nın Özellikleri Nelerdir?

1. Görselleştirme

Grafana, verilerinizi görselleştirmek için geniş bir panel seçeneği sunar. Paneller, Grafana kontrol panelindeki temel görselleştirmeleri sağlayan yapı taşlarıdır. Her panel, belirli bir veri kaynağından gelen verileri göstermek için kullanılır. Paneller, çeşitli grafik türlerini (gösterge panelleri, histogramlar, çubuk grafikler vb.) veya günlükler ve uyarılar gibi bilgi portrelerini barındırabilir. Örneğin, Grafana’da Prometheus veri kaynağıyla ilişkilendirilmiş bir gösterge paneli oluşturabilirsiniz. Bu panel, Prometheus’ta saklanan CPU kullanım verilerini sorgular ve bu verileri görsel olarak temsil eder. Bu şekilde, kullanıcılar Prometheus’un sağladığı verilere dayalı olarak Grafana

üzerinde interaktif ve gerçek zamanlı olarak grafikler oluşturabilir ve verileri anlamlandırabilirler.

2. Dashboards

Grafana'nın kontrol panelleri, çeşitli kaynaklardan toplanan verilere yeni bir anlam düzeyi kazandırır. Bu gösterge tabloları daha sonra diğer ekip üyeleriyle ve diğer ekiplerle paylaşarak işbirliğine ve verilerin ve sonuçlarının daha kapsamlı araştırılmasına olanak sağlanır. Size ve ekibinize özel kontrol panelleri oluşturun ve gelişmiş sorgulama ve dönüştürme yeteneklerini kullanarak panellerinizi istediğiniz görselleştirmeleri oluşturacak şekilde özelleştirin.

Bir olayın veya beklenmeyen sistem davranışının nedenini mümkün olduğu kadar çabuk bulmaya çalışırken, ilgili tüm verileri ve veri ilişkilerini anlamak hayati önem taşır. Grafana, ekipler ve ekip üyeleri arasında verilerin kesintisiz görselleştirilmesine ve taşınmasına olanak tanır, böylece ekipler sorunun kökenine hızlıca inip sorunu çözebilir

3. Uyarı

Grafana, sistemlerinizde meydana gelen anormal durumları takip etmek ve haberdar olmak için birçok bildirim kanalıyla entegre olabilir. Bu bildirim kanalları arasında e-posta, Slack, PagerDuty ve daha fazlası bulunur.

Bir uyarı oluşturmak için öncelikle bir uyarı kuralı tanımlamanız gerekir. Bu kural, belirli bir durumun veya koşulun gerçekleştiğini tespit ederek bir uyarıyı tetikler. Grafana'da bir uyarı kuralı oluşturduktan sonra, bu kuralı bir tetikleyici olarak yapılandırabilirsiniz. Tetikleyici, belirli bir koşulun ihlal edilmesi durumunda uyarıyı tetikler.

Örneğin, CPU kullanımının belirli bir eşiği aştığında bir uyarı almak istiyorsanız, Grafana'da bir uyarı kuralı oluşturabilir ve bu kuralı CPU kullanımını izleyen bir tetikleyici olarak yapılandırabilirsiniz. Ardından, bu tetikleyiciyi e-posta veya Slack

gibi bir bildirim kanalıyla ilişkilendirerek, CPU kullanımı belirlenen eşiği aştığında otomatik olarak bir uyarı alabilirsiniz.

Bu şekilde, Grafana uyarı kuralı ve tetikleyici özellikleri sayesinde sisteminizdeki önemli durumları izleyebilir ve hızlı bir şekilde haberdar olabilirsiniz. Bu da sistem sağlığını korumanıza ve kesinti süresini azaltmanıza yardımcı olur.

4. Ek Açıklamalar

Grafana, grafiklere notlar eklemenize ve önemli noktaları işaretlemenize sağlar. Bu özellik, grafikler üzerinde açıklamalar eklemenizi ve önemli bilgileri vurgulamanızı kolaylaştırır. Bu notlar, gelecekteki eylemler için hatırlatma olarak kullanılabilir, yeni takım üyelerine bağlam sağlayabilir veya özel olayları grafiklerinizde belirtmenizi sağlar.

5. Açık Kaynak

Grafana tamamen açık kaynaklı bir projedir ve aktif bir topluluk tarafından desteklenmektedir. Bu, kullanıcılara kendi eklentilerini geliştirme ve yayınlama veya başkaları tarafından geliştirilen eklentileri kullanma esnekliği sağlar. Eklentilerin kurulumu genellikle kaynak kodunu indirme ve manuel olarak çalıştırma yoluyla kolaydır.

2.18. Docker Nedir?

Docker, yazılım uygulamalarını konteyner teknolojisi kullanarak paketlemeye, dağıtmaya ve çalıştırmaya olanak tanıyan bir platformdur. Docker konteynerleri, uygulamanın tüm bağımlılıklarını (kod, çalışma zamanı, sistem araçları, kütüphaneler vb.) tek bir birimde paketler ve bu birimlerin herhangi bir ortamda çalışmasını sağlar.

Docker konteynerleri, sanal makinelerden farklı olarak hafif ve hızlıdır. Sanal makineler, her biri kendi işletim sistemini çalıştıran ve kaynaklarını paylaşmayan ayrı ayrı işletim sistemleri üzerinde çalışırken, Docker konteynerleri, ana işletim sistemi çekirdeğiyle aynı kaynakları paylaşır ve bu nedenle daha az bellek ve işlemci gücü tüketirler.

2.19. Grafana ve Docker Kurulumu

1. Grafana'yı docker ile sanal makinemize kurmak için ilk önce docker'ı kurmalıyız. Aşağıdaki komutları prometheus'un kurulu olduğu sanal makinenin CLI'nda çalıştırınız.

1.1. Sistem Güncellemesi

```
`sudo apt update  
sudo apt upgrade`
```

1.2. Gerekli paketlerin Kurulumu

```
`sudo apt install apt-transport-https ca-certificates curl software-  
properties-common`
```

1.3. Docker GPG Anahtarı Ekleme

```
`curl -fsSL https://download.docker.com/linux/ubuntu/gpg |  
sudo apt-key add -`
```

1.4. Docker Deposunu Ekleme

```
`sudo add-apt-repository "deb [arch=amd64]  
https://download.docker.com/linux/ubuntu $(lsb_release -cs)  
stable"`
```

1.5. Paketi Güncelleme ve Docker Kurulumu

```
`sudo apt update  
sudo apt install docker-ce`
```

1.6. Docker Servisini Başlatma

```
`sudo systemctl start docker`
```

1.7. Docker'ın Başlangıçta Çalışmasını Sağlama:

```
`sudo systemctl enable docker`
```

1.8. Docker'ı Başlatıp Kontrol Etme:

```
`sudo systemctl status docker`
```

```
docker --version`
```

2. Grafana'yı kurmak için Grafana image dosyasını indirmeliyiz. Bunun için aşağıdaki kodu yazmalıyız. Eğer hata verirse başına sudo yazınız.

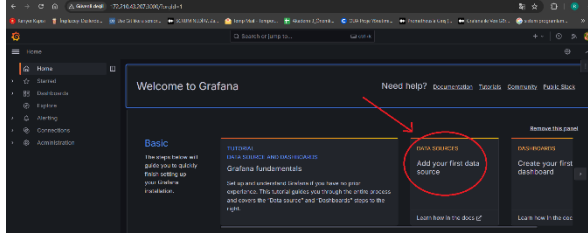
```
`docker run -d --name=grafana -p 3000:3000 -v grafana_config:/etc/grafana -v grafana_data:/var/lib/grafana -v grafana_logs:/var/log/grafana grafana/grafana`
```

3. Grafana 3000 portunu kullanır. Grafana'yı çalıştırmak için bu portu açmamız gerekiyor. Daha önceden Azure CLI yüklediğimiz için tekrar yüklememize gerek yok. O yüzden “az vm open-port --resource-group grup_adi --name vm_adi --port 3000 --priority 898” komutu ile 3000 portunu açabiliriz. Bu komutta eğer priority kısmına 900 olarak girersek 9090 portu ile öncelik sıraları çakışacaktı, 899 girseydik 9100 portu ile çakışacaktı. Bunun önüne geçmek için 898 yazıyoruz. Buradaki “grup_adi” yazan kısma ve “vm_adi” yazan kısma sanal sunucumuzun bilgilerinden ulaşabiliriz.

```
ridvanks@prometheus:~/prometheus-2.52.0-rc.1.linux-amd64$ az vm open-port --resource-group pr  
ometheus group --name prometheus --port 3000 --priority 898
```

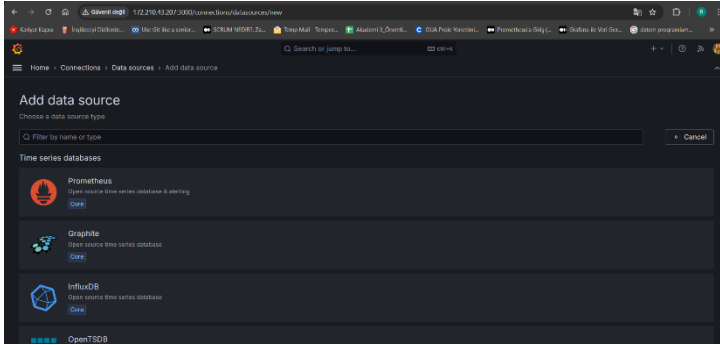
Şekil 43. Elasticsearch indirme sitesi.

4. Şimdi “ip_adresi:3000” adresine gidelim ve giriş yapalım. Default kullanıcı adı ve şifre admin’dir. Bu işlemi yaptıktan sonra prometheus’u, localde çalışan node_exporter’ı ve farklı bir sanal makinede olan node_exporter’ı çalıştıralım.
5. Grafana sitesine gidelim (ip_adresi:3000) yeni data source ekleyelim.



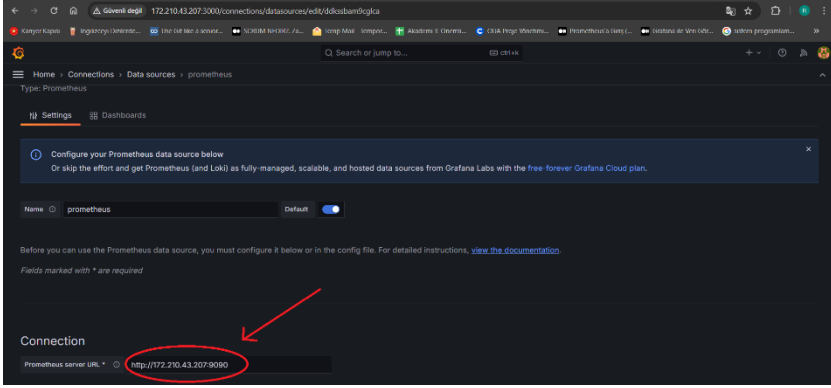
Şekil 44. Grafana ayarları.

6. Açılan ekrandan Prometheus’u seçelim.



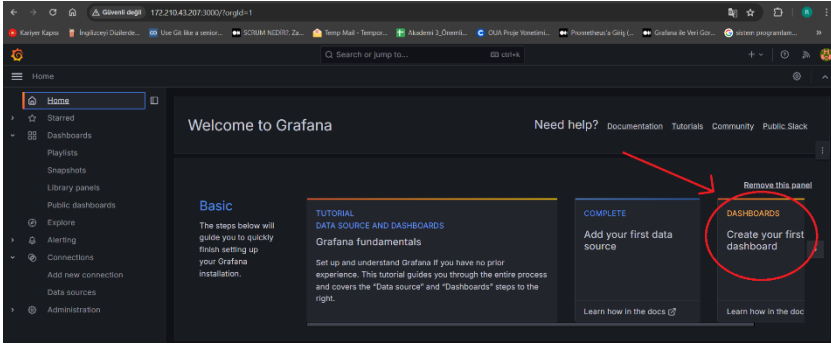
Şekil 45. Grafana veri ekleme.

7. Connection kısmına prometheus'un adresini yani "ip_adresi:9090" ifadesini yazalım. Aşağı kısımdan kaydedip ana sayfaya gidelim.



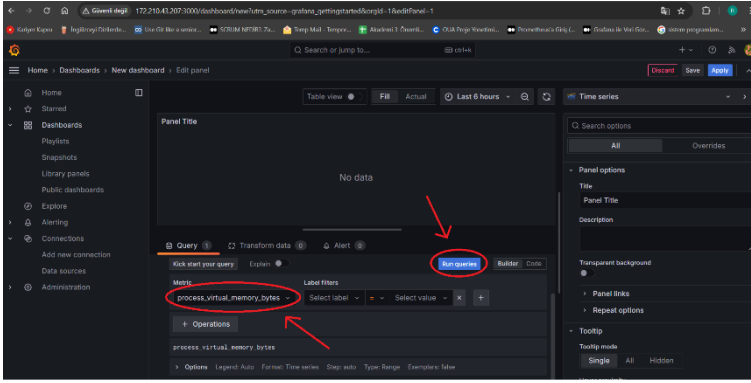
Şekil 46. Grafana IP adresi ayarları.

8. New dashboard kısmına girelim ve yeni bir dashboard ekleyelim.



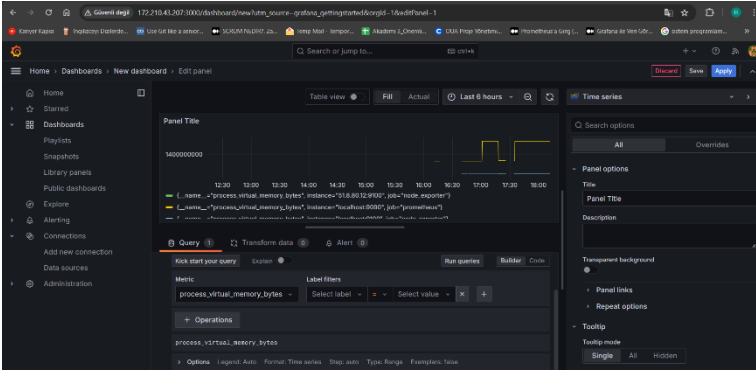
Şekil 47. Dashboard ekleme.

9. Query kısmının altındaki Metric yazan yere “process_virtual_memory_bytes” yazalım ve “Run Query” ye basalım. Sağ taraftaki pencereden istediğimiz görselleştirmeyi yapabiliriz. Save butonuna basalım, dashboard’a ve panele isim verip kaydedelim.



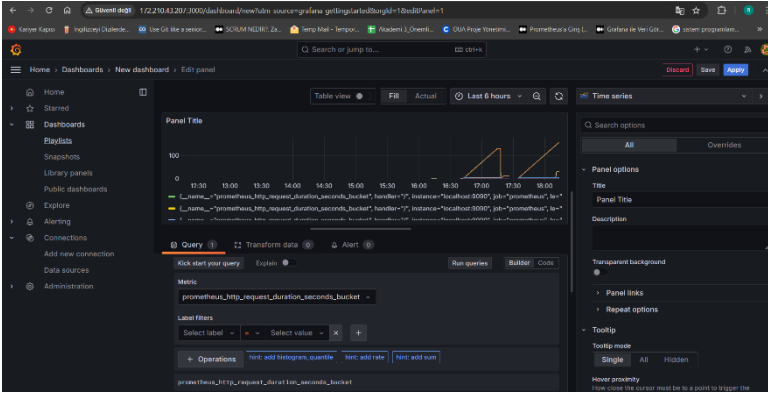
Şekil 48. Önbellek ayarları.

Bize aşağıdaki gibi bir grafik verecektir.



Şekil 49. Grafik çıktısı.

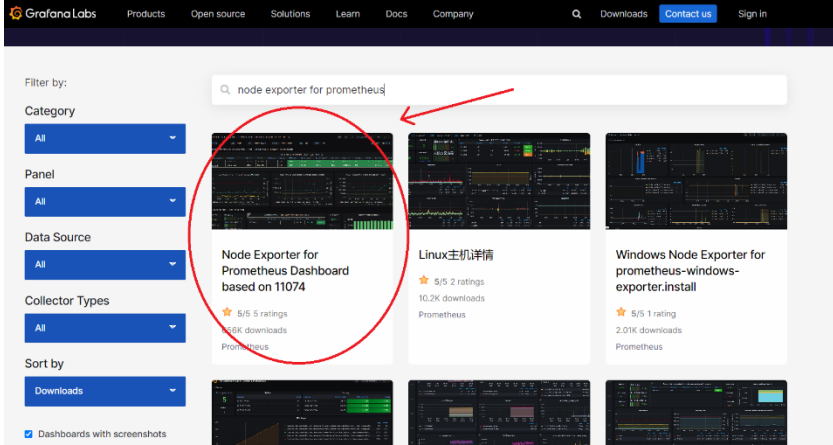
10. Yeni bir panel açalım ve Query kısmına “prometheus_http_request_duration_seconds_bucket” yazalım. Yine bu panele de isim verip kapatalım.



Şekil 50. İstek ayarlarının ayarlanması.

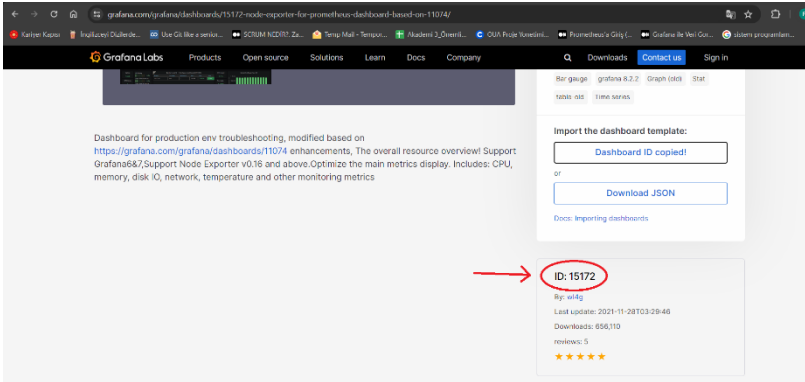
11. Yukarıdaki gibi dashboard eklersek uzun sürer ve bizi uğraştırır. Uğraşmamak için Google a “grafana dashboards” yazıyoruz ve resmi siteye gidiyoruz. Ardından arama kısmına “node exporter for prometheus” yazıyoruz ve tıklıyoruz.

İzleme ve Kayıtlama Sistemleri



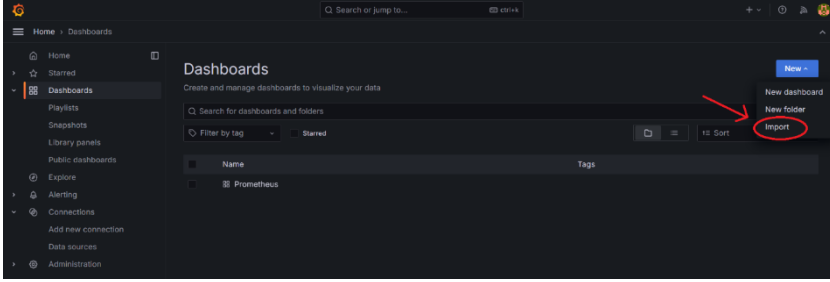
Şekil 51. Grafana sitesi.

12. Alt tarafta bulunan ID'yi kopyalıyoruz.



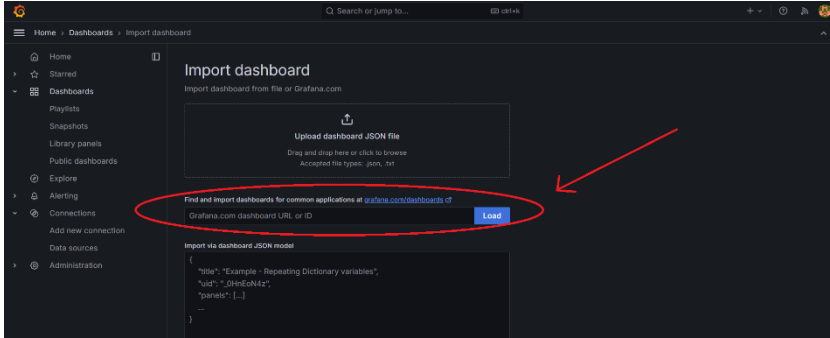
Şekil 52. Eklenti ID'si kopyalama.

13. Grafana sitesine(ip_adresi:3000) geri dönüyoruz ve “Dashboards” sekmesinde bulunan “New” butonuna basıyoruz. Ardından “Import” seçeneğini seçiyoruz.



Şekil 53. Dashboards konfigürasyonları.

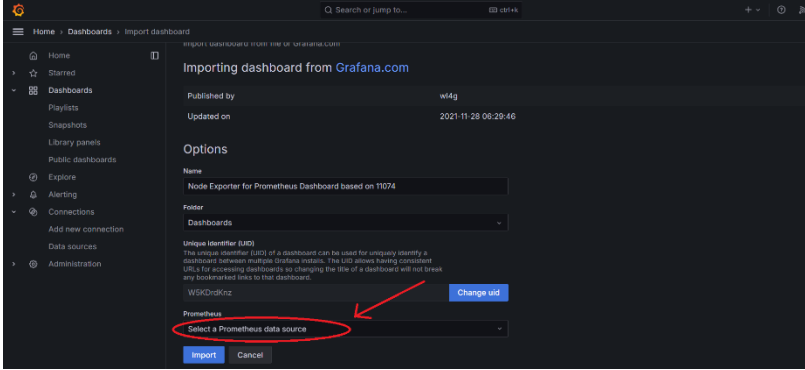
14. Grafana dashboard sitesinden seçtiğimiz dashboard'un linkini veya id'sini, aşağıdaki yere yapıştırıyoruz ve “Load” butonuna tıklıyoruz.



Şekil 54. Dashboard eklenmesi.

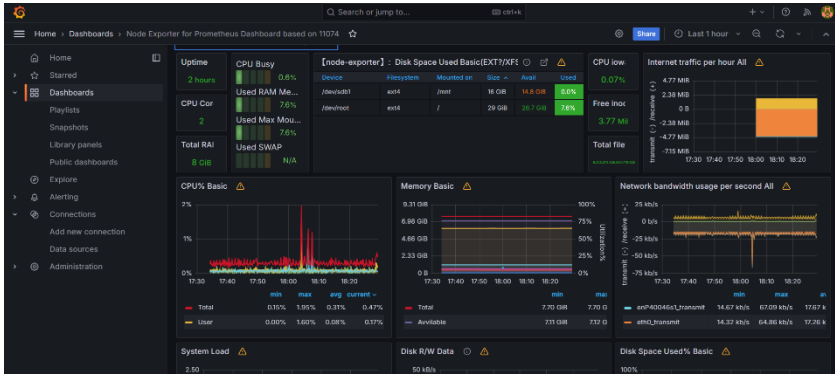
15. Ardından data source seçiyoruz ve import ediyoruz.

İzleme ve Kayıtlama Sistemleri



Şekil 55. Veri kaynağının seçimi.

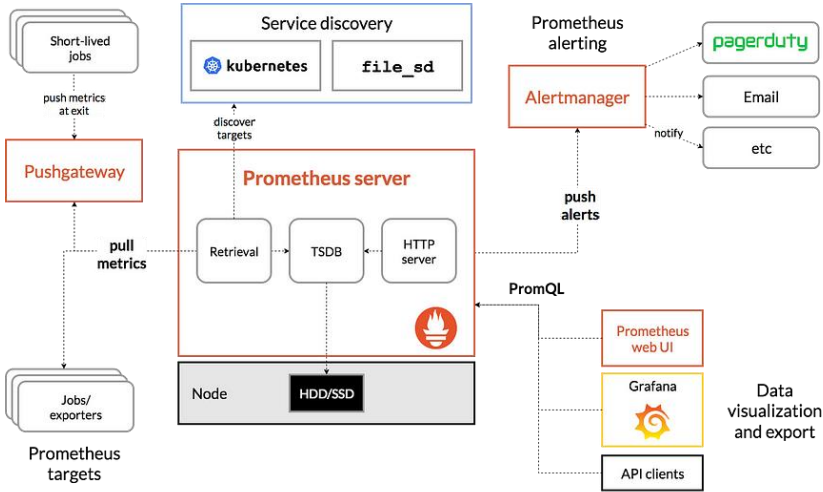
16. Grafikler otomatik olarak oluşturulacaktır.



Şekil 56. Grafiklerin son hali.

2.20. Alert Manager Nedir?

Alert Manager, Prometheus ekosisteminde kullanılan bir bileşendir ve Prometheus'un ürettiği uyarıları işleyip yöneterek belirlenen alıcılara iletilmesini sağlar. Alert Manager, Prometheus'un topladığı metrik verilere dayanarak tanımlanan uyarıları değerlendirir ve belirli bir eşiği aşan veya belirli bir durumu işaret eden uyarıları alır.



Şekil 57. Prometheus ve uyarı sisteminin şeması

2.21. Alert Manager Kurulumu

1. Alert manager'ı kurmak için ilk önce "prometheus.io" sayfasına gidin ve "DOWNLOAD" kısmını açın. Açılan sayfayı aşağıya doğru kaydırın ve "alertmanager" kısmını bulun. Sanal sunucumuz-

sunucumuzun bilgilerinden ulaşabiliriz. Bu işlemin ardından alertmanager'ı “./alertmanager” komutu ile başlatın. Alert manager’a “ip_adresi:9093” ifadesi ile bağlanabiliriz.

```
ridvanks@prometheus:~/alertmanager-0.27.0.linux-amd64$ az vm open-port --resource-group prometheus group --name prometheus --port 9093 --priority 897
```

Şekil 61. Port ayarlamaları.

5. “prometheus.yml” dosyasını açalım ve içeriğini aşağıdaki gibi değiştirelim.

ÖNCESİ

```
# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'
'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"
```

Şekil 62. .yaml dosyası değişiklik öncesi

SONRASI

```
# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
      - targets:
        - 'localhost:9093'

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'
'.
rule_files:
  - first_rules.yml
  # - "second_rules.yml"
```

Şekil 63. .yaml dosyası değişiklik sonrası.

Değişiklikleri kaydedip çıkıyoruz.

6. Şimdi “first_rules” adında bir kural dosyası oluşturmamız. Bunun için konsola “vi first_rules.yml” yazıyoruz. İçerisine aşağıdaki ifadeleri yazıyoruz.

```
groups:
- name: first_rules
  rules:
  - alert: InstanceDown
    expr: up == 0
    for: 1m
    labels:
      severity: "critical"
    annotations:
      summary: "Endpoint {{ $labels.instance }} down"
      description: "{{ $labels.instance }} of job {{ $labels.job }} has been down for more than 1 minutes."

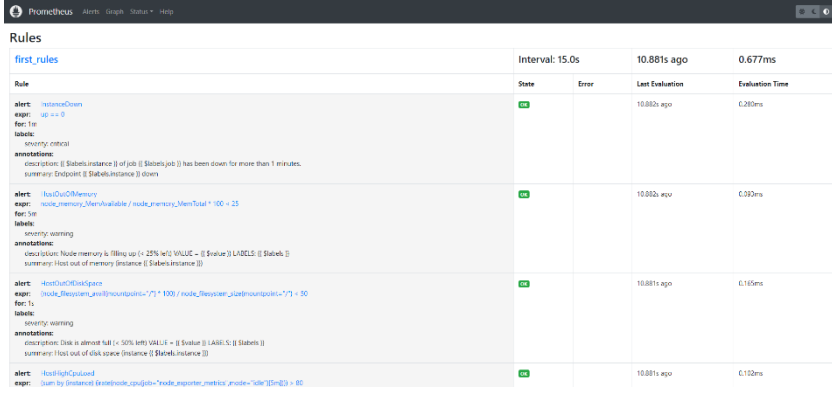
  - alert: HostOutOfMemory
    expr: node_memory_MemAvailable / node_memory_MemTotal * 100 < 25
    for: 5m
    labels:
      severity: warning
    annotations:
      summary: "Host out of memory (instance {{ $labels.instance }})"
      description: "Node memory is filling up (< 25% left)\n VALUE = {{ $value }}\n LABELS: {{ $labels }}"

  - alert: HostOutOfDiskSpace
    expr: (node_filesystem_avail{mountpoint="/" } * 100) / node_filesystem_size{mountpoint="/" } < 50
    for: 1s
    labels:
      severity: warning
    annotations:
      summary: "Host out of disk space (instance {{ $labels.instance }})"
      description: "Disk is almost full (< 50% left)\n VALUE = {{ $value }}\n LABELS: {{ $labels }}"

  - alert: HostHighCpuLoad
    expr: (sum by (instance) (irate(node_cpu{job="node_exporter_metrics",mode="idle"}[5m]))) > 80
    for: 5m
    labels:
      severity: warning
    annotations:
      summary: "Host high CPU load (instance {{ $labels.instance }})"
      description: "CPU load is > 80%\n VALUE = {{ $value }}\n LABELS: {{ $labels }}"
```

Şekil 64. Kuralların belirlenmesi.

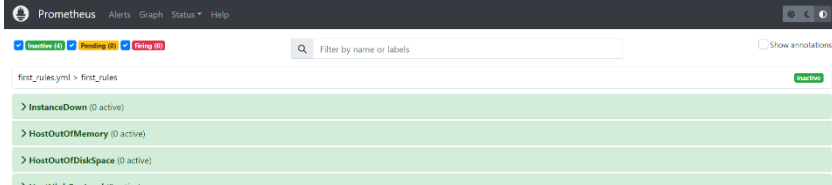
Bu pencereyi de kaydedip kapatıyoruz. Ardından prometheus’u çalıştırıp kuralları kontrol edebiliriz. Bunun için “ip_adresi:9090” adresine gidip yukarıdaki kısımdan “Rules” kısmına basmalıyız.



Rules		Interval: 15.0s	10.881s ago	0.677ms
Rule	State	Error	Last Evaluation	Evaluation Time
Rule alert: InstanceDown expr: up == 0 for: 1m labels: severity: critical annotations: description: {{ \$labels.instance }} of job {{ \$labels.job }} has been down for more than 1 minutes. summary: Endpoint {{ \$labels.instance }} down	on		10.882s ago	6.280ms
Rule alert: HostOutOfMemory expr: node_memory_MemAvailable / node_memory_MemTotal * 100 < 25 for: 5m labels: severity: warning annotations: description: Node memory is filling up (+ 25% left) VALUE = {{ \$value }} LABELS: {{ \$labels }} summary: Host out of memory (instance {{ \$labels.instance }})	on		10.882s ago	0.392ms
Rule alert: HostOutOfDiskSpace expr: (node_filesystem_avail{mountpoint="/"} * 100) / node_filesystem_size{mountpoint="/"} < 30 for: 1s labels: severity: warning annotations: description: Disk is almost full (+ 30% left) VALUE = {{ \$value }} LABELS: {{ \$labels }} summary: Host out of disk space (instance {{ \$labels.instance }})	on		10.881s ago	6.165ms
Rule alert: HostHighCpuUsed expr: (sum by (instance) (rate(node_cpu{mode="idle", exporter="node_exporter_metrics", instance=~"\$labels.instance"}[5m]) * 100) > 80)	on		10.881s ago	6.112ms

Şekil 65. Kurallar arayüzü.

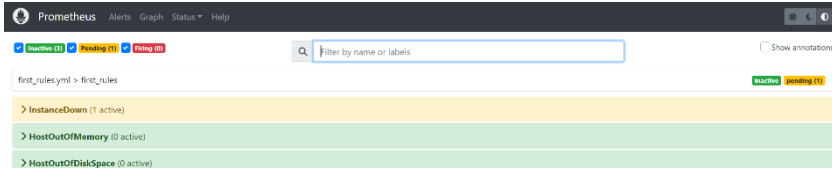
7. “Alerts” kısmına basarak uyarıları da görebiliriz.



Alert	State	Labels	Annotations
InstanceDown (0 active)	active		
HostOutOfMemory (0 active)	active		
HostOutOfDiskSpace (0 active)	active		

Şekil 66. Var olan kurallar.

8. Eğer herhangi bir node_exporter’ı kapatırsak uyarı verir.



Alert	State	Labels	Annotations
InstanceDown (1 active)	pending		
HostOutOfMemory (0 active)	active		
HostOutOfDiskSpace (0 active)	active		

Şekil 67. Kural devre dışı bırakmak.

2.22. Alert Manager İle E-Posta Bildirimi Almak

1. E-posta bildirimi almak için “alertmanager.yml” dosyasını güncellemeliyiz. İçerisini aşağıdaki gibi güncelleyin. Buradaki “auth_password” kısmına yazmak için uygulama şifresi almalısınız. Bunun için bir sonraki adımdan devam edin.

ÖNCESİ

```
route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 1h
  receiver: 'web.hook'
receivers:
- name: 'web.hook'
  webhook_configs:
    - url: 'http://127.0.0.1:5001/'
inhibit_rules:
- source_match:
    severity: 'critical'
  target_match:
    severity: 'warning'
  equal: ['alertname', 'dev', 'instance']
```

Şekil 68. Uyarı sisteminin konfigürasyon öncesi.

SONRASI

```
global:
  resolve_timeout: 1m

route:
  group_by: ['alertname']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 1h
  receiver: 'email-notifications'

receivers:
- name: 'email-notifications'
  email_configs:
  - to: karasubasiridvan@gmail.com
    from: karasubasiridvan@gmail.com
    smarthost: smtp.gmail.com:587
    auth_username: karasubasiridvan@gmail.com
    auth_identity: karasubasiridvan@gmail.com
    auth_password: [REDACTED]
    send_resolved: true

inhibit_rules:
- source_match:
  severity: 'critical'
  target_match:
  severity: 'warning'
  equal: ['alertname', 'dev', 'instance']
```

Şekil 69. Uyarı sisteminin konfigürasyon sonrası.

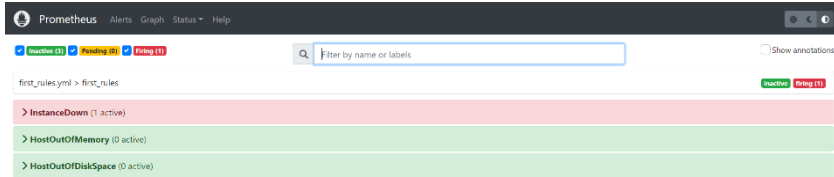
2. Uygulama şifresi almak için gmail hesabınıza girin ve sağ üstten hesap resminize basın. Açılan sekmeden “Google Hesabınızı yönetin” butonuna basın. Ardından arama çubuğuna “uygulama şifreleri” yazın.

İzleme ve Kayıtlama Sistemleri



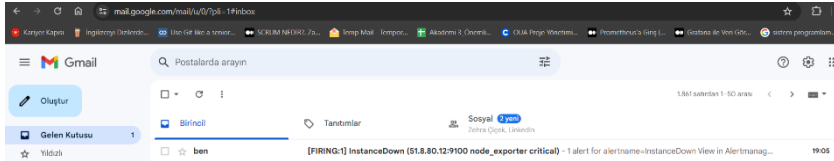
Şekil 70. Google uygulama şifreleri.

3. Bir tane uygulama şifresi oluşturun ve uygulama şifrenize isim verin. Ardından bu şifreyi “alertmanager.yml” dosyasının içerisindeki “auth_password” kısmına yapıştırın.
4. Şimdi prometheus’u, localde çalışan node_exporter’u, sanal makinede çalışan node_exporter’u ve alertmanager’ı çalıştırın. node_exporter’lardan herhangi birisini kapatın. “Alerts” sayfasına girin. Bir uyarı alacaksınız.



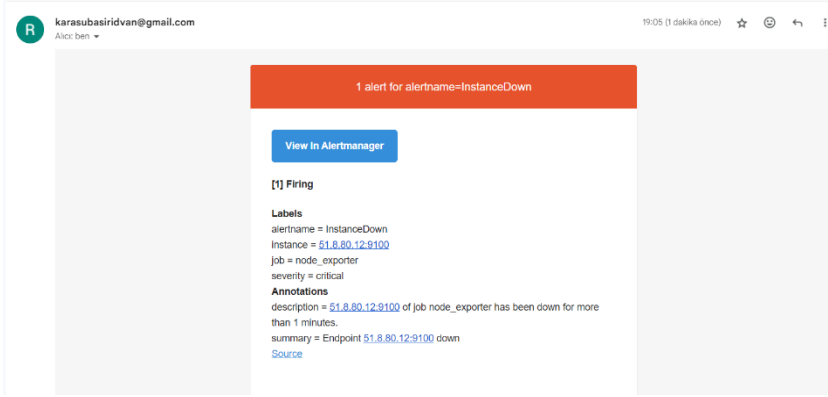
Şekil 71. Uyarı sisteminin çalıştırılması

Şimdi e-postanızı kontrol edin. Bir uyarı e-postası alacaksınız.



Şekil 72. Uyarı e-postası.

E-posta içeriği



Şekil 73. Uyarı e-postası içeriği.

8. ELK STACK NEDİR?

ELK Stack diğer bir adıyla Elastic Stack olarak bilinir. ELK Stack, Elasticsearch, Logstash ve Kibana'nın kısaltmasına verilen addır. ELK Stack, uygulama ve sistemlerimizdeki kayıtları, toplamak, işlemek, depolamak, analiz etmek ve görselleştirmek için kullanılır. Bu, sistem yöneticilerine, geliştiricilere ve işletmelere daha iyi bir anlayış ve izleme sağlayarak hızlı sorun giderme, performans iyileştirmeleri, güvenlik analizi ve daha fazlasını yapmalarına yardımcı olur.

8.1. Elasticsearch'e Giriş

ELK Stack'in en önemli parçası olan Elasticsearch , kurucusu Shay Banon tarafından ilk versiyonunu Şubat 2010 yılında açık kaynak kodlu olarak yayınladı. Elasticsearch , Java programlama dili ile “Apache Lucene” altyapısı üzerine geliştirilmiş “NoSQL” veri tabanıdır. Elasticsearch 'ün çıkış amacı “Big Data” yani büyük veri olarak adlandırılan verileri analiz etme ve arama yapmak için çıkmıştır. Kısaca Elasticsearch 'e analiz ve metin arama aracı diyebiliriz.

8.2. Elasticsearch Bileşenleri

Node

Elasticsearch 'ün çalışan clusterin parçası olan tek bir sunucudur. Sunucunun adını ifade eder. Tek fiziksel ve sanal sunucu, RAM, depolama ve işlem gücü gibi fiziksel kaynaklarının özelliklerine bağlı olarak birden fazla node barındırır. Clusterlar arama ve indeksleme gibi yetenekleri nodelar (düğümler) sayesinde gerçekleşir

Cluster

ElasticSearch 'te cluster birbirine bağlı bir veya daha fazla node bir araya gelerek oluşturduğu gruptur. Cluster, tüm veriler için tüm nodelarda toplu indeksleme ve arama yetenekleri sağlar.

Index

Index, ElasticSearch 'te sorgulama yapabileceğiniz en üst düzey varlıktır. Indeks'in ilişkisel veri tabanı şemasındaki bir veri tabanına benzer olduğunu düşünebilirsiniz. Farklı türde belgelerin ve bunların özelliklerinin bir koleksiyonudur.

Shard

ElasticSearch, bir indeksi birden fazla parçaya, yani shard'a bölerek veriyi yatay olarak parçalar. Her shard, belgenin tüm özelliklerini içerir ancak dizinden daha az JSON nesnesi içerir. Bu yatay bölme, her shard'ın bağımsız bir düğümde depolanabileceği anlamına gelir. İlk shard, bir dizinin orijinal yatay bölümüdür ve daha sonra bu ana shard'lar kopyalanarak çoğaltılır. Shard'lar, ElasticSearch kümesindeki iş yükünü dağıtarak ve paralelleştirerek performansı artırır. Bu sayede, yeni makineler eklenerek ElasticSearch kümesi ölçeklenebilir.

Replicas

Verilerin her birinin kopyasının bulunduğu başka makinelerdir. ElasticSearch , kullanıcının indekslerin parçalarının bir veya daha fazla kopyasını oluşturmanıza olanak tanır. Çoğaltma, yalnızca arıza durumunda verilerin kullanılabilirliğini artırmaya yardımcı olmakla kalmaz, aynı zamanda bu kopyalarda paralel arama işlemi gerçekleştirerek arama performansını da artırır.

Document

ElasticSearch 'te "Type" yapısı içerisinde yer alan satırları temsil eder. JSON ile ifade edilir. Type'ler bu yapılardan oluşur.

Type'ler birden fazla dokümana sahiptir. Kısaca dokümanlara sisteme verdiğimiz JSON dosyaları diyebiliriz.

Type

İlişkisel veri tabanlarındaki tablolar olarak adlandırılabilir. Aynı indeks içerisinde birden fazla type barındırabilir. İndeks'e koyduğumuz verileri kategorize etmemize yarar.

8.3. Elasticsearch Kurulumu

ElasticSearch 'ü indirmek oldukça basittir. ElasticSearch için en az Java 8 Oracle JDK sürüm 1.8.0_131 önerilir. Kurulumu başlamadan önce kurulum yapacağınız işletim sisteminin ElasticSearch sürümünü <https://www.elastic.co/support/matrix> sitesinden destekleyip desteklemediğini kontrol ediniz.

Download Elasticsearch

1 Download and unzip Elasticsearch

Choose platform:

Windows

Windows

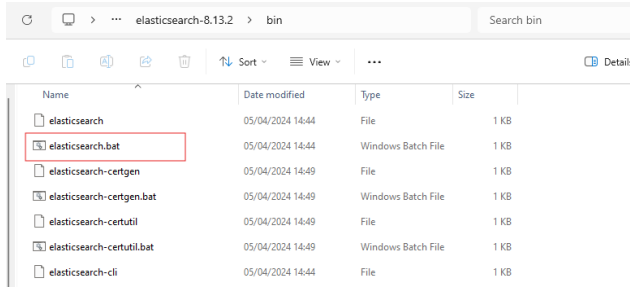
sha

asc

Şekil 74. Elasticsearch indirme sitesi.

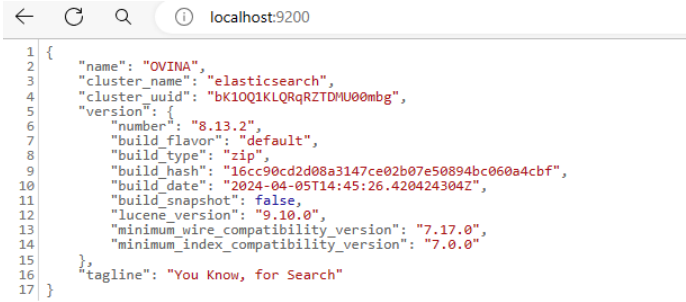
ElasticSearch kurulumu için öncelikle <https://www.elastic.co/downloads/elasticsearch> açıp hangi işletim sistemine indirmek istiyorsanız o platformu seçin. Biz Windows üzerinden işlem yapacağız. Ardından Windows yazılı olan indirme

butonuna tıklayın. İndirmiş olduğunuz dosyanın içerisinde bulunan bin dizininde “elasticsearch.bat” dosyası yer almaktadır. Bu dosyayı yönetici olarak çalıştırdıktan sonra Elasticsearch 'ün kurulumu tamamlanacaktır.



Şekil 75. Klasör içeriği.

Eğer kurulumu doğru yaptıysanız, tarayıcınıza <http://localhost:9200/> yazdığınızda karşınıza aşağıdaki gibi bir ekran gelmelidir.



```
1 {
2   "name": "OVINA",
3   "cluster_name": "elasticsearch",
4   "cluster_uuid": "bK10Q1KLQRqRZTDMU00mbg",
5   "version": {
6     "number": "8.13.2",
7     "build_flavor": "default",
8     "build_type": "zip",
9     "build_hash": "16cc90cd2d08a3147ce02b07e50894bc060a4cbf",
10    "build_date": "2024-04-05T14:45:26.420424304Z",
11    "build_snapshot": false,
12    "lucene_version": "9.10.0",
13    "minimum_wire_compatibility_version": "7.17.0",
14    "minimum_index_compatibility_version": "7.0.0"
15  },
16   "tagline": "You Know, for Search"
17 }
```

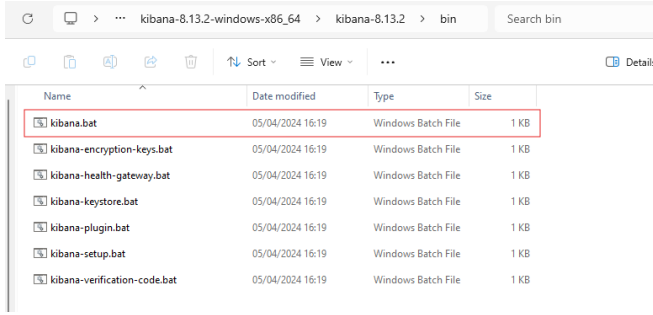
Şekil 76. Localhost:9200.

8.4. Kibana ve Kibana Kurulumu

Kibana'nın ilk sürümünü 2013 yılında Rashid Khan tarafından geliştirildi. Kibana ELK Stack'te görselleştirme görevini üstlenir. Kibana kullanımının temel amacı veri görselleştirmektir. Elasticsearch'te bulunan verileri analiz etmek için kullanıcı dostu bir arayüz sağlar. Kibana geliştiricilere ve kullanıcılara verileri hakkında bilgi edinebilmesi için tablolar, grafikler ve rapor gibi araçlar oluşturur.

Kibana indirmek oldukça basittir. Kurulumu başlamadan önce kurulum yapacağınız işletim sisteminin Kibana sürümünü <https://www.elastic.co/support/matrix> sitesinden destekleyip desteklemediğini kontrol ediniz. Bu sayede beklenmedik hatalar almazsınız.

Download Kibana



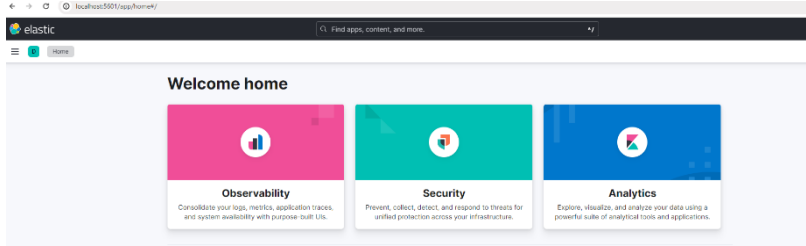
Şekil 77. Kibana kurulum dosyası.

Kibana kurulumu için öncelikle <https://www.elastic.co/downloads/kibana> açıp hangi işletim sistemine indirmek istiyorsanız o platformu seçin. Biz Windows üzerinden işlem yapacağız. Ardından Windows yazılı olan indirme butonuna tıklayın. İndirmiş olduğunuz dosyanın içerisinde bulunan bin dizininde “kibana.bat” dosyası yer almaktadır. Bu dosyayı yönetici olarak çalıştırdıktan sonra Kibana kurulumu tamamlanacaktır.

Eğer kurulumu doğru yaptıysanız, tarayıcınıza <http://localhost:5601/> yazdığınızda karşınıza aşağıdaki gibi bir ekran gelmelidir.

8.5. Logstash Nedir ve Logstash Kurulumu?

Logstash ilk olarak Jordan Sissel tarafından piyasaya sürüldü. Logstash ELK Stack'te veriyi belirtilen kaynaktan çekip toplama ve işleme görevini üstlenir. Logstash, sistem kayıtları, web sitesi kayıtları, sunucu kayıtları gibi kaynaklardan verileri kolayca almamızı sağlayan



Şekil 78. Kibana arayüzü.

bir veri alma aracı diyebiliriz. Logstash'te sunulan önceden oluşturulmuş filtreler sayesinde veri türlerini kolayca dönüştürebiliriz ve Elasticsearch'te indekslemeye ve özel veri dönüştürmelerine gerek kalmadan sorgulamaya başlayabilmemizi sağlar.

Logstash, veriyi bir pipeline (veri akış hattı) üzerinde işler ve birbirinden bağımsız birden çok pipeline'ı çalıştırabilir.

Logstash'te 3 temel plugin kullanılmaktadır. Bunlar Input, filter ve output pluginleridir.



Şekil 79. Logstash yapısı.

Input

Input, belirli kaynaktan veriyi çekmek için kullanılır. Örneğin input, Beats veya Windows kayıt olsun. Logstash bu durumda Beats'e veya Windows Kayıt'a gelen her veriyi okur ve filtrelemesi için filter pluginine yönlendirir. Yaygın olarak kullanılan input pluginleri: File, Http, Csv, Kafka, Beats, ElasticSearch, S3, syslog vb.

Filter

Filter ile inputdan gelen verileri filtreleme işlemi yapılır. Bu sayede istenmeyen bazı veriler veya parametler silinebilir ya da değerlerin veya formatın değiştirilmesi gibi işlemler yapılabilir. Veriler filtrelendikten sonra outputa yönlendirilir. Yaygın olarak kullanılan filter pluginleri: Clone, Csv, Grok, JSON vb.

Output

Output, veriler filtrelendikten sonra verilerin çıktısını nereye verileceğini belirler. Yaygın olarak kullanılan output pluginleri: ElasticSearch , File, Http, Csv vb.

Logstash Kurulumu

Logstash indirmek oldukça basittir. Kurulumu başlamadan önce kurulum yapacağınız işletim sisteminin Logstash sürümünü <https://www.elastic.co/support/matrix> sitesinden destekleyip desteklemediğini kontrol ediniz. Bu sayede beklenmedik hatalar almazsınız.

Logstash kurulumu için öncelikle <https://www.elastic.co/downloads/logstash> açıp hangi işletim sistemine indirmek istiyorsanız o platformu seçin. Biz Windows üzerinden işlem yapacağız. Ardından Windows yazılı olan indirme butonuna tıklayın.

Download Logstash

1 Download and unzip Logstash

Choose platform:

Windows

Windows

sha asc

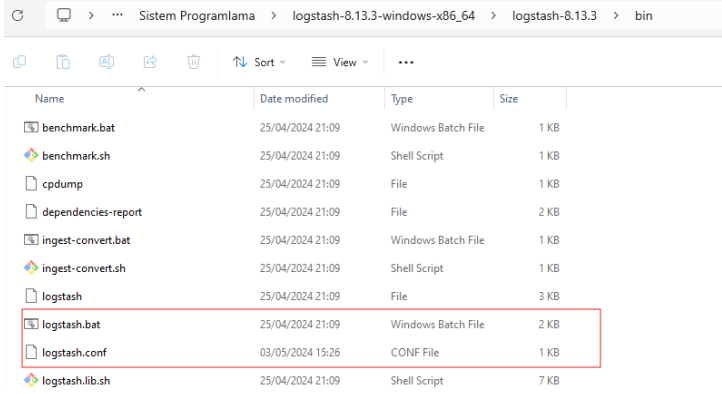
Şekil 80. Logstash indirme sitesi.

Kurulumu yaparken Logstash'i yapılandırmamız gerekiyor bunun için logstash.conf adında yapılandırma dosyası hazırlamamız gerekiyor. Şekilde verilen yapılandırma kodlarını logstash.conf dosyası içerisine yazıyoruz. Ardından oluşturduğumuz dosyayı, Logstash için indirmiş olduğumuz klasörde bulunan "bin" dizinine taşıyoruz. logstash.conf dosyası ile aynı dizinde bulunmalıdır.

```
input {  
  stdin {  
  }  
}  
  
output {  
  stdout {  
  }  
}
```

Şekil 81. Konfigürasyon dosyası.

Son olarak "bin" dizininde terminal ekranını açarak, logstash -f logstash.conf yazarak çalıştırıyoruz. Bu işlemle Logstash, belirtilen yapılandırma dosyasını okur ve belirtilen stdout'a çıktı verir.



Name	Date modified	Type	Size
benchmark.bat	25/04/2024 21:09	Windows Batch File	1 KB
benchmark.sh	25/04/2024 21:09	Shell Script	1 KB
cpdump	25/04/2024 21:09	File	1 KB
dependencies-report	25/04/2024 21:09	File	2 KB
ingest-convert.bat	25/04/2024 21:09	Windows Batch File	1 KB
ingest-convert.sh	25/04/2024 21:09	Shell Script	1 KB
logstash	25/04/2024 21:09	File	3 KB
logstash.bat	25/04/2024 21:09	Windows Batch File	2 KB
logstash.conf	03/05/2024 15:26	CONF File	1 KB
logstash.lib.sh	25/04/2024 21:09	Shell Script	7 KB

Şekil 82. Logstash kurulum dosyaları.

8.6. Beats

Beats, veri göndericilere yönelik ücretsiz bir platformdur. Binlerce sistemden Logstash veya Elasticsearch'e veri gönderir. Beats, sunucularda veya kapsayıcılarda (container) çalıştırılabilen bir veri toplama aracıdır.

8.7. Filebeat

Günlükler ve diğer veriler için hafif göndericidir. Bilgisayar, bulut veya güvenlik cihazları gibi cihazlardan fark etmeksizin Filebeat, belirtilen kayıt dosyalarını izler, toplar ve bunları indeksleme için Elasticsearch veya Logstash'e gönderir.

8.8. Winlogbeat

Windows da bulunan olay günlüklerini Logstash veya Elasticsearch'e iletir.

8.9. Metricbeat

Bu hafif gönderici, sistemlerinizden ve hizmetlerinizden ölçümler toplar. Ölçümler, CPU kullanımı, bellek kullanımı, disk kullanımı ve daha fazlasını içerebilir. Metricbeat, çeşitli sistemler ve hizmetler için istatistikleri toplamak ve bunları merkezi bir konuma göndermek için kullanılır. Örneğin, Redis, NGINX, Apache, MongoDB gibi hizmetlerden istatistikleri toplamak için kullanılabilir.

2.10. Packetbeat

Ağ verileri için hafif bir göndericidir. Ağ trafiği üzerindeki paketlerden elde edilen verileri toplar.

2.11. Auditbeat

Denetim verileri için kullanılan bir hafif göndericidir. Sistemlerde yapılan değişiklikler, yetkilendirme ve kimlik doğrulama etkinlikleri gibi denetim izleri toplar.

2.12. Heartbeat

Bu, hizmetlerin çalışma sürelerini izlemek için kullanılan bir hafif göndericidir. Belirli bir hizmetin erişilebilirliğini ve performansını izler ve belirli bir süre içinde hizmete erişilebilir olup olmadığını kontrol eder.

3. UYGULAMA

3.1. Senaryo

Bu uygulama, Beats'in Winlogbeat hafif göndericisini kullanarak Windows Olay Görüntüleyicisi'nden kayıtları toplayarak, Elasticsearch'e göndermeyi ve ardından Kibana üzerinde analiz edip görselleştirmeyi hedeflemektedir.

3.2. Winlogbeat Kurulumu:

Winlogbeat indirmek oldukça basittir. Kurulumu başlamadan önce kurulum yapacağınız işletim sisteminin Winlogbeat sürümünü <https://www.elastic.co/support/matrix> sitesinden destekleyip desteklemediğini kontrol ediniz. Bu sayede beklenmedik hatalar almazsanız.

Winlogbeat kurulumu için öncelikle <https://www.elastic.co/downloads/beats/winlogbeat> açıp hangi işletim sistemine indirmek istiyorsanız o platformu seçin. Ardından Windows yazılı olan indirme butonuna tıklayın.

Download Winlogbeat

1 Download and unzip Winlogbeat

Choose platform:

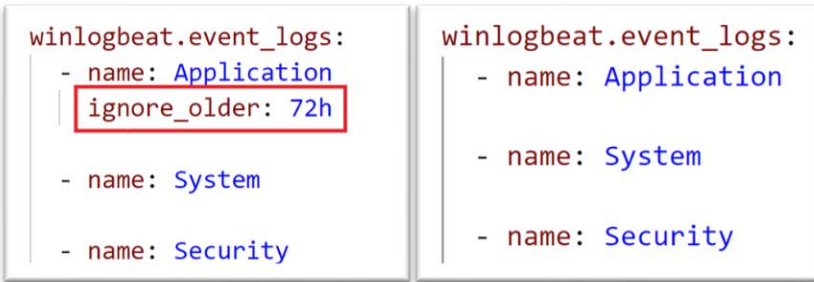
Windows ZIP x86_64

Windows ZIP x86_64 sha asc

data	04/05/2024 14:38	File folder	
kibana	30/04/2024 04:36	File folder	
logs	04/05/2024 14:31	File folder	
module	30/04/2024 04:36	File folder	
build_hash.txt	30/04/2024 04:36	Text Document	1 KB
fields.yml	30/04/2024 04:36	YAML File	405 KB
install-service-winlogbeat.ps1	30/04/2024 04:36	Windows PowerS...	1 KB
LICENSE.txt	30/04/2024 04:36	Text Document	14 KB
NOTICE.txt	30/04/2024 04:36	Text Document	2,971 KB
README.md	30/04/2024 04:36	MD File	1 KB
uninstall-service-winlogbeat.ps1	30/04/2024 04:36	Windows PowerS...	1 KB
winlogbeat.exe	30/04/2024 04:36	Application	75,509 KB
winlogbeat.reference.yml	30/04/2024 04:36	YAML File	64 KB
winlogbeat.yml	04/05/2024 14:36	YAML File	8 KB
winlogbeat-oid.yml	04/05/2024 14:36	YAML File	8 KB

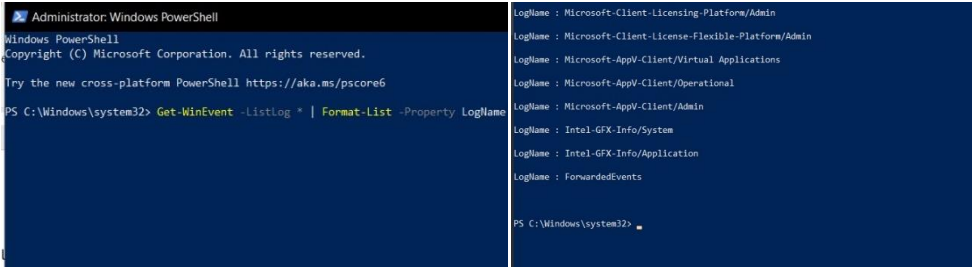
Şekil 83. Winlogbeat indirme sitesi ve kurulum dosyaları.

Windows kayıtlarını alabilmemiz için Winlogbeat'i yapılandırmanız gerekiyor. Bu sebeple winlogbeat.yml adındaki yapılandırma dosyası üzerinde değişiklik yapmamız gerekiyor. İlk olarak şekilde kutu ile gösterilen “ignore_older: 72h” etiketini kaldırıyoruz. Bu komut silinmezse 72 saat öncesi log kayıtlarını alamayız.



Şekil 84. Etiketin önceki ve sonraki hali.

Ardından Powershell'i yönetici olarak çalıştırıp, `Get-WinEvent -ListLog * | Format-List -Property LogName` komutunu yürütüyoruz. Bu komut Windows olay kayıtlarını almamızı ve konsol ekranına kayıtların adlarını yazdırmamızı sağlar. Böylelikle istediğimiz kayıtları görebiliriz.



Şekil 85. Powershell ile Windwos logların alınması.

Windows kayıtlarını bulduktan sonra winlogbeat.yml dosyamıza “-name: HardwareEvents” etiketini ekliyoruz. HardwareEvents, donanım olaylarına ilişkin bilgileri içerir. Donanım hataları, sürücü güncellemeleri, donanım değişiklikleri gibi donanım olaylarını izlemek ve bu bilgileri toplamak için kullanılır.

```
winlogbeat.event_logs:
```

```
- name: Application
```

```
- name: HardwareEvents
```

```
- name: System
```

```
- name: Security
```

Şekil 86. Winlogbeat.yml dosyası.

Ardından winlogbeat.yml içerisinde bulunan Kibana'yı yapılandırmak ve host'unu belirlemek için yorum satırı içerisinde bulunan host:"localhost:5601" etiketini yorum satırından çıkartıyoruz.

```
# ===== Kibana =====
# Starting with Beats version 6.0.0, the dashboards are loaded via the Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:

# Kibana Host
# Scheme and port can be left out and will be set to the default (http and 5601)
# In case you specify and additional path, the scheme is required: http://localhost
# IPv6 addresses should always be defined as: https://[2001:db8::1]:5601
#host: "localhost:5601"

# Kibana Space ID
# ID of the Kibana Space into which the dashboards should be loaded. By default,
# the Default Space will be used.
#space.id:

# ===== Kibana =====
# Starting with Beats version 6.0.0, the dashboards
# This requires a Kibana endpoint configuration.
setup.kibana:

# Kibana Host
# Scheme and port can be left out and will be set
# In case you specify and additional path, the sc
# IPv6 addresses should always be defined as: htt
host: "localhost:5601"

# Kibana Space ID
# ID of the Kibana Space into which the dashboard
# the Default Space will be used.
#space.id:
```

Şekil 87. Localhost:5601 etiketinin kaldırılması.

Daha sonra yeni bir Powershell açarak (yönetici olarak), yapılandırma dosyamızın olduğu dizine gidiyoruz. Ardından yapılandırma dosyamızın doğruluğunu test etmek için .\winlogbeat.exe test config -c .\winlogbeat.yml -e komutumuzu çalıştırıyoruz. Eğer herhangi bir hata yoksa şekildeki gibi çıktı alırız.

```
{"log.level":"info","@timestamp":"2024-05-05T14:41:16.342+0300","log.logger":"metric_registry","log.origin":
"github.com/elastic/beats/v7/libbeat/monitoring/inputmon.NewInputRegistry","file.name":"inputmon/input.go",
3}, {"message":"registering","service.name":"winlogbeat","input_type":"winlog","id":"ForwardedEvents","key":"
ts","uid":"49db2144-314f-4a87-beda-c10f095d517d","ecs.version":"1.6.0"}
{"log.level":"info","@timestamp":"2024-05-05T14:41:16.342+0300","log.logger":"metric_registry","log.origin":
"github.com/elastic/beats/v7/libbeat/monitoring/inputmon.NewInputRegistry","file.name":"inputmon/input.go",
3}, {"message":"registering","service.name":"winlogbeat","input_type":"winlog","id":"HardwareEvents","key":"H
","uid":"3ceb3ceb-4d6a-40f7-8af1-eae741344f20","ecs.version":"1.6.0"}
Config OK
PS C:\Elastic\Slack\7.16.1\winlogbeat>
```

Şekil 88. Komut çıktısı

.\winlogbeat.exe setup --dashboards komutu ile “winlogbeat” uygulamasının yapılandırılmasını ve başlatılmasını sağlıyoruz. “--dashboards” komutu ile gösterge panolarının yapılandırılmasını ve oluşturulmasını sağlıyoruz.

```
PS C:\Elastic Slack\7.16.1\winlogbeat> .\winlogbeat.exe setup --dashboards
Loading dashboards (Kibana must be running and reachable)
Loaded dashboards
PS C:\Elastic Slack\7.16.1\winlogbeat> █
```

Şekil 89. Gösterge panolarının yapılandırılması

Bir sonraki adım olarak kayıt verilerini toplamak ve ElasticSearch’e göndermek için .\install-service-winlogbeat.ps1 komutunu çalıştırıyoruz.

```
PS C:\Elastic Slack\7.16.1\winlogbeat> .\install-service-winlogbeat.ps1
[SC] DeleteService SUCCESS

Status      Name             DisplayName
-----
Stopped     winlogbeat       winlogbeat

PS C:\Elastic Slack\7.16.1\winlogbeat> █
```

Şekil 90. .\install-service-winlogbeat.ps1 komut çıktısı.

Son olarak Winlogbeat servisini başlatmamız için Start-Service winlogbeat komutunu çalıştırmalıyız.

```
PS C:\Elastic Slack\7.16.1\winlogbeat> Start-Service winlogbeat
PS C:\Elastic Slack\7.16.1\winlogbeat> █
```

Şekil 91. Start-Service winlogbeat komutunun çalıştırılması.

Kaynakça

- [1] Zaman Serisi Verileri - Azure Architecture Center. Docs.microsoft.com: (<https://docs.microsoft.com/tr-tr/azure/architecture/data-guide/scenarios/time-series>) (Erişim: 19 Şubat 2024)
- [2] Monitoring Nedir?. MediaClickCMS: (<https://www.mediatick.com.tr/tr/blog/monitoring-nedir>) (Erişim: 23 Şubat 2024)
- [3] Metrik - Analytics Yardım. Support.google.com: (<https://support.google.com/analytics/answer/6086087?hl=tr>) (Erişim: 25 Şubat 2024).
- [4] Grafana Nedir?. Ceaksan:(<https://ceaksan.com/tr/grafana-nedir/>) (Erişim: 1 Mart 2024)
- [5] Grafana Cloud - Send Data - Metrics - Metrics from Prometheus - Prometheus Config Examples - NoAgent Linux Node. Grafana Labs: (https://grafana.com/docs/grafana-cloud/send-data/metrics/metrics-prometheus/prometheus-config-examples/noagent_linuxnode/) (Erişim: 7 Mart 2024)
- [6] Çabuk, M. Prometheus ve Grafana Öğreniyoruz. Github: (https://github.com/muratcabuk/prometheus-ve-grafana-ogreniyoruz/blob/master/prometheus/9.grafana_integration.m) (8 Mart 2024)
- [7] Prometheus - Monitoring system & Time Series Database. Prometheus: (<https://prometheus.io/>) (Erişim: 13 Mart 2024)
- [8] Prometheus - Monitoring system & time series database. Grafana Labs: (<https://prometheus.io/>) (Erişim: 15 Mart 2024)

- [9] Bhatt, K. (2020). Prometheus and Grafana for Monitoring. Medium: (<https://medium.com/nerd-for-tech/prometheus-and-grafana-for-monitoring-4bbcd9e50b27>) (Erişim: 20 Mart 2024)
- [10] Prometheus Documentation. Prometheus: (<https://prometheus.io/docs/>) (22 Mart 2024)
- [11] Grafana - Open-Source Analytics & Monitoring Platform. Grafana Labs: (<https://grafana.com/grafana/>) (Erişim: 1 Nisan 2024)
- [12] (2019). The Prometheus Alertmanager. Robust Perception: (<https://prometheus.io/docs/alerting/alertmanager/>) (Erişim: 3 Nisan 2024)
- [13] Prometheus and Grafana Monitoring. The DevOps School: (<https://www.devopsschool.com/blog/prometheus-and-grafana-monitoring/>) (Erişim: 10 Nisan 2024)
- [14] (2020). Instrumenting Applications. Robust Perception: (<https://prometheus.io/docs/practices/instrumentation>) (Erişim: 13 Nisan 2024)
- [15] Grafana Cloud - Send Data - Metrics - Metrics from Prometheus - Prometheus Config Examples - NoAgent Linux Node. Grafana Labs: (https://grafana.com/docs/grafana-cloud/send-data/metrics/metrics-prometheus/prometheus-config-examples/noagent_linuxnode/) (Erişim: 20 Nisan 2024)
- [16] How To Set Up Prometheus and Grafana on a VPS for Monitoring. DigitalOcean: (<https://www.digitalocean.com/community/tutorials/how-to-set-up-prometheus-and-grafana-on-a-vps-for-monitoring>) (Erişim: 22 Nisan 2024)
- [17] Monitoring with Prometheus and Grafana. CloudAcademy: (<https://cloudacademy.com/blog/monitoring-with-prometheus-and-grafana/>) (Erişim: 26 Nisan 2024)

- [18] Prometheus vs. Grafana: Understanding the Differences. Jelvix: (<https://jelvix.com/blog/prometheus-vs-grafana>) (Erişim: 2 Mayıs 2024)
- [19] Kubernetes Monitoring with Prometheus and Grafana. Sysdig: (<https://sysdig.com/blog/kubernetes-monitoring-with-prometheus-grafana/>) (Erişim: 2 Mayıs 2024)
- [20] Different Types Of Logs In SIEM And Their Log Formats. ManageEngine: (<https://www.manageengine.com/log-management/siem/collecting-and-analysing-different-log-types.html>) (Erişim: 1 Mayıs 2024)
- [21] Kumar, S. (14 Şubat 2024). TYPES OF LOGS. Medium: (<https://sharath-kumar.medium.com/types-of-logs-5cc6cdb40482>) (Erişim: 20 Nisan 2024)
- [22] (2024, Nisan 13). Logging (computing). Wikipedia: ([https://en.wikipedia.org/wiki/Logging_\(computing\)](https://en.wikipedia.org/wiki/Logging_(computing))) (Erişim: 3 Mayıs 2024)
- [23] Definition: What Is a Log File?. Sematext: (<https://sematext.com/glossary/log-file/>) (Erişim: 1 Mayıs 2024)
- [24] Sharif, A. (2022, Aralık 21). What is a Log File?. CrowdStrike: (<https://www.crowdstrike.com/cybersecurity-101/observability/log-file/>) (Erişim: 29 Şubat 2024)
- [25] (2023, Aralık 18). Overview of System Logging. Juniper: (<https://www.juniper.net/documentation/us/en/software/junos/net-work-mgmt/topics/topic-map/system-logging.html>) (Erişim: 20 Nisan 2024)
- [26] Linux Logging Basics. Solarwinds: (<https://www.loggly.com/ultimate-guide/linux-logging-basics/>) (Erişim: 25 Nisan 2024)

- [27] Janani. (2021, Kasım 15). Log File. Atatus: (<https://www.atatus.com/glossary/log-file/>) (Erişim: 1 Mayıs 2024)
- [28] Log Nedir? İşletmeler İçin Log Kaydı Tutmak Neden Önemlidir?. Hosting.com.tr: (<https://www.hosting.com.tr/bilgi-bankasi/log-nedir/>) (Erişim: 1 Mayıs 2024)
- [29] (2021, Eylül 26). Log Nedir?. Adli Bilişim Hizmetleri: (<https://adlibilisimhizmetleri.com/log-nedir/>) (Erişim: 24 Nisan 2024)
- [30] Ersöz Enes, D. (2022, Aralık 19). Windows Sistem Log Dosyaları. Medium: (<https://medium.com/@davutenesersoz/windows-sistem-log-dosyalar%C4%B1-ve-olay-t%C3%BCrleri-5c9a725e96bb>) (Erişim: 24 Nisan 2024)
- [31] Linux Sistem Logları. Veriloji: (<https://blog.veriloji.com/linux-sistem-loglari/>) (Erişim: 3 Mayıs 2024)
- [32] Log Nedir? Log Kaydı ve Log Tutma. Wmaracı: (<https://wmaraci.com/nedir/log>) (Erişim: 29 Nisan 2024)