

NAGIOS İLE SİSTEM ve NETWORK İZLEME

Çağrı Ersen

cagri.ersen@gmail.com

<http://www.syslogs.org>

<https://twitter.com/CagriErsen>

ÖZGÜR YAZILIM VE LINUX GÜNLERİ

5-6 NİSAN 2013

İSTANBUL BİLGİ ÜNİVERSİTESİ / SANTRAL KAMPÜSÜ

- **Sistem ve Ağ izleme Konsepti**
- **Nagios Nedir ?**
- **Çalışma Prensipleri**
- **Nagios Pluginleri**
- **Yapılandırma Mantığı ve Örnekler**

AĞ ve SİSTEM İZLEME KONSEPTİ ÜZERİNE TEMEL BİLGİLER

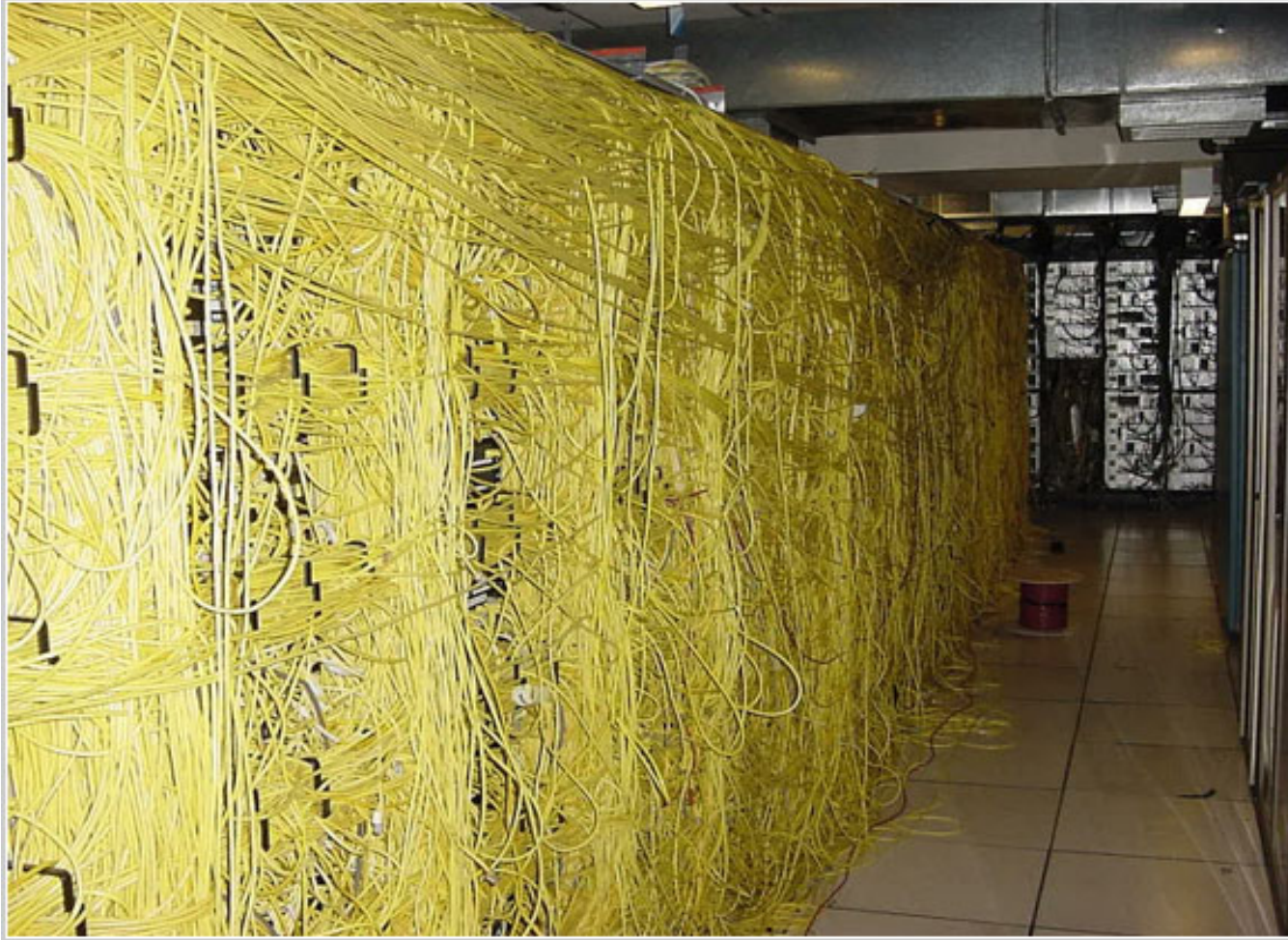
Bilgisayar ağıları, üzerlerindeki bileşenler ve servislerden meydana gelen organizmalardır.

Duyguları olmasa da dertleri, tasaları vardır.

Bu nedenle en iyi ağ, kendini ifade etme yeteneği olan, problemini dile getirebilen ağdır.



Zira gerçekten karmaşık olabilirler;



ve örneğin bir kablo yerinden çıkıverebilir...

SERVİS KALİTESİ

Bu bakış açısı ile “*monitoring*” kavramı bir ağ'ın kendini ifade etme yetisine tekabül etmektedir.

Ağınızın, üzerindeki problemleri olabilecek en kısa sürede raporlaması, sorunları servis verilen son kullanıcıdan önce farkederek gidermeye olanak sağlar.

Bu nedenle, “*Servis Kalitesine*” doğrudan etkisi vardır.

ANALİZ ve PLANLAMA

Ağın sürekli izlenmesi ile trend ve performans gibi kıymetli bilgiler içeren bir veritabanı oluşturulmuş olur.

Analize olanak sağlayan bu veri, yönetmekle sorumlu olduğunuz yapıyı daha iyi tanımanıza olanak sağlar.

Bu şekilde operasyonel süreçler daha iyi yönetilebilir ve planlamaya yönelik öngörüler daha tutarlı olur.

NAGIOS NEDİR ?

Nagios, IT altyapılarıyla ilgili problemleri, iş süreçleri etkilenmeden önce tespit ederek çözmeye olanak sağlayan açık kaynak kodlu bir monitoring sistemidir.

- ✓ İlk sürümü 1999 yılında yayınlanmıştır.
- ✓ Linux altında çalışmak üzere geliştirilmiş olsa da Unix türevlerinde de sorunsuz çalışabilir.
- ✓ Community tarafından binlerce proje (Plugins, Frontends, Conf. Tools vs.) geliştirilmiştir.
- ✓ GPL Lisansı ile dağıtılmaktadır.

Temel Özellikler

- ✓ Tek bir araç ile işletim sistemlerinin, uygulama ve servislerin, network ve IT altyapı bileşenlerinin izlenebilmesine olanak sağlar.
- ✓ HTTP/s, SMTP, POP3, IMAP, SSH, FTP vs. gibi network servislerini,
- ✓ CPU, Memory Disk kullanımı gibi sistem kaynaklarını doğrudan izleyebilir.

Temel Özellikler

- ✓ Basit plugin yapısı sayesinde, kendi plugin'lerinizi yazabilir, özel uygulama ve servislerinizi izleyebilirsiniz.
- ✓ Parallel check özelliği sayesinde aynı anda birden fazla iş yapar.
- ✓ Parent host özelliği ile network hiyerarşisi tanımlanabilir.

Temel Özellikler

- ✓ Teknik ekibe, email ya da sms üzerinden alarmlar gönderir.
- ✓ Teknik ekip problemi zamanında çözmezse, alarmı eskale edip ekip yöneticisine iletir :(
- ✓ Parent host özelliği gereksiz alarm önüne geçer. Parent host down ise arkasındaki diğer host ve servisler için bildirim göndermez.

Temel Özellikler

- ✓ Contactlar gruplandırılabilir; Bu şekilde alarmlar sadece ilgililerine gönderilir.
- ✓ Contactlar için belirli zaman aralıkları tanımlanabilir ve bildirimler bu zaman dilimlerine göre iletilirler.
- ✓ Uygulama ve Servislerde oluşan problemlere otomatik olarak müdahale edilebilir.

Temel Özellikler

- ✓ Basit ve temiz web arayüzü sayesinde tüm IT altyapısının merkezi tek bir mecra üzerinden görüntülenebilmesini sağlar.
- ✓ İzlenen her bir “şey” için detaylı durum bilgilerini web arabirimi üzerinden alabilme imkanı verir.
- ✓ Geçmişe yönelik detaylı availability raporları alınmasına olanak sağlar.

ÇALIŞMA PRENSİBİ

Kontrol (Check) İşlemleri

Nagios sunucusu üzerinde tüm izleme işlemlerinden sorumlu olan “*Nagios daemon*”'ı çalışır.

Bu daemon, kendisine belirtilen periyotlarda belirtilen host ve servislerde kontrol işlemleri yapacak pluginlerin çalıştırılmasını sağlar.

Pluginlerin çalışmasının ardından alınan durum bilgileri Nagios'a gönderilir.

Çalışma Prensipleri

Check işlemleri “host check” ve “service check” olmak üzere iki ayrı dala ayrılır.

Host Check işlemi “ağıdaki cihazların” çalışır olup olmadıklarını kontrole yönelik yapılır.

Service Check işlemleri ise, network servisleri, sistem kaynakları vs. gibi host'lar üzerinde çalışan bileşenlerin durumlarını saptamak üzere yapılır.

Host Check

En basit haliyle host check işlemi, uzaktaki sisteme ping paketleri göndermek sureti ile yapılır.

check-host-alive ismi verilen bu işlemde, check_ping plugini üzerinden gönderilen ping paketlerine cevap alınıp alınmadığı ve ping süreleri ile paket kayıp bilgileri Nagios'a bildirir.

Nagios da bu bilgilerden hareketle, daha önceden tanımlanan eşik değerlerine göre hostun durumuna karar verir.

Service Check

Service check işlemlerinde http(s),ftp, ssh vs gibi network servisleri ile cpu, memory disk gibi kaynak kullanım bilgileri kontrol edilir.

İlgili servislerin network üzerinden doğrudan erişilir olması durumunda, Nagios'un kendi üzerinde bulunan check_tcp, check_http, gibi pluginler kullanılır.

Örnek olarak herhangi bir network servisine ait portun erişilir olup olmadığı, porta açılan bağlantılara alınan cevaplar üzerinden ilgili servisin durumu gibi bilgiler bahsi geçen pluginler üzerinden Nagios'a iletilirler.

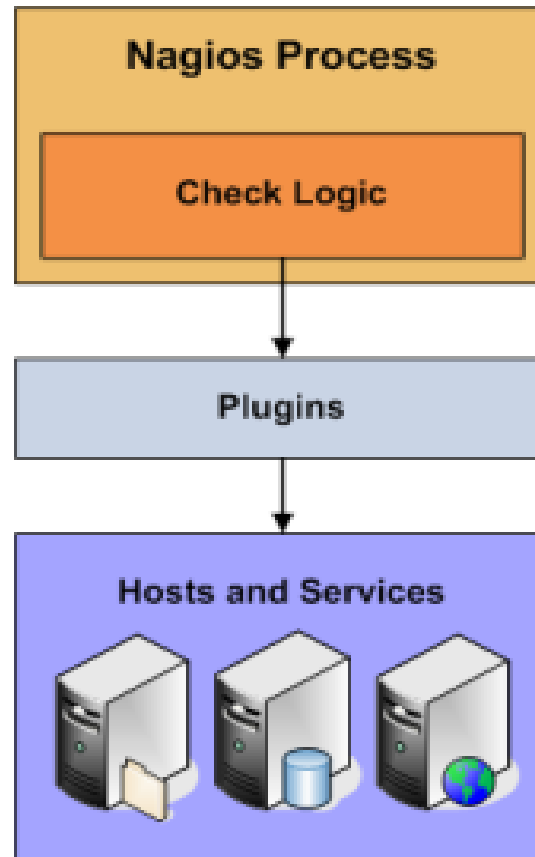
Çalışma Prensipleri

Eğer disk, memory, cpu kullanımı gibi uzak sunucunun doğrudan erişilemeyecek tipteki bileşenleri monitor edilecekse; ssh üzerinden ve ya NRPE (Nagios Remote Plugin Executor) isimli uygulama kullanılarak kontroller gerçekleştirilir.

Doğrudan erişilemeyen servislere yönelik kontrol işlemlerinde, nagios daemon uzaktaki sunucuya, check etmek istediği kaynak ile ilgili nagios plugin'ini çalıştırmasını ve sonucu kendisine bildirmesini söyler.

Bu tip işlemlerin adı “active check”'tir.

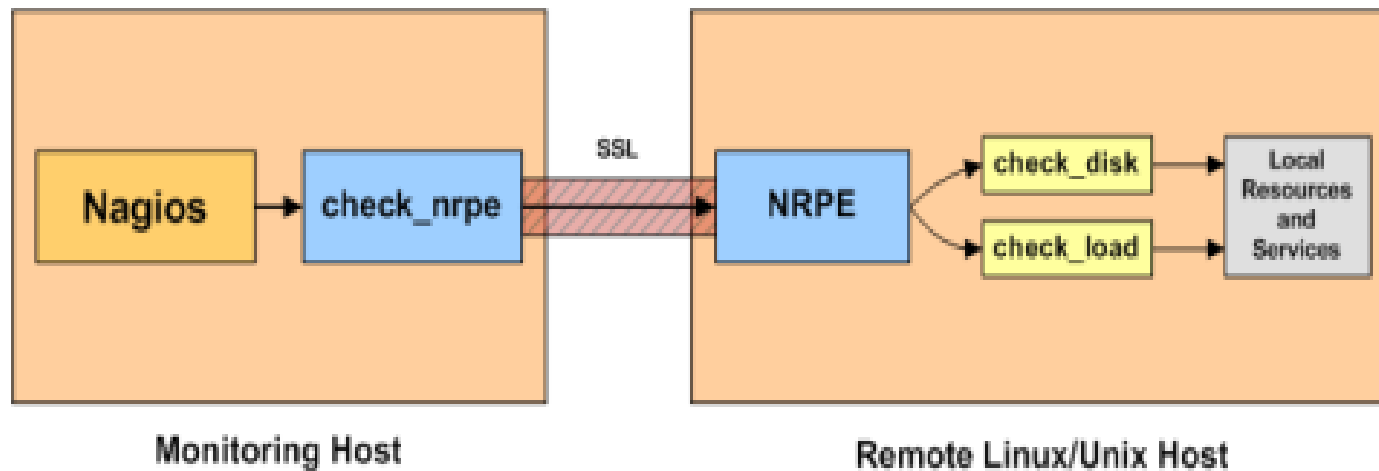
Active Check



Active Check işleminde plugin'ler Nagios tarafından tetiklenirler.

Çalışma Prensipleri

NRPE üzerinden yapılan active check işlemi



Nagios, check_nrpe plugini ile uzaktaki hostun 5666 no'lu tcp portunu dinleyen NRPE daemon'ına bağlanır ve kontrol edilecek kaynağa ait plugin'i çalıştırmasını söyler.

Passive Check

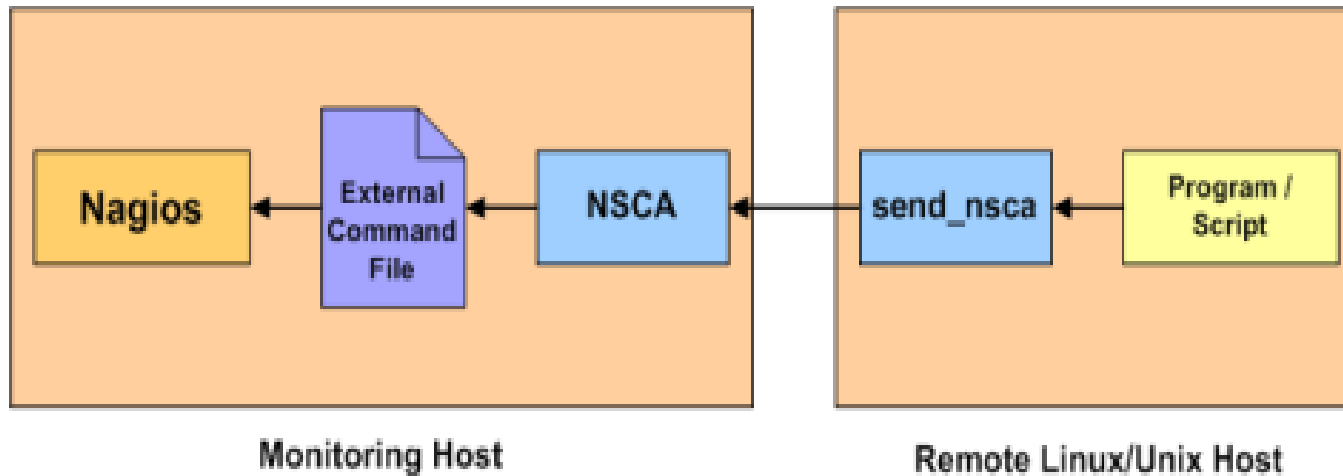
Bir diğer kontrol şekli de uzaktaki sunucuya doğrudan erişimin olmadığı durumlarda kullanılan passive check isimli yöntemdir.

Arada firewall vs. gibi trafiği engelleyen bir nokta olması ya da host üzerindeki NRPE'ye erişememeniz halinde kullanılabilir.

Bu yöntemde, Nagios uzak sisteme bağlanmak yerine; uzak sistem check plugin'lerini kendisi çalıştırır ve Nagios üzerinde çalışan NSCA (Nagios Service Check Acceptor) daemon'ına bildirir.

Çalışma Prensipleri

NSCA üzerinden yapılan passive check işlemi:



Bu check yönteminde, uzak sistem kendi üzerinde status kontrolünü yapar ve send_nsca uygulamasını kullanarak Nagios üzerindeki 5667 no'lu tcp portunu dinleyen NSCA daemon'ına check durumunu iletir.

Durumlar ve Durum Tipleri

Nagios'un bir host ya da servisin durumu ile ilgili net karara varması için gerekli iki öge bulunur:

- Host ya da Servis'in durumu
 - UP, DOWN, UNREACHABLE.
 - OK, WARNING, CRITICAL, UNKNOWN
- Host ya da Servis durumunun tipi.
 - SOFT, HARD

Soft State:

Bir host ya da servis için dönen cevap UP ya da OK değilse, durumdan kesin emin olunabilmesi için belirlediğimiz periyod'da ve adette yeniden kontrol (re-check) yapılır.

Örnek olarak 15 dakikada bir yapılan bir cpu load kontrolünde status OK olarak dönmediyse, check işleminin her üç dakikada bir üç kez daha yapılmasını belirtebiliriz.

Bu tekrar kontrollerin birinde “durum” OK olarak bildirilecek olursa herhangi bir alarm gönderilmez. Bu durum “SOFT Recovery” olarak adlandırılmaktadır.

Hard State:

Bir host ya da servis için dönen cevap UP ve ya OK değilse ve tekrar kontrol işlemleri tamamlanmışsa bu durum HARD olarak değerlendirilir ve konuyla ilgili alarm üretilerek muhattaplarına gönderilir.

Bu şekilde Nagios, sadece “gerçek” problemlerle ilgili bildirim göndererek “alarm bombardımanı” yapmamış olur.

Bir host ya da servis'in durumu HARD iken re-check aşamasında OK ya da UP cevabı alınırsa bu durum “Hard Recover” olarak adlandırılır.

NAGIOS PLUGIN'LERİ

Nagios Pluginleri

Nagios, tüm check işlemlerini “plugin” ismi verilen “external program”lara yaptırır.

Plugin'ler, komut satırından çalışabilen, host ve servislerin durumlarını check etmek üzere geliştirilmiş küçük binary'ler ya da scriptlerdir. (shell, perl vs.)

Temel olarak, komut satırından çalıştırılabilen bu programcıklar, host ya da servislerin durumlarını kontrol edip Nagios'a bildirirler.

Nagios Pluginleri

Nagios'un resmi bir plugin paketi bulunmaktadır; ve Nagios sunucu ile NRPE üzerinden monitor edilecek tüm host'ların üzerinde kurulu olması gerekir.

Ayrıca, topluluk tarafından geliştirilen binlerce plugin'in yayınlandığı Nagios Exchange isimli bir web sitesi bulunmaktadır.

Herhangi bir kaynak/servisi kontrol etmek üzere komut satırından çalıştırılabilecek hemen “her şey” için bir plugin yazılabilmesi mümkündür.

Nagios Pluginleri

Nagios Ekibi Tarafından Geliştirilen Pluginler

apt, breeze, by_ssh, clamd, cluster, dhcp, dig, disk, disk_smb, dns, dummy, file_age, flexlm, fping, ftp, game, hpjd, http, icmp, ide_smart, ifoperstatus, ifstatus, imap, ircd, jabber, ldap, load, log, mailq, mrtg, mrtgtraf, mssql, mysql, mysql_query, nagios, netdns, nntp, nntps, nt, ntp_peer, ntp_time, nwstat, oracle, overcr, pgsq, ping, pop, procs, radius, real, rpc, sensors, simap, smtp, snmp, spop, ssh, ssmtp, swap, tcp, time, udp, ups, users, wave

Nagios Pluginleri

Topluluk Tarafından Geliştirilen Pluginler

apache, apc_ups, arping, asterisk, backup, bgpstate ,
cluster, cluster2, compaq_insight, dns_random, email_loop,
fan_cpq_present, fan_fsc_present, hltherm, hprsc,
http-with-client-certificate, hw, inodes, inodes-freebsd,
ipxping, javapro, joy, linux RAID, lmmon, log2, lotus,
maxchannels, maxwanstate, mem, ms_spooler, mssql,
nagios, nagios_db, nagios_db_pg, netapp, nmap,
oracle_instance, oracle_tbs, ora_table_space, pcpmetric,
pfstate, qmailq, rbl, remote_nagios_status, rrd_data, sap,
smart, smb, snmp_disk_monitor, snmp_printer,
snmp_process_monitor, snmp_procs, sockets, temp_cpq,
temp_fsc, timeout, traceroute, traceroute-pure_perl,
uptime, vcs, wave, wins

TANIMLAMALAR

Nagios Dizinleri

/usr/local/nagios: Nagios ana dizini.

/usr/local/nagios/bin: Binary dosyaları.

/usr/local/nagios/etc: Temel yapılandırma dosyaları.

/usr/local/nagios/etc/objects: Nesne tanım dosyaları.

/usr/local/nagios/libexec: Nagios Pluginleri.

/usr/local/nagios/sbin: CGI dosyaları.

/usr/local/nagios/share: Web GUI dosyaları.

/usr/local/nagios/var: Cache, lock ve log dosyaları.

Nagios Yapılandırma Dosyaları

nagios.cfg: Ana nagios yapılandırma dosyası.

cgi.cfg: CGI yapılandırması.

resource.cfg: Kaynak yapılandırma dosyası.

commands.cfg: Komutların tanımlandığı dosya.

contacts.cfg: Kontakt ve kontakt gruplarının tanımlandığı dosya.

templates.cfg: Host ve servis tanımları için şablon dosyası.

nagios.cfg

Tam yolu, /usr/local/nagios/etc/nagios.cfg'dir.

İçerisinde Nagios'un çalışma yapısı ile ilgili ana yapılandırma direktifleri tanımlanır. Tüm diğer yapılandırma dosyalarına bu dosya içerisinde atıfta bulunulur.

#dikkate alınacak yapılandırma dosyaları.

cfg_file=/usr/local/nagios/etc/objects/commands.cfg

cfg_file=/usr/local/nagios/etc/objects/contacts.cfg

cfg_file=/usr/local/nagios/etc/objects/timeperiods.cfg

cfg_file=/usr/local/nagios/etc/objects/templates.cfg

İzlenecek hostlara ait yapılandırma dosyaları.

cfg_file=/usr/local/nagios/etc/objects/localhost.cfg

resource.cfg

/usr/local/nagios/etc/resource.cfg pathinde bulunur.

Kullanıcı macroları (\$USERx) burada tanımlanır.

Örneğin olarak öntanımlı olarak şu macro bulunur:

\$USER1\$=/usr/local/nagios/libexec

Bu macro default nagios plugin'lerine ait yolu göstermektedir.

Ayrıca, CGI'lar tarafından erişilmediği için username/password gibi özel bilgiler de bu dosyada tanımlanır.

commands.cfg

/usr/local/nagios/etc/objects/commands.cfg yolunda bulunur.

Check ve alarm gönderme vs gibi işler için çalıştırılacak komutlar bu dosya içerisinde belirlenir.

Örneğin bir servisi kontrol etmek üzere tanımlama yaptığınızda, Nagios ilgili kontrol için çalıştırılacak komutu bu dosya içerisinden öğrenir.

commands.cfg host check komutu

'check-host-alive' command definition

define command{

*command_name **check-host-alive***

*command_line **\$USER1\$/check_ping -H***

\$HOSTADDRESS\$ -w 3000.0,80% -c 5000.0,100% -p 5

}

Bu örnekte, host durum kontrolleri için check-host-alive komutu tanımlanmaktadır. Nagios, command olarak “check-host-alive”'ı gördüğü her yerde ilgili host için bu komutu yürütür. (\$USER1\$ resource.cfg'de tanımlanmış plugin dizinine atıfta bulunmaktadır.)

contacts.cfg

Tam yolu /usr/local/nagios/etc/objects/contacts.cfg'dir.

Bildirimlerin gönderileceği kişilere ve gruplara ait contact bilgileri ve burada tanımlanır.

define contact{

<i>contact_name</i>	<i>nagiosadmin</i>
<i>alias</i>	<i>Nagios Admin</i>
<i>service_notification_period</i>	<i>24x7</i>
<i>host_notification_period</i>	<i>24x7</i>
<i>service_notification_options</i>	<i>w,u,c,r,f,s</i>
<i>host_notification_options</i>	<i>d,u,r,f,s</i>
<i>service_notification_commands</i>	<i>notify-service-by-email</i>
<i>host_notification_commands</i>	<i>notify-host-by-email</i>
<i>email</i>	<i>nagiosadmin@email.com</i>

contacts.cfg

Contacts.cfg içerisinde grup tanımlamaları da yapılabilir:

```
define contactgroup{
```

<i>contactgroup_name</i>	<i>SMS</i>
<i>alias</i>	<i>SMS ile bildirim alanlar.</i>
<i>members</i>	<i>nagiosadmin,Kisi1,Kisi2</i>
<i>}</i>	

Örneğin SMS ya da e-mail için ayrı iki contact grubu oluşturup kimi durumlarda sms kimi durumlarda sadece e-mail gönderilmesini sağlayabilirsiniz.

templates.cfg

/usr/local/nagios/etc/objects/templates.cfg yolunda bulunur.

Bu dosyada host ve servisler için önceden şablonlar oluşturuyoruz. Böylece tanımlama işlemlerinde tekrar tekrar aynı satırları yazmak yerine “use” parametresi ile ilgili template'e atıfta bulunarak o şablona ait özelliklerin kullanılmasını bildiriyoruz.

templates.cfg

```
define host{
```

<i>name</i>	<i>generic-host</i>	
<i>notifications_enabled</i>	<i>1</i>	→ <i>Bildirimler devrede.</i>
<i>event_handler_enabled</i>	<i>1</i>	→ <i>Otomatik aksiyon devrede.</i>
<i>flap_detection_enabled</i>	<i>1</i>	→ <i>Flap tespiti.</i>
<i>notification_period</i>	<i>24x7</i>	→ <i>Bildirim periyodu.</i>
<i>Register</i>	<i>0</i>	→ <i>Bu tanımı register etme.</i>

```
}
```

Bu tanımlamadan sonra artık her host tanımlamada “*use generic-host*” ifadesi kullanılabilir.

templates.cfg

```
define host{  
    name                linux-server  
    use                 generic-host;generic-host şablonunu kullan  
    check_period        24x7                ;24x7 çalış.  
    check_interval      5                   ;5 dakikada bir kontrol yap.  
    retry_interval      1                   ;nonUP için re-check aralığı.  
    max_check_attempts  10                  ;nonUP için max re-check.  
    check_command       check-host-alive   ;Çalıştırılacak komut.  
    notification_period workhours          ;Bildirim saatleri.  
    notification_interval 120              ;Her 2 saatte bir yeniden bildir.  
    notification_options d,u,r            ;Bu stateler için bildirim yap.  
    contact_groups      admins             ;Admin grubuna bildir.  
    Register            0                  ;Bu tanım, template'tir register etme.  
}
```

templates.cfg

```
define service{
  name                generic-service
  active_checks_enabled 1 ;active check yapılabilir.
  passive_checks_enabled 1 ;passive check yapılabilir.
  Parallelize_check 1 ;Paralel check yapılabilir.
  notifications_enabled 1 ;Bildirimler devrede.
  event_handler_enabled 1 ;Otomatik aksiyonlar devrede.
  flap_detection_enabled 1 ;“flap” durumunu tespit et.
  check_period 24x7 ;27x7 check et.
  normal_check_interval 10 ;Her 10 dakikada bir check.
  max_check_attempts 3 ;nonOK durumlarda max 3 kez check.
  retry_check_interval 2 ;nonOK durumlar için 2 dk'de bir check
  contact_groups admins ;Alarmları admin grubuna gönder.
  notification_options w,u,c,r ;Bu stateler için bildirim yap
  notification_interval 60 ;Her 1 saatte bir yeniden bildir.
  notification_period 24x7 ;Bildirimleri 24x7 periyodunda yap.
  Register 0 ;Bu bir template; register etme.
```

Host Tanımlamaları

```
define host{  
  use          linux-server ;linux-server isimli şablonunu dahil et.  
  host_name    Sunucu1      ;Bu host'un ismi.  
  alias        Web Sunucu   ;Host'un alternatif adı.  
  Address      10.0.0.10    ;Host'un adresi.  
}
```

Bu tanımlamaya göre, ilgili Sunucu1 ismi verilen 10.0.0.10 ip adresli host, “linux-server” şablonunda belirtildiği şekilde check-host-alive (check_ping) komutu kullanılarak check edilecektir.

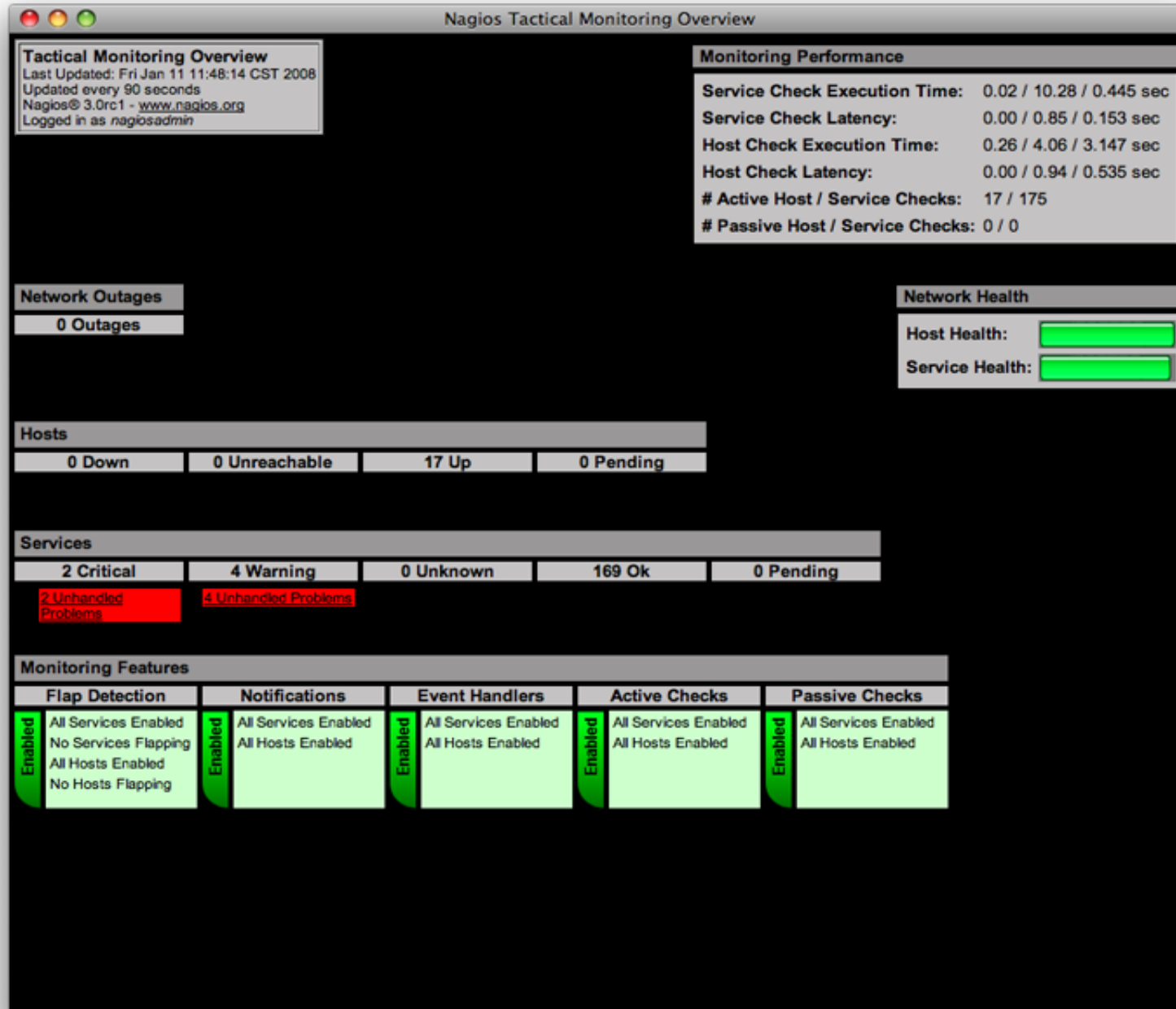
Servis Tanımlamaları

```
define service{
use          generic-service          ;Kullanılacak şablon
host_name    Sunucu1                 ;Servisin çalıştığı host.
service_description CPU Load         ;Açıklaması.
check_command check_nrpe!check_load ;Komut.
}
```

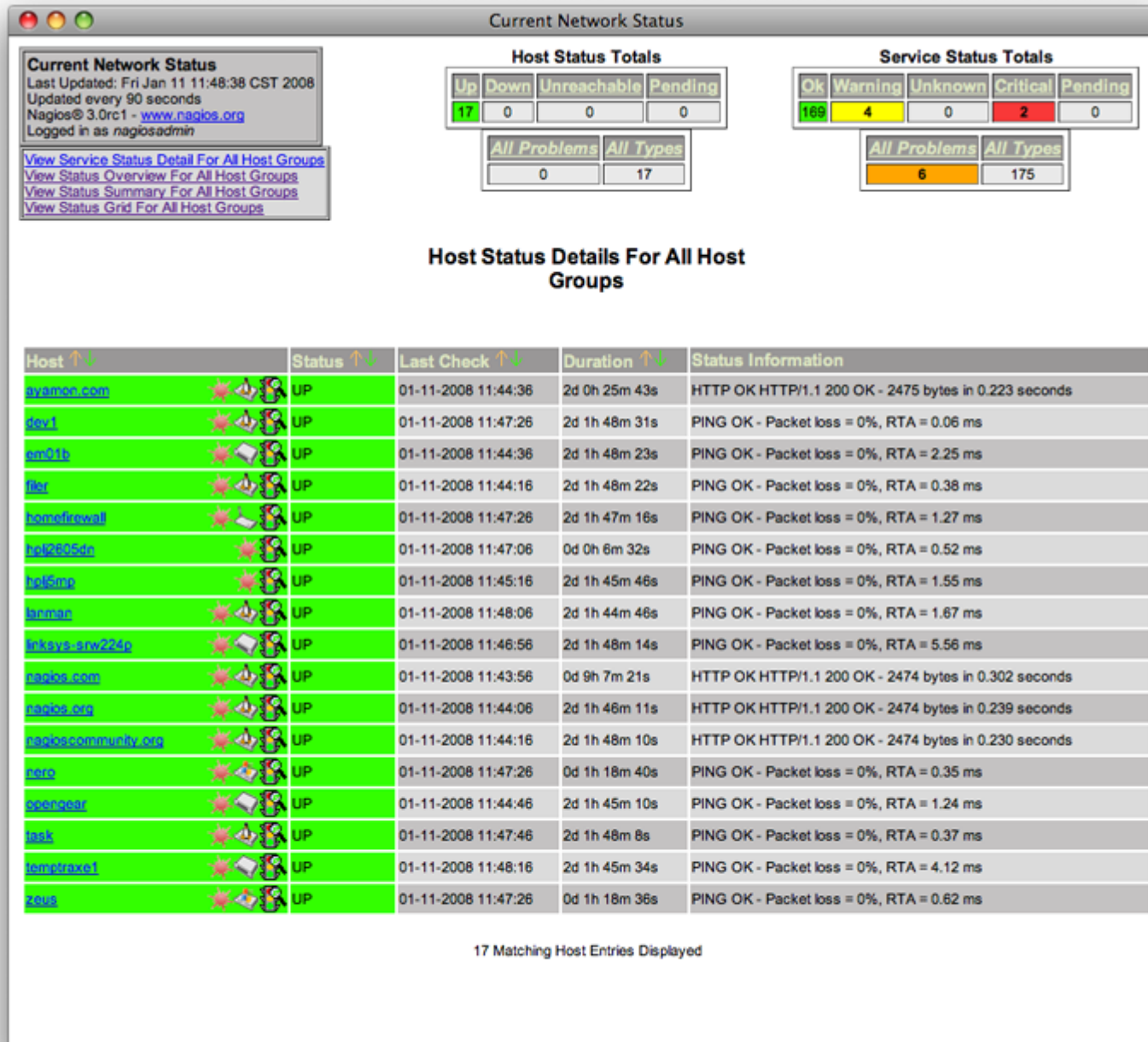
```
define service{
use          generic-service
host_name    Sunucu1
service_description HTTP Response Check
check_command check_http!-u /falanca.php -t 5 -s "Giriş Yap:"
}
```

```
define service{
use          generic-service
host_name    Sunucu1
service_description SMTP Response Check
check_command check_smtp!-t 10 -e "220 mail.falan.com"
```

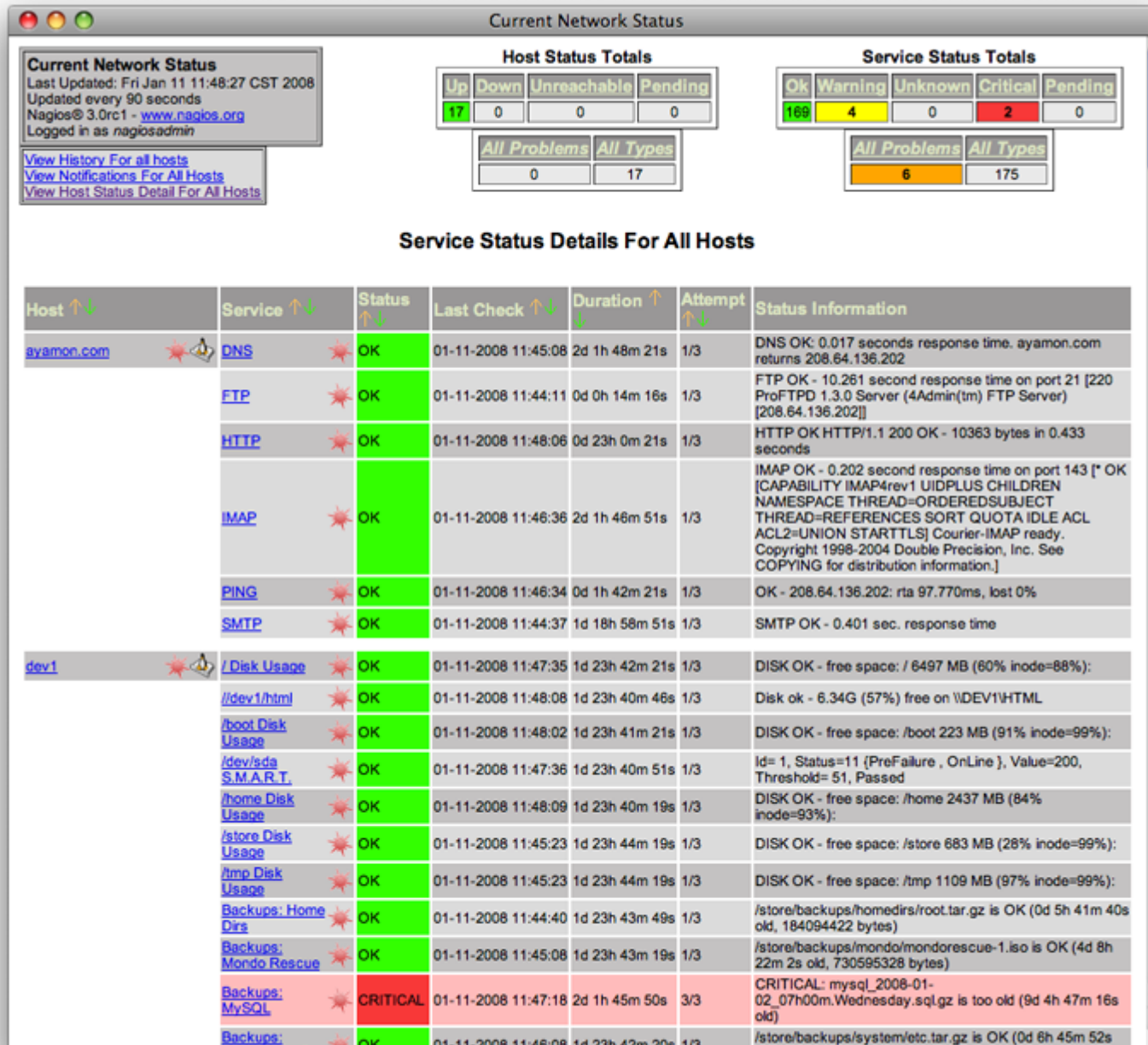
Genel Görünüm Ekranı



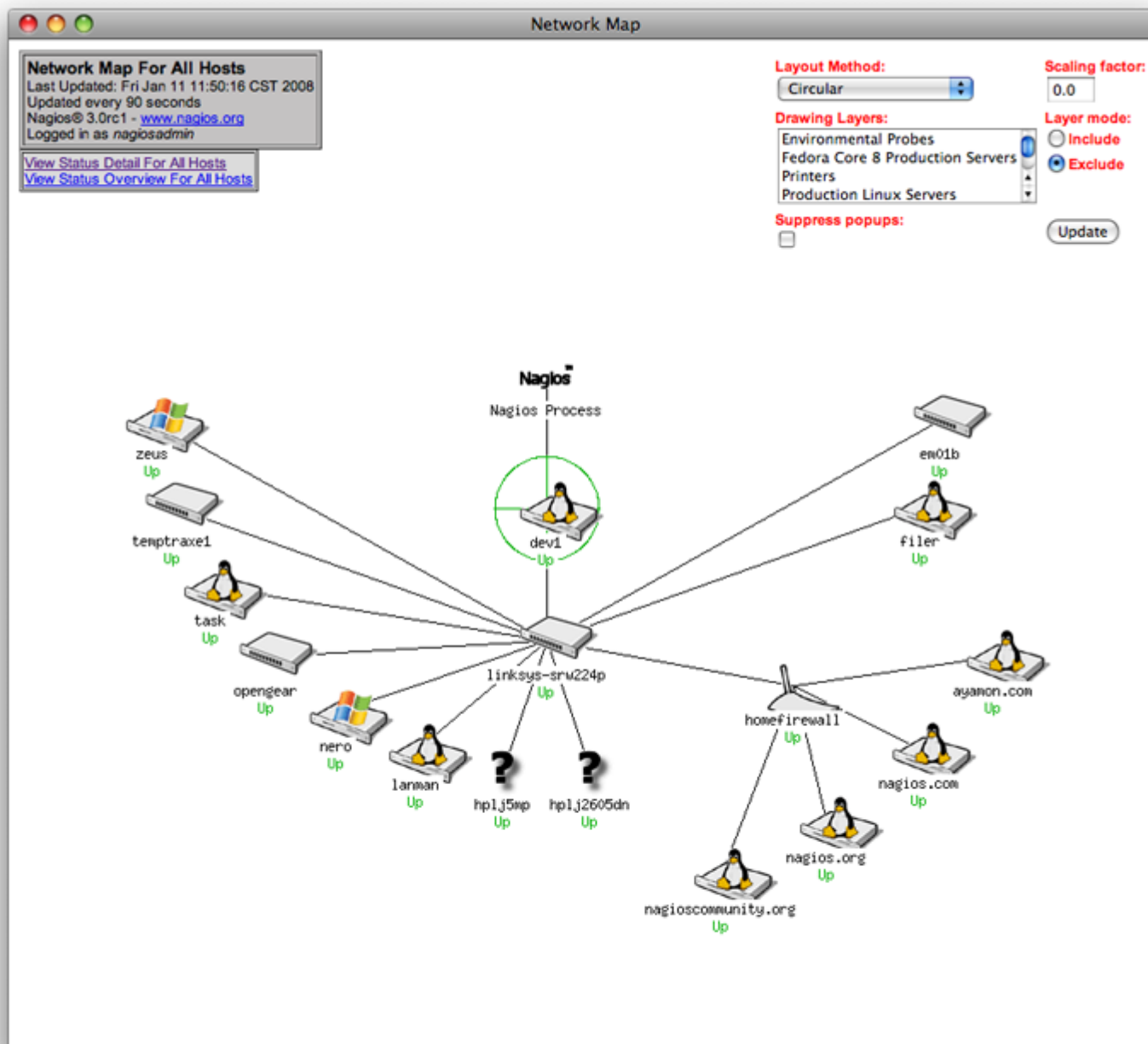
Host Detay Ekranı



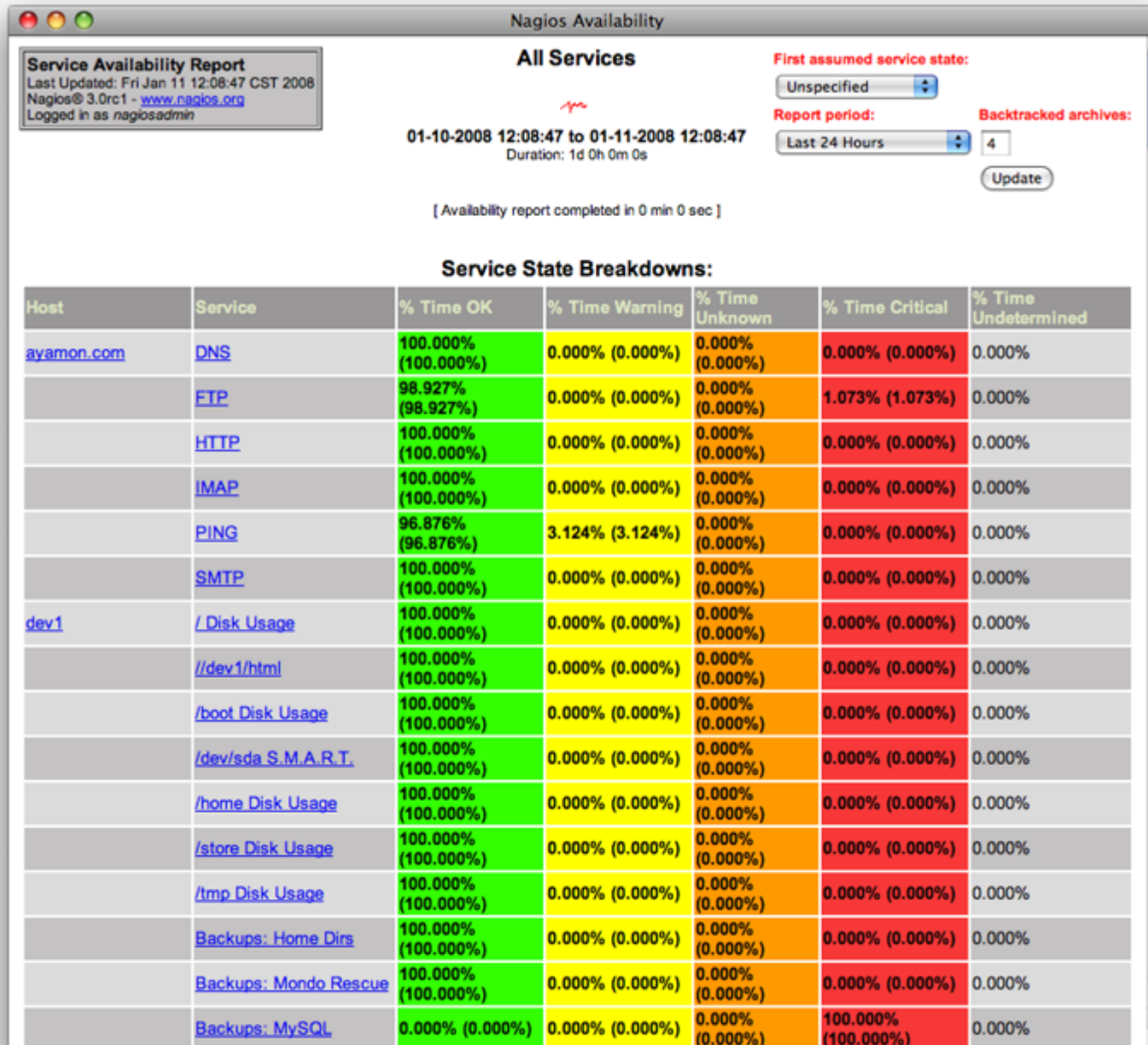
Servis Detay Ekranı



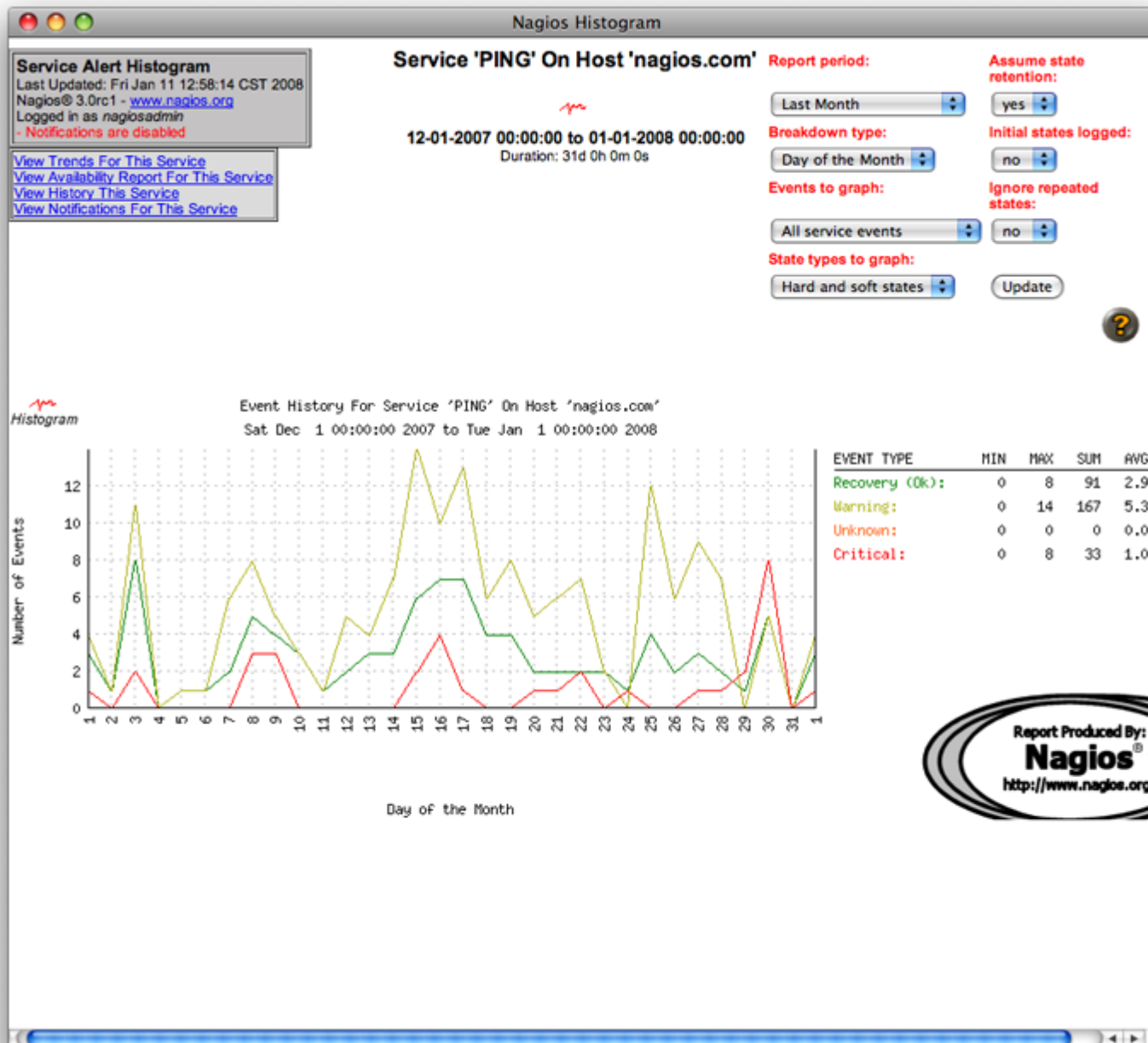
Durum Haritası



Servis Kalite Raporları



Alarm Histogramları



Detaylı kurulum ve yapılandırma dökümanı için:

<http://syslogs.org/nagios-kurulumu-ve-yapilandirmasi>

Nagios Resmi Dökümanları için:

<http://nagios.sourceforge.net/docs/nagioscore/3/en/>

TEŞEKKÜRLER