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	

Revision #5
Created 26 December 2024 01:36:01 by Ayşegül Sarıkaya
Updated 26 December 2024 02:01:29 by Ayşegül Sarıkaya