Puppet ile Dağıtım

Puppet, nerede çalıştırılırsa çalıştırılsın tüm yazılımlarınızı otomatik olarak denetlemenizi, teslim etmenizi, çalıştırmanızı ve geleceğe hazırlamanızı sağlayan açık kaynaklı bir yazılım aracıdır. Birçok Unix benzeri sistemde ve Microsoft Windows'ta çalışır ve sistem yapılandırmasını tanımlamak için kendi beyan dilini içerir. Kullanımı çok basittir ve Wazuh'u kolayca kurmanıza ve yapılandırmanıza olanak tanır.

- Puppet Master'ı Kurma
- Puppet Agent Yükleme
- Puppet Sertifikalarının Kurulumu
- Wazuh Modülünü Kurun
- Puppet Aracılığıyla Bir Yığın Kurun
- Wazuh Kullanıcıları İçin Şifreyi Değiştirme
- Puppet Aracılığıyla Wazuh Agent Yükleme
- Wazuh Puppet Referansı

Puppet Master'ı Kurma

Bu bölüm puppet-master'ın nasıl kurulacağını açıklar . Resmi kurulum kılavuzunu kontrol etmek için bu bağlantıyı takip edin .

DNS yapılandırmanız yoksa, ad çözümlemesi için hosts dosyanızı kullanmalısınız. Dosyayı düzenleyin `/etc/hosts`ve aşağıdakileri ekleyin:

```
[puppet master ip] puppet puppet-master
[puppet agent ip] puppet-agent
```

CentOS/RHEL/Fedora'ya Kurulum

Puppet yum deposunu ve ardından “puppetserver” paketini yükleyin. Linux dağıtımınız için puppet deposunu yüklemek için gereken doğru rpm dosyasını bulmak için bu dizine bakın. Örneğin, CentOS 8 veya RHEL 8 için Puppet 7'yi yüklemek için aşağıdakileri yapın:

```
sudo rpm -Uvh https://yum.puppet.com/puppet7-release-el-8.noarch.rpm
yum -y install puppetserver
```

Yüklü ikili dosya ile varsayılan ikili dosyanız arasında sembolik bir bağlantı oluşturun:

```
ln -s /opt/puppetlabs/bin/puppet /bin
ln -s /opt/puppetlabs/server/bin/puppetserver /bin
```

Debian/Ubuntu'ya Kurulum

Manifest, wazuh'u kurmak için aşağıdaki sürümleri destekler.

- **Debian** : 7 (hırıltılı), 8 (jessie), 9 (stretch), 10 (buster), 11 (boğa gözü), 12 (kitap kurdu)
- **Ubuntu** : 12.04 (Hassas Pangolin), 14.04 (Güvenilir Tahr), 15.04 (Canlı Vervet), 15.10 (Kurnaz Kurtadam), 16.04 (Xenial Xerus), 16.10 (Yakkety Yak), 18.04 (Biyonik Kunduz), 20.04 (Focal Fossa), 22.04 (Reçel Denizanası)

Kurun `curl`ve :`apt-transport-https``lsb-release`

```
apt-get update
apt-get install curl apt-transport-https lsb-release wget
```

Uygun Puppet apt deposunu ve ardından “puppetserver” paketini yükleyin. Linux dağıtımınız için Puppet 7 deposunu yüklemek üzere doğru deb dosyasını bulmak için https://apt.puppetlabs.com adresine bakın.

```
wget https://apt.puppet.com/puppet7-release-focal.deb
dpkg -i puppet7-release-focal.deb
apt-get update
apt-get install -y puppetserver
```

Yüklü ikili dosya ile varsayılan ikili dosyanız arasında sembolik bir bağlantı oluşturun:

```
ln -s /opt/puppetlabs/bin/puppet /bin
ln -s /opt/puppetlabs/server/bin/puppetserver /bin
```

Bellek Tahsisi

Varsayılan olarak, Puppet Server 2 GB RAM kullanacak şekilde yapılandırılacaktır. Ancak, bir VM'de Puppet Server ile denemeler yapmak istiyorsanız, güvenli bir şekilde 512 MB kadar az bellek tahsis edebilirsiniz. Puppet Server bellek tahsisini değiştirmek için, aşağıdaki init yapılandırma dosyasını düzenleyebilirsiniz.

- `/etc/sysconfig/puppetserver`-- CentOS/RHEL/Fedora
- `/etc/default/puppetserver`-- Debian/Ubuntu

Değişkendeki 2g'yi `JAVA_ARGS`Puppet Server'a tahsis etmek istediğiniz bellek miktarıyla değiştirin. Örneğin, 1 GB bellek tahsis etmek için ; kullanın . 512 MB için .`JAVA_ARGS="-Xms1g -Xmx1g"` `JAVA_ARGS="-Xms512m -Xmx512m"`

Yapılandırma

`/etc/puppetlabs/puppet/puppet.conf`Puppet sunucusunu yapılandırmak için dosyayı düzenleyin . `[main]` Bölüme aşağıdaki ayarları ekleyin. Bölüm yoksa, bölümü oluşturmanız gerekir. Kendi DNS'inizi kurduysanız `puppet`ve yerine `puppet-master`Tam Nitelikli Alan Adlarınızı (FQDN'ler) koyun.

```
[main]
server = puppet-master
dns_alt_names = puppet, puppet-master
```

> Not: Yapılandırma dosyasında bulursanız `templatedir=$confdir/templates`, o satırı silin. Kullanım dışı bırakıldı.

Ubuntu/Debian makineleri için, puppetserver başlamazsa. puppetserver dosyasını düzenleyin, `/etc/default/puppetserver`. Bellek boyutunu 1G veya 512MB olarak değiştirmek için aşağıdaki satırı değiştirin:

```
JAVA_ARGS="-Xms512m -Xmx512m -Djruby.logger.class=com.puppetlabs.jruby_utils.jruby.Slf4jLogger"
```

Daha sonra Puppet Server'ınızı başlatın:

Systemd

```
systemctl start puppetserver
systemctl enable puppetserver
systemctl status puppetserver
```

SysV Başlatma

```
service puppetserver start
update-rc.d puppetserver
```

Puppet Agent Yükleme

Bu bölümde puppet-agent'ın nasıl kurulacağı anlatılmaktadır . Resmi kurulum kılavuzunu kontrol etmek için bu bağlantıyı takip edin .

Puppet Server'ınızda yaptığınız gibi, Puppet deposunu da aracı sisteminize `apt` yüklediğinizi varsayıyoruz .`yum`

DNS yapılandırmanız yoksa, ad çözümlemesi için hosts dosyanızı kullanmalısınız.

Dosyayı düzenleyin `/etc/hosts`ve Puppet ana makinesinin ve aracısının IP adresini ve ana bilgisayar adını ekleyin:Bu başlığa kalıcı bağlantı

```
[puppet master ip] puppet puppet-master
[puppet agent ip] puppet-agent
```

CentOS/RHEL/Fedora'ya Kurulum

Puppet yum deposunu ve ardından "puppet-agent" paketini yükleyin. Linux dağıtımınız için puppet deposunu yüklemek için gereken doğru rpm dosyasını bulmak için bu dizine bakın. Örneğin, CentOS 7 veya RHEL 7 için Puppet 7'yi yüklemek için aşağıdakileri yapın:

```
sudo rpm -Uvh https://yum.puppet.com/puppet7-release-el-8.noarch.rpm
yum -y install puppet-agent
```

Yüklü ikili dosya ile varsayılan ikili dosyanız arasında sembolik bir bağlantı oluşturun:

```
ln -s /opt/puppetlabs/bin/puppet /bin
```

Debian/Ubuntu'ya Kurulum

Manifest, wazuh'u kurmak için aşağıdaki sürümleri destekler.

- **Debian** : 7 (hırıltılı), 8 (jessie), 9 (stretch), 10 (buster), 11 (boğa gözü), 12 (kitap kurdu)
- **Ubuntu** : 12.04 (Hassas Pangolin), 14.04 (Güvenilir Tahr), 15.04 (Canlı Vervet), 15.10 (Kurnaz Kurtadam), 16.04 (Xenial Xerus), 16.10 (Yakkety Yak), 18.04 (Biyonik Kunduz),

20.04 (Focal Fossa), 22.04 (Reçel Denizanası)

Kurun `curl`ve :`apt-transport-https``lsb-release`

```
apt-get update
apt-get install curl apt-transport-https lsb-release wget
```

Uygun Puppet apt deposunu ve ardından “puppet-agent” paketini yükleyin. Linux dağıtımınız için puppet deposunu yüklemek üzere doğru deb dosyasını bulmak için https://apt.puppetlabs.com adresine bakın.

```
wget https://apt.puppet.com/puppet7-release-focal.deb
dpkg -i puppet7-release-focal.deb
apt-get update
apt-get install -y puppet-agent
```

Yüklü ikili dosya ile varsayılan ikili dosyanız arasında sembolik bir bağlantı oluşturun:

```
ln -s /opt/puppetlabs/bin/puppet /bin
```

Windows'a Kurulum

1. Windows puppet-agent paketini indirin.
   Bu paket Puppet'ın tüm ön koşullarını bir araya getiriyor.

   > Not: Bu, Puppet 7.16 sürüm aracısı için pakettir. Başka bir pakete ihtiyaç duyulursa, tüm paketlerin indirilebildiği resmi dizine gidin.

2. Puppet'ı kurun.
   1. Windows GUI'yi kullanma:

   - GUI'yi yükseltilmiş ayrıcalıklarla çalıştırın.
   - Kurulum sırasında Puppet sizden Puppet ana sunucunuzun ana bilgisayar adını ister.
   - Ana bilgisayara bağlanmayacak bağımsız Puppet düğümleri için varsayılan ana bilgisayar adını (*puppet) kullanın. Ayrıca komut satırına yüklemek ve aracı*

başlatma modunu Devre Dışı olarak ayarlamak isteyebilirsiniz .
- Kurulum tamamlandıktan sonra Puppet kurulmuş ve çalışır hale gelecektir.

2. Komut satırını kullanarak:

```
msiexec /qn /norestart /i puppet-agent-<VERSION>-x64.msi
```

İsteğe bağlı olarak, kurulumun ilerlemesini bir dosyaya kaydetmeyi belirtebilirsiniz . Ayrıca, Puppet'ı kurarken önceden yapılandırmak için çeşitli MSI özelliklerini ayarlayabilirsiniz.`/l*v install.txt`

Aracı Yapılandırması

Puppet aracısını yapılandırmak için düğümdeki yapılandırma dosyasını düzenleyin.

- `/etc/puppetlabs/puppet/puppet.conf`Linux sistemleri için
- `C:\ProgramData\PuppetLabs\puppet\etc\puppet.conf`Windows sistemleri için

`server`Ayarı dosyanın bölümüne ekleyin `[main]`. Kendi DNS'inizi ayarladıysanız, `puppet-master`Puppet sunucunuzun Tam Nitelikli Alan Adı (FQDN) ile değiştirin.

```
[main]
server = puppet-master
```

> Not: Puppet sunucusunun FQDN'si Puppet aracı ana bilgisayarı tarafından çözülmelidir.

Puppet hizmetini yeniden başlatın ve durumunu kontrol edin:

```
puppet resource service puppet ensure=running enable=true
sudo systemctl status puppet
```

Puppet Sertifikalarının Kurulumu

Bir sertifika oluşturmak ve imzalamak için aşağıdaki adımları izleyin:

1. Puppet aracısında, boş bir sertifika oluşturmak için şu komutu çalıştırın:

   ```
   puppet agent -t
   ```

2. Puppet sunucu tarafında, onaylanması gereken mevcut sertifikaları listeleyin:

   ```
   puppetserver ca list
   ```

   Düğüm ana bilgisayar adınızı içeren bir liste çıktısı vermelidir.

3. Sertifikayı onaylayın ve bunu `pending-agent-node` aracınızın düğüm adıyla değiştirin:

   ```
   puppetserver ca sign --certname pending-agent-node
   ```

   Tüm sertifikalar bununla onaylanabilir:

   ```
   puppetserver ca sign --all
   ```

4. Puppet aracı düğümüne geri dönün ve puppet aracısını tekrar çalıştırın:

   ```
   puppet agent -t
   ```

> Not: Başarılı bir sertifika imzalama için özel ağ DNS'inin ön koşul olduğunu unutmayın.

Wazuh Modülünü Kurun

Bu modül Nicolas Zin tarafından yazılmış ve Jonathan Gazeley ve Michael Porter tarafından güncellenmiştir. Wazuh, onu sürdürme amacıyla çatallamıştır. Katkılarından dolayı yazarlara teşekkür ederiz.

Wazuh Modülünü Kurun

Puppet Forge'dan Wazuh modülünü indirin ve kurun:

```
puppet module install wazuh-wazuh --version 4.9.2
```

Output

```
Notice: Preparing to install into /etc/puppetlabs/code/environments/production/modules ...
Notice: Downloading from https://forgeapi.puppet.com ...
Notice: Installing -- do not interrupt ...
/etc/puppetlabs/code/environments/production/modules
└─┬ wazuh-wazuh (v4.9.2)
  ├── puppet-nodejs (v7.0.1)
  ├── puppet-selinux (v3.4.1)
  ├── puppetlabs-apt (v7.7.1)
  ├─┬ puppetlabs-concat (v6.4.0)
  │ └── puppetlabs-translate (v2.2.0)
  ├── puppetlabs-firewall (v2.8.1)
  ├─┬ puppetlabs-powershell (v4.1.0)
  │ └── puppetlabs-pwshlib (v0.10.1)
  └── puppetlabs-stdlib (v6.6.0)
```

Bu modül Wazuh aracısını ve yöneticisini kurar ve yapılandırır.

Puppet Aracılığıyla Bir Yığın Kurun

Tek Düğüm

Tek düğümlü bir yığını dağıtmak için aşağıda gösterilen bildirimi kullanabilirsiniz. Bu yığın şunlardan oluşur:

- Wazuh gösterge paneli
- Wazuh dizinleyici
- Wazuh yöneticisi
- Dosyabeat

Aşağıdaki içerikle `stack.pp`dosyayı oluşturun :`/etc/puppetlabs/code/environments/production/manifests/`

- `puppet-aio-node`: Puppet aracısının ana bilgisayar adı veya IP adresi.
- `puppet-server`: Wazuh modülü kurulduğunda Puppet sunucusunun ana bilgisayar adı veya IP adresi.

```
$discovery_type = 'single-node'
stage { 'certificates': }
stage { 'repo': }
stage { 'indexerdeploy': }
stage { 'securityadmin': }
stage { 'dashboard': }
stage { 'manager': }
Stage[certificates] -> Stage[repo] -> Stage[indexerdeploy] -> Stage[securityadmin] -> Stage[manager] -> Stage[
Exec {
timeout => 0,
}
node "puppet-server" {
class { 'wazuh::certificates':
  indexer_certs => [['node-1','127.0.0.1']],
  manager_certs => [['master','127.0.0.1']],
  dashboard_certs => ['127.0.0.1'],
  stage => certificates,
}
}
node "puppet-aio-node" {
class { 'wazuh::repo':
stage => repo,
}
class { 'wazuh::indexer':
  stage => indexerdeploy,
}
class { 'wazuh::securityadmin':
stage => securityadmin
}
class { 'wazuh::manager':
  stage => manager,
}
class { 'wazuh::filebeat_oss':
  stage => manager,
}
class { 'wazuh::dashboard':
  stage => dashboard,
```

```
}
}
```

Çoklu Düğüm

Aşağıdaki çoklu düğüm bildirimini kullanarak, aşağıdaki düğümlerden oluşan dağıtılmış bir yığını üç farklı sunucuya veya Sanal Makineye (VM) dağıtabilirsiniz.

- 3 dizinleyici düğümü
- Yönetici ana düğümü
- Yönetici işçi düğümü
- Pano düğümü

Her uygulamayı yüklediğiniz sunucuların IP adreslerini mutlaka eklemelisiniz.

```
$node1host   = '<WAZUH_INDEXER_NODE1_IP_ADDRESS>'
$node2host   = '<WAZUH_INDEXER_NODE2_IP_ADDRESS>'
$node3host   = '<WAZUH_INDEXER_NODE3_IP_ADDRESS>'
$masterhost    = '<WAZUH_MANAGER_MASTER_IP_ADDRESS>'
$workerhost    = '<WAZUH_MANAGER_WORKER_IP_ADDRESS>'
$dashboardhost = '<WAZUH_DASHBOARD_IP_ADDRESS>'
$indexer_node1_name = 'node1'
$indexer_node2_name = 'node2'
$indexer_node3_name = 'node3'
$master_name = 'master'
$worker_name = 'worker'
$cluster_size = '3'
$indexer_discovery_hosts = [$node1host, $node2host, $node3host]
$indexer_cluster_initial_master_nodes = [$node1host, $node2host, $node3host]
$indexer_cluster_CN = [$indexer_node1_name, $indexer_node2_name, $indexer_node3_name]
# Define stage for order execution
stage { 'certificates': }
stage { 'repo': }
stage { 'indexerdeploy': }
stage { 'securityadmin': }
stage { 'dashboard': }
stage { 'manager': }
Stage[certificates] -> Stage[repo] -> Stage[indexerdeploy] -> Stage[securityadmin] -> Stage[manager] -> Stage[
Exec {
timeout => 0,
}
node "puppet-server" {
class { 'wazuh::certificates':
  indexer_certs => [["$indexer_node1_name","$node1host" ],["$indexer_node2_name","$node2host" ],["$indexer_
  manager_master_certs => [["$master_name","$masterhost"]],
  manager_worker_certs => [["$worker_name","$workerhost"]],
  dashboard_certs => ["$dashboardhost"],
  stage => certificates
}
```

```
class { 'wazuh::repo':
stage => repo
}
}
node "puppet-wazuh-indexer-node1" {
class { 'wazuh::repo':
stage => repo
}
class { 'wazuh::indexer':
  indexer_node_name => "$indexer_node1_name",
  indexer_network_host => "$node1host",
  indexer_node_max_local_storage_nodes => "$cluster_size",
  indexer_discovery_hosts => $indexer_discovery_hosts,
  indexer_cluster_initial_master_nodes => $indexer_cluster_initial_master_nodes,
  indexer_cluster_CN => $indexer_cluster_CN,
  stage => indexerdeploy
}
class { 'wazuh::securityadmin':
indexer_network_host => "$node1host",
stage => securityadmin
}
}
node "puppet-wazuh-indexer-node2" {
class { 'wazuh::repo':
stage => repo
}
class { 'wazuh::indexer':
  indexer_node_name => "$indexer_node2_name",
  indexer_network_host => "$node2host",
  indexer_node_max_local_storage_nodes => "$cluster_size",
  indexer_discovery_hosts => $indexer_discovery_hosts,
  indexer_cluster_initial_master_nodes => $indexer_cluster_initial_master_nodes,
  indexer_cluster_CN => $indexer_cluster_CN,
  stage => indexerdeploy
}
}
node "puppet-wazuh-indexer-node3" {
class { 'wazuh::repo':
stage => repo
}
class { 'wazuh::indexer':
  indexer_node_name => "$indexer_node3_name",
  indexer_network_host => "$node3host",
  indexer_node_max_local_storage_nodes => "$cluster_size",
  indexer_discovery_hosts => $indexer_discovery_hosts,
  indexer_cluster_initial_master_nodes => $indexer_cluster_initial_master_nodes,
  indexer_cluster_CN => $indexer_cluster_CN,
  stage => indexerdeploy
}
}
node "puppet-wazuh-manager-master" {
class { 'wazuh::repo':
```

```
stage => repo
}
class { 'wazuh::manager':
  ossec_cluster_name => 'wazuh-cluster',
  ossec_cluster_node_name => 'wazuh-master',
  ossec_cluster_node_type => 'master',
  ossec_cluster_key => '012345678901234567890123456789012',
  ossec_cluster_bind_addr => "$masterhost",
  ossec_cluster_nodes => ["$masterhost"],
  ossec_cluster_disabled => 'no',
  stage => manager
}
class { 'wazuh::filebeat_oss':
  filebeat_oss_indexer_ip => "$node1host",
  wazuh_node_name => "$master_name",
  stage => manager
}
}
node "puppet-wazuh-manager-worker" {
class { 'wazuh::repo':
stage => repo
}
class { 'wazuh::manager':
  ossec_cluster_name => 'wazuh-cluster',
  ossec_cluster_node_name => 'wazuh-worker',
  ossec_cluster_node_type => 'worker',
  ossec_cluster_key => '012345678901234567890123456789012',
  ossec_cluster_bind_addr => "$masterhost",
  ossec_cluster_nodes => ["$masterhost"],
  ossec_cluster_disabled => 'no',
  stage => manager
}
class { 'wazuh::filebeat_oss':
  filebeat_oss_indexer_ip => "$node1host",
  wazuh_node_name => "$worker_name",
  stage => manager
}
}
node "puppet-wazuh-dashboard" {
class { 'wazuh::repo':
stage => repo,
}
class { 'wazuh::dashboard':
  indexer_server_ip  => "$node1host",
  manager_api_host   => "$masterhost",
  stage => dashboard
}
}
```

Manifest'te açıklanan kukla düğümlerinin IP adresleriyle olan ilişkisi şu şekildedir:

- `puppet-wazuh-indexer-node1`= `node1host`. Wazuh indeksleyici node1.

- `puppet-wazuh-indexer-node2`= `node2host`. Wazuh indeksleyici node2.
- `puppet-wazuh-indexer-node3`= `node3host`. Wazuh indeksleyici node3.
- `puppet-wazuh-manager-master`= `masterhost`. Wazuh yönetici ustası.
- `puppet-wazuh-manager-worker`= `workerhost`. Wazuh yönetici işçi.
- `puppet-wazuh-dashboard`= `dashboardhost`. Wazuh panosu düğümü.

Sınıfın , Wazuh modülünün kurulu olduğu `wazuh::certificates`Puppet sunucusunda (`puppet-server`) uygulanması gerekir. Bu, arşiv modülünün Wazuh yığın dağıtımındaki tüm sunuculara dosyaları dağıtmak için kullanılması nedeniyle gereklidir.

Daha fazla Wazuh dizinleyici düğümüne ihtiyacınız varsa, yeni değişkenler ekleyin. Örneğin `indexer_node4_name`ve `node4host`. Bunları aşağıdaki dizilere ekleyin:

- `indexer_discovery_hosts`
- `indexer_cluster_initial_master_nodes`
- `indexer_cluster_CN`
- `indexer_certs`

`puppet-wazuh-indexer-node2`Ek olarak, veya benzeri yeni bir düğüm örneği eklemeniz gerekir `puppet-wazuh-indexer-node3`. Wazuh dizinleyici node1 örneğinin aksine, bu örnekler . çalıştırmaz `securityadmin`.

Bir Wazuh yönetici çalışan sunucusu eklemeniz gerekirse, . gibi yeni bir değişken ekleyin `worker2host` . Değişkeni diziye ekleyin `manager_worker_certs`. Örneğin, `['worker',"$worker2host"]`. Ardından, düğüm örneğini `puppet-wazuh-manager-worker`yeni sunucuyla çoğaltın.

Dosyayı `/etc/puppetlabs/code/environments/production/manifests/`Puppet master'ınıza yerleştirin. Belirtilen düğümde `runinterval`, ayarlandığı gibi, zaman `puppet.conf`geçtikten sonra yürütülür. Ancak, bildirimi belirli bir düğümde hemen çalıştırmak istiyorsanız, düğümde aşağıdaki komutu çalıştırın:

```
puppet agent -t
```

Wazuh Kullanıcıları İçin Şifreyi Değiştir

Wazuh kullanıcı parolalarınızı değiştirmek için Parola Yönetimi bölümündeki talimatları izleyin . Parolaları değiştirdikten sonra, Wazuh Stack'i dağıtmak için kullanılan sınıflar içinde yeni parolaları ayarlayın.

Dizinleyici Kullanıcıları

- `admin`kullanıcı:

```
node "puppet-agent.com" {
  class { 'wazuh::dashboard':
    dashboard_password => '<NEW_PASSWORD>'
  }
}
```

- `kibanaserver`kullanıcı:

```
node "puppet-agent.com" {
  class { 'wazuh::filebeat_oss':
    filebeat_oss_elastic_password  => '<NEW_PASSWORD>'
  }
}
```

Wazuh API Kullanıcıları

- `wazuh-wui`kullanıcı:

```
node "puppet-agent.com" {
  class { 'wazuh::dashboard':
    dashboard_wazuh_api_credentials => '<NEW_PASSWORD>'
  }
}
```

Puppet Aracılığıyla Wazuh Agent Yükleyin

Sınıfın kurulmasıyla ajan yapılandırılır `wazuh::agent`.

İşte bir manifesto örneği `wazuh-agent.pp`(lütfen `<MANAGER_IP_ADDRESS>`yöneticinizin IP adresiyle değiştirin).

```
node "puppet-agent.com" {
  class { 'wazuh::repo':
  }
  class { "wazuh::agent":
    wazuh_register_endpoint => "<MANAGER_IP_ADDRESS>",
```

```
    wazuh_reporting_endpoint => "<MANAGER_IP_ADDRESS>"
  }
}
```

Dosyayı Puppet ana makinenize yerleştirin ve belirtilen düğümde ayarlanan zamandan `/etc/puppetlabs/code/environments/production/manifests/` sonra yürütülecektir . Ancak, önce çalıştırmak istiyorsanız, Puppet aracısında aşağıdaki komutu deneyin. `runinterval` `puppet.conf`

```
puppet agent -t
```

Referans Wazuh Kuklası

Bölümler	Değişkenler	Fonksiyonlar
Wazuh yönetici sınıfı	Uyarılar Yetkili Küme Küresel Yerel dosya Kök kontrolü Sistem kontrolü Syslog çıktısı Güvenlik Açığı Tespiti Wazuh API Wodle sorgusu Wodle Sistem Toplayıcısı Çeşitli	e-posta_uyarısı emretmek aktifcevap
Wazuh ajan sınıfı	Aktif Tepki Acente kaydı İstemci ayarları Yerel dosya Kök kontrolü SCA Sistem kontrolü Wodle sorgusu Wodle Sistem Toplayıcısı Çeşitli	

Puppet Aracılığıyla Bir Yığın Kurun

Bu modül Nicolas Zin tarafından yazılmış ve Jonathan Gazeley ve Michael Porter tarafından güncellenmiştir. Wazuh, onu sürdürme amacıyla çatallamıştır. Katkılarından dolayı yazarlara teşekkür ederiz.

Wazuh Modülünü Kurun

Puppet Forge'dan Wazuh modülünü indirin ve kurun:

```
puppet module install wazuh-wazuh --version 4.9.2
```

Output

```
Notice: Preparing to install into /etc/puppetlabs/code/environments/production/modules ...
Notice: Downloading from https://forgeapi.puppet.com ...
Notice: Installing -- do not interrupt ...
/etc/puppetlabs/code/environments/production/modules
└─┬ wazuh-wazuh (v4.9.2)
  ├── puppet-nodejs (v7.0.1)
  ├── puppet-selinux (v3.4.1)
  ├── puppetlabs-apt (v7.7.1)
  ├─┬ puppetlabs-concat (v6.4.0)
  │ └── puppetlabs-translate (v2.2.0)
  ├── puppetlabs-firewall (v2.8.1)
  ├─┬ puppetlabs-powershell (v4.1.0)
  │ └── puppetlabs-pwshlib (v0.10.1)
  └── puppetlabs-stdlib (v6.6.0)
```

Bu modül Wazuh aracısını ve yöneticisini kurar ve yapılandırır.

Puppet Aracılığıyla Bir Yığın Kurun

Tek Düğüm

Tek düğümlü bir yığını dağıtmak için aşağıda gösterilen bildirimi kullanabilirsiniz. Bu yığın şunlardan oluşur:

- Wazuh gösterge paneli
- Wazuh dizinleyici
- Wazuh yöneticisi
- Dosyabeat

Aşağıdaki içerikle `stack.pp`dosyayı oluşturun :`/etc/puppetlabs/code/environments/production/manifests/`

- `puppet-aio-node`: Puppet aracısının ana bilgisayar adı veya IP adresi.
- `puppet-server`: Wazuh modülü kurulduğunda Puppet sunucusunun ana bilgisayar adı veya IP adresi.

```
$discovery_type = 'single-node'
stage { 'certificates': }
stage { 'repo': }
stage { 'indexerdeploy': }
stage { 'securityadmin': }
stage { 'dashboard': }
stage { 'manager': }
Stage[certificates] -> Stage[repo] -> Stage[indexerdeploy] -> Stage[securityadmin] -> Stage[manager] -> Stage[
Exec {
timeout => 0,
}
node "puppet-server" {
class { 'wazuh::certificates':
  indexer_certs => [['node-1','127.0.0.1']],
  manager_certs => [['master','127.0.0.1']],
  dashboard_certs => ['127.0.0.1'],
  stage => certificates,
}
}
node "puppet-aio-node" {
class { 'wazuh::repo':
stage => repo,
}
class { 'wazuh::indexer':
  stage => indexerdeploy,
}
class { 'wazuh::securityadmin':
stage => securityadmin
}
class { 'wazuh::manager':
  stage => manager,
}
class { 'wazuh::filebeat_oss':
  stage => manager,
}
class { 'wazuh::dashboard':
  stage => dashboard,
```

```
}
}
```

Çoklu Düğüm

Aşağıdaki çoklu düğüm bildirimini kullanarak, aşağıdaki düğümlerden oluşan dağıtılmış bir yığını üç farklı sunucuya veya Sanal Makineye (VM) dağıtabilirsiniz.

- 3 dizinleyici düğümü
- Yönetici ana düğümü
- Yönetici işçi düğümü
- Pano düğümü

Her uygulamayı yüklediğiniz sunucuların IP adreslerini mutlaka eklemelisiniz.

```
$node1host   = '<WAZUH_INDEXER_NODE1_IP_ADDRESS>'
$node2host   = '<WAZUH_INDEXER_NODE2_IP_ADDRESS>'
$node3host   = '<WAZUH_INDEXER_NODE3_IP_ADDRESS>'
$masterhost    = '<WAZUH_MANAGER_MASTER_IP_ADDRESS>'
$workerhost    = '<WAZUH_MANAGER_WORKER_IP_ADDRESS>'
$dashboardhost = '<WAZUH_DASHBOARD_IP_ADDRESS>'
$indexer_node1_name = 'node1'
$indexer_node2_name = 'node2'
$indexer_node3_name = 'node3'
$master_name = 'master'
$worker_name = 'worker'
$cluster_size = '3'
$indexer_discovery_hosts = [$node1host, $node2host, $node3host]
$indexer_cluster_initial_master_nodes = [$node1host, $node2host, $node3host]
$indexer_cluster_CN = [$indexer_node1_name, $indexer_node2_name, $indexer_node3_name]
Define stage for order execution
stage { 'certificates': }
stage { 'repo': }
stage { 'indexerdeploy': }
stage { 'securityadmin': }
stage { 'dashboard': }
stage { 'manager': }
Stage[certificates] -> Stage[repo] -> Stage[indexerdeploy] -> Stage[securityadmin] -> Stage[manager] ->
Stage[dashboard]
Exec {
```

```
timeout => 0,
}
node "puppet-server" {
class { 'wazuh::certificates':
  indexer_certs => [["$indexer_node1_name","$node1host" ],["$indexer_node2_name","$node2host"
],["$indexer_node3_name","$node3host" ]],
  manager_master_certs => [["$master_name","$masterhost"]],
  manager_worker_certs => [["$worker_name","$workerhost"]],
  dashboard_certs => ["$dashboardhost"],
  stage => certificates
}
class { 'wazuh::repo':
stage => repo
}
}
node "puppet-wazuh-indexer-node1" {
class { 'wazuh::repo':
stage => repo
}
class { 'wazuh::indexer':
  indexer_node_name => "$indexer_node1_name",
  indexer_network_host => "$node1host",
  indexer_node_max_local_storage_nodes => "$cluster_size",
  indexer_discovery_hosts => $indexer_discovery_hosts,
  indexer_cluster_initial_master_nodes => $indexer_cluster_initial_master_nodes,
  indexer_cluster_CN => $indexer_cluster_CN,
  stage => indexerdeploy
}
class { 'wazuh::securityadmin':
indexer_network_host => "$node1host",
stage => securityadmin
}
}
node "puppet-wazuh-indexer-node2" {
class { 'wazuh::repo':
stage => repo
}
class { 'wazuh::indexer':
  indexer_node_name => "$indexer_node2_name",
```

```
  indexer_network_host => "$node2host",
  indexer_node_max_local_storage_nodes => "$cluster_size",
  indexer_discovery_hosts => $indexer_discovery_hosts,
  indexer_cluster_initial_master_nodes => $indexer_cluster_initial_master_nodes,
  indexer_cluster_CN => $indexer_cluster_CN,
  stage => indexerdeploy
}
}
node "puppet-wazuh-indexer-node3" {
class { 'wazuh::repo':
stage => repo
}
class { 'wazuh::indexer':
  indexer_node_name => "$indexer_node3_name",
  indexer_network_host => "$node3host",
  indexer_node_max_local_storage_nodes => "$cluster_size",
  indexer_discovery_hosts => $indexer_discovery_hosts,
  indexer_cluster_initial_master_nodes => $indexer_cluster_initial_master_nodes,
  indexer_cluster_CN => $indexer_cluster_CN,
  stage => indexerdeploy
}
}
node "puppet-wazuh-manager-master" {
class { 'wazuh::repo':
stage => repo
}
class { 'wazuh::manager':
  ossec_cluster_name => 'wazuh-cluster',
  ossec_cluster_node_name => 'wazuh-master',
  ossec_cluster_node_type => 'master',
  ossec_cluster_key => '01234567890123456789012345678912',
  ossec_cluster_bind_addr => "$masterhost",
  ossec_cluster_nodes => ["$masterhost"],
  ossec_cluster_disabled => 'no',
  stage => manager
}
class { 'wazuh::filebeat_oss':
  filebeat_oss_indexer_ip => "$node1host",
  wazuh_node_name => "$master_name",
```

```
  stage => manager
}
}
node "puppet-wazuh-manager-worker" {
class { 'wazuh::repo':
stage => repo
}
class { 'wazuh::manager':
  ossec_cluster_name => 'wazuh-cluster',
  ossec_cluster_node_name => 'wazuh-worker',
  ossec_cluster_node_type => 'worker',
  ossec_cluster_key => '01234567890123456789012345678912',
  ossec_cluster_bind_addr => "$masterhost",
  ossec_cluster_nodes => ["$masterhost"],
  ossec_cluster_disabled => 'no',
  stage => manager
}
class { 'wazuh::filebeat_oss':
  filebeat_oss_indexer_ip => "$node1host",
  wazuh_node_name => "$worker_name",
  stage => manager
}
}
node "puppet-wazuh-dashboard" {
class { 'wazuh::repo':
stage => repo,
}
class { 'wazuh::dashboard':
  indexer_server_ip  => "$node1host",
  manager_api_host   => "$masterhost",
  stage => dashboard
}
}
```

Manifest'te açıklanan kukla düğümlerinin IP adresleriyle olan ilişkisi şu şekildedir:

- `puppet-wazuh-indexer-node1`= `node1host`. Wazuh indeksleyici node1.
- `puppet-wazuh-indexer-node2`= `node2host`. Wazuh indeksleyici node2.
- `puppet-wazuh-indexer-node3`= `node3host`. Wazuh indeksleyici node3.
- `puppet-wazuh-manager-master`= `masterhost`. Wazuh yönetici ustası.

- `puppet-wazuh-manager-worker`= `workerhost`. Wazuh yönetici işçi.
- `puppet-wazuh-dashboard`= `dashboardhost`. Wazuh panosu düğümü.

Sınıfın , Wazuh modülünün kurulu olduğu `wazuh::certificates`Puppet sunucusunda (`puppet-server`) uygulanması gerekir. Bu, arşiv modülünün Wazuh yığın dağıtımındaki tüm sunuculara dosyaları dağıtmak için kullanılması nedeniyle gereklidir.

Daha fazla Wazuh dizinleyici düğümüne ihtiyacınız varsa, yeni değişkenler ekleyin. Örneğin `indexer_node4_name`ve `node4host`. Bunları aşağıdaki dizilere ekleyin:

- `indexer_discovery_hosts`
- `indexer_cluster_initial_master_nodes`
- `indexer_cluster_CN`
- `indexer_certs`

`puppet-wazuh-indexer-node2`Ek olarak, veya benzeri yeni bir düğüm örneği eklemeniz gerekir `puppet-wazuh-indexer-node3`. Wazuh dizinleyici node1 örneğinin aksine, bu örnekler . çalıştırmaz `securityadmin`.

Bir Wazuh yönetici çalışan sunucusu eklemeniz gerekirse, . gibi yeni bir değişken ekleyin `worker2host` . Değişkeni diziye ekleyin `manager_worker_certs`. Örneğin, `['worker',"$worker2host"]`. Ardından, düğüm örneğini `puppet-wazuh-manager-worker`yeni sunucuyla çoğaltın.

Dosyayı `/etc/puppetlabs/code/environments/production/manifests/`Puppet master'ınıza yerleştirin. Belirtilen düğümde `runinterval`, ayarlandığı gibi, zaman `puppet.conf`geçtikten sonra yürütülür. Ancak, bildirimi belirli bir düğümde hemen çalıştırmak istiyorsanız, düğümde aşağıdaki komutu çalıştırın:

```
puppet agent -t
```

Wazuh Kullanıcıları İçin Şifreyi Değiştir

Wazuh kullanıcı parolalarınızı değiştirmek için Parola Yönetimi bölümündeki talimatları izleyin . Parolaları değiştirdikten sonra, Wazuh Stack'i dağıtmak için kullanılan sınıflar içinde yeni parolaları ayarlayın.

Dizinleyici Kullanıcıları

- `admin`kullanıcı:

```
node "puppet-agent.com" {
  class { 'wazuh::dashboard':
    dashboard_password => '<NEW_PASSWORD>'
  }
}
```

- `kibanaserver`kullanıcı:

```
node "puppet-agent.com" {
  class { 'wazuh::filebeat_oss':
    filebeat_oss_elastic_password  => '<NEW_PASSWORD>'
  }
}
```

Wazuh API Kullanıcıları

- `wazuh-wui`kullanıcı:

```
node "puppet-agent.com" {
  class { 'wazuh::dashboard':
    dashboard_wazuh_api_credentials => '<NEW_PASSWORD>'
  }
}
```

Puppet Aracılığıyla Wazuh Agent Yükleyin

Sınıfın kurulmasıyla ajan yapılandırılır `wazuh::agent`.

İşte bir manifesto örneği `wazuh-agent.pp`(lütfen `<MANAGER_IP_ADDRESS>`yöneticinizin IP adresiyle değiştirin).

```
node "puppet-agent.com" {
  class { 'wazuh::repo':
  }
  class { "wazuh::agent":
    wazuh_register_endpoint => "<MANAGER_IP_ADDRESS>",
    wazuh_reporting_endpoint => "<MANAGER_IP_ADDRESS>"
  }
}
```

Dosyayı Puppet ana makinenize yerleştirin ve belirtilen düğümde ayarlanan zamandan `/etc/puppetlabs/code/environments/production/manifests/`sonra yürütülecektir . Ancak, önce çalıştırmak istiyorsanız, Puppet aracısında aşağıdaki komutu deneyin.`runinterval``puppet.conf`

```
puppet agent -t
```

Wazuh Kullanıcıları İçin Şifreyi Değiştirme

Wazuh kullanıcı parolalarınızı değiştirmek için Parola Yönetimi bölümündeki talimatları izleyin . Parolaları değiştirdikten sonra, Wazuh Stack'i dağıtmak için kullanılan sınıflar içinde yeni parolaları ayarlayın.

Dizinleyici Kullanıcıları

- `admin`kullanıcı:

```
node "puppet-agent.com" {
  class { 'wazuh::dashboard':
    dashboard_password => '<NEW_PASSWORD>'
  }
}
```

- `kibanaserver`kullanıcı:

```
node "puppet-agent.com" {
  class { 'wazuh::filebeat_oss':
    filebeat_oss_elastic_password  => '<NEW_PASSWORD>'
  }
}
```

Wazuh API Kullanıcıları

- `wazuh-wui`kullanıcı:

```
node "puppet-agent.com" {
  class { 'wazuh::dashboard':
    dashboard_wazuh_api_credentials => '<NEW_PASSWORD>'
  }
}
```

Puppet Aracılığıyla Wazuh Agent Yükleyin

Sınıfın kurulmasıyla ajan yapılandırılır `wazuh::agent`.

İşte bir manifesto örneği `wazuh-agent.pp`(lütfen `<MANAGER_IP_ADDRESS>`yöneticinizin IP adresiyle değiştirin).

```
node "puppet-agent.com" {
  class { 'wazuh::repo':
  }
  class { "wazuh::agent":
    wazuh_register_endpoint => "<MANAGER_IP_ADDRESS>",
    wazuh_reporting_endpoint => "<MANAGER_IP_ADDRESS>"
  }
}
```

Dosyayı Puppet ana makinenize yerleştirin ve belirtilen düğümde ayarlanan zamandan `/etc/puppetlabs/code/environments/production/manifests/`sonra yürütülecektir . Ancak, önce çalıştırmak istiyorsanız, Puppet aracısında aşağıdaki komutu deneyin.`runinterval``puppet.conf`

```
puppet agent -t
```

Puppet Aracılığıyla Wazuh Agent Yükleme

Sınıfın kurulmasıyla ajan yapılandırılır `wazuh::agent`.

İşte bir manifesto örneği `wazuh-agent.pp`(lütfen `<MANAGER_IP_ADDRESS>`yöneticinizin IP adresiyle değiştirin).

```
node "puppet-agent.com" {
  class { 'wazuh::repo':
  }
  class { "wazuh::agent":
    wazuh_register_endpoint => "<MANAGER_IP_ADDRESS>",
    wazuh_reporting_endpoint => "<MANAGER_IP_ADDRESS>"
  }
}
```

Dosyayı Puppet ana makinenize yerleştirin ve belirtilen düğümde ayarlanan zamandan `/etc/puppetlabs/code/environments/production/manifests/`sonra yürütülecektir . Ancak, önce çalıştırmak istiyorsanız, Puppet aracısında aşağıdaki komutu deneyin.`runinterval`puppet.conf

```
puppet agent -t
```

Wazuh Puppet Referansı