

Crontab

Unix temelli sistemlerde belirli zaman dilimlerinde çalıştırılması istenen komut, script veya uygulamaların otomatize halde çalışmasını sağlamak için Crontab servisi kullanılmaktadır. Crontab servisinde görev tanımı yapmanın iki farklı yolu bulunmaktadır. Bunlar;

- 1- “/etc/” dizini altında Cron dizinleri bulunmaktadır. Hazırlanan script dosyaları bu dizinlere konumlandırılarak saatlik, günlük, haftalık veya aylık olarak çalıştırılması sağlanabiliyor.

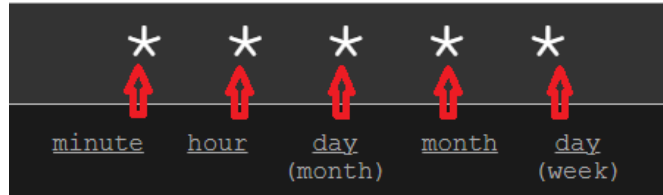
```
vm-user@Ubuntu-VM:~$ ls -alh /etc | grep -i cron*
-rw-r--r-- 1 root root 335 Mar 23 2022 anacrontab
drwxr-xr-x 2 root root 4,0K Kas 1 16:19 cron.d
drwxr-xr-x 2 root root 4,0K Oca 5 22:18 cron.daily
drwxr-xr-x 2 root root 4,0K Ağu 9 14:48 cron.hourly
drwxr-xr-x 2 root root 4,0K Ağu 9 14:50 cron.monthly
-rw-r--r-- 1 root root 1,2K Mar 23 2022 crontab
drwxr-xr-x 2 root root 4,0K Ağu 9 14:51 cron.weekly
vm-user@Ubuntu-VM:~$
```

- 2- “/etc/” dizini altında bulunan Crontab isimli dosya kullanılarak daha spesifik zaman dilimlerinde çalıştırılması istenen görev tanımları yapılabiliyor.

Crontab Dosyasında Kullanılan Zaman Formatı

Crontab dosyasında görev tanımı yapabilmek için öncelikle tanımlanacak görevin çalıştırılacağı zaman dilimlerinin belirlenmesi gerekiyor. Zaman tanımı için 5 farklı zaman dilimi kullanılıyor. Bunlar sırasıyla;

- Minute, görevin kaç dakikada bir çalışacağını veya hangi dakikada çalışacağını belirlemek için kullanılıyor (0 - 59).
- Hour, görevin kaç saatte bir çalışacağını veya hangi saatlerde çalışacağını belirlemek için kullanılıyor (0 - 23).
- Month (day), görevin aynı hangi günlerinde çalışacağını belirtmek için kullanılıyor (1 - 31).
- Month, görevin hangi aylarda çalışacağını belirtmek için kullanılıyor (1 - 12).
- Day (Week), görevin haftanın hangi günlerinde çalışacağını belirtmek için kullanılıyor (1 - 7, Mon-Tue-Wed...).



Bir görevin çalışacağı zaman dilimi tanımlarken kullanılan sembollerin anlamları;

- “*”, belirli bir zaman dilimi belirtilmediği zamanlarda “her zaman” anlamında kullanılıyor. Örnek olarak “* * * * *” tanımının anlamı her dakikada çalıştırılacağını gösterir.
- “,”, belirli zaman tanımlarını sıralamak için kullanılıyor. Örnek olarak “15 12 1,3,5 * *” tanımının anlamı her ayın 1,3 ve 5. günü, saat 12:15’de çalıştırılacağını gösterir.
- “-”, bir zaman aralığı belirtilirken kullanılıyor. Örnek olarak “00 1 1-10 * *” tanımının anlamı her ayın ilk 10 günü, saat 1:00’da çalıştırılacağını gösterir.
- “/”, tekrar sıklığı belirtilirken kullanılıyor. Örnek olarak “*/3 * * * *” tanımının anlamı her 3 dk’da bir çalıştırılacağını gösterir.

```
GNU nano 6.2 /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab`
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
# You can also override PATH, but by default, newer versions inherit it from the environment
#PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
17 * * * * root cd / && run-parts --report /etc/cron.hourly
25 6 * * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

| → Crontab dosyasının içeriğine bakıldığında; görselden de anlaşılacağı gibi aslında Cron dizinleri (cron.hourly, cron.daily, cron.weekly, cron.monthly) Crontab dosyasında saatlik, günlük, haftalık veya aylık olarak çalıştırılması için tanımlı birkaç satırdan ibarettir. Yani bu dizinlere benzer şekilde yılda bir çalıştırılması istenen görevler için örnek olarak /etc dizini altında “cron.yearly” adında yeni bir dizini oluşturularak bu dizin için Crontab dosyasında “00 00 1 1 *” gibi bir zaman tanımı yapılabilir.

Zaman tanımlarına örnek vermek gerekirse (örnekler <https://crontab.guru/> sitesi üzerinde gerçekleştirilmiştir. Zaman tanımlarının nasıl yapıldığı ve tanımların hangi periyotlarda çalışacağını burada inceleyebilirsiniz);

- Görevin, her saatin 15. dakikasında çalışması için “15 * * * *” tanımı kullanılır.

```
“At minute 15.”
next at 2023-01-15 19:15:00
then at 2023-01-15 20:15:00
then at 2023-01-15 21:15:00
then at 2023-01-15 22:15:00
then at 2023-01-15 23:15:00
15 * * * *
```

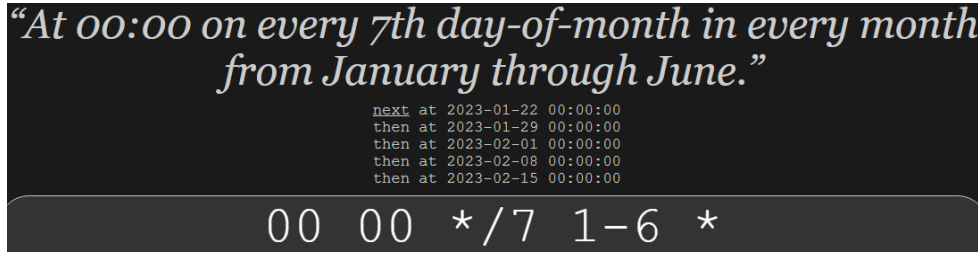
- Görevin, her iki günde bir saat 17:00’de çalışması için “00 17 * */2 *” tanımı kullanılır.

```
“At 17:00 on every 2nd day-of-month.”
next at 2023-01-17 17:00:00
then at 2023-01-19 17:00:00
then at 2023-01-21 17:00:00
then at 2023-01-23 17:00:00
then at 2023-01-25 17:00:00
00 17 * */2 * *
```

- Görevin, her ayın 7, 14 ve 21. günlerinde saat 17:00’de çalışması için “00 17 * 7,14,21 *” tanımı kullanılır.

```
“At 17:00 on day-of-month 7, 14, and 21.”
next at 2023-01-21 17:00:00
then at 2023-02-07 17:00:00
then at 2023-02-14 17:00:00
then at 2023-02-21 17:00:00
then at 2023-03-07 17:00:00
00 17 7,14,21 * *
```

- Görevin, yılın ilk yarısında her 7 günde bir saat 00:00'da çalışması için "00 00 */7 1-6 *" tanımı kullanılır.

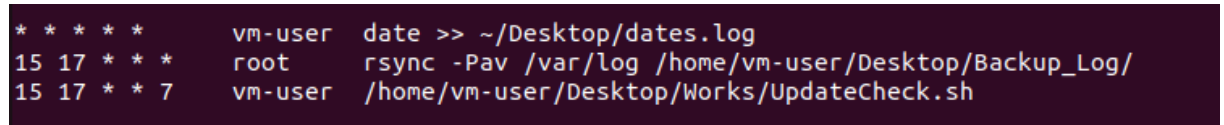


Crontab Dosyasında Zamanlanmış Görev Oluşturma

Crontab dosyasında görev tanımı oluşturmak için öncelikle Root yetkisine sahip olunmalı. Root yetkisiyle Crontab dosyasını açıldıktan sonra "<Time Infos> <User for Exec> <Command | Script File>" formatında görev tanımları yapılıyor. Dikkat edilmesi gereken noktalar;

- Komutları doğrudan uygulamadan önce CLI üzerinde denenerek doğru çalıştığından emin olmalıdır (Eksik bir komut, tanımlanan görevin çalışmamasına neden olacaktır).
- Komutu çalıştıracak kullanıcı seçilirken, kullanıcının işlemleri gerçekleştirebilecek yetkiye sahip olup olmadığına dikkat edilmelidir. Yeterli yetkiye sahip olmayan kullanıcılar kullanılırsa tanım çalışmayacaktır.
- Dosya ve dizinler için Path tanımı yapılırken dikkatli olunmalıdır (özellikle Tilde (~) kullanılıyorsa).

Örnek Tanımlar



- İlk satırda, her dakikada tarih ve saat bilgilerinin "dates.log" isimli dosyaya yazılması sağlanıyor.
- İkinci satırda, hergün saat 17:15'te /var/log dizininin Desktop'a yedeklenmesi sağlanıyor.
 - /var/log dizini altında işlem yapabilmek için Root yetkilerine sahip olmak gerekiyor. Bu nedenle çalıştırılacak kullanıcı Root olarak seçildi.
- Üçüncü satırda güncellemelerin Pazar günleri saat 17:15'te kontrol edilmesi ve masaüstüne kaydedilmesini sağlayan scriptin çalıştırılması sağlanmıştır.

NOT

- Tanımlarda zaman dilimlerini sembollerle ifade etmek yerine @yearly @daily @hourly @reboot gibi anahtar kelimeler de kullanılabilir.