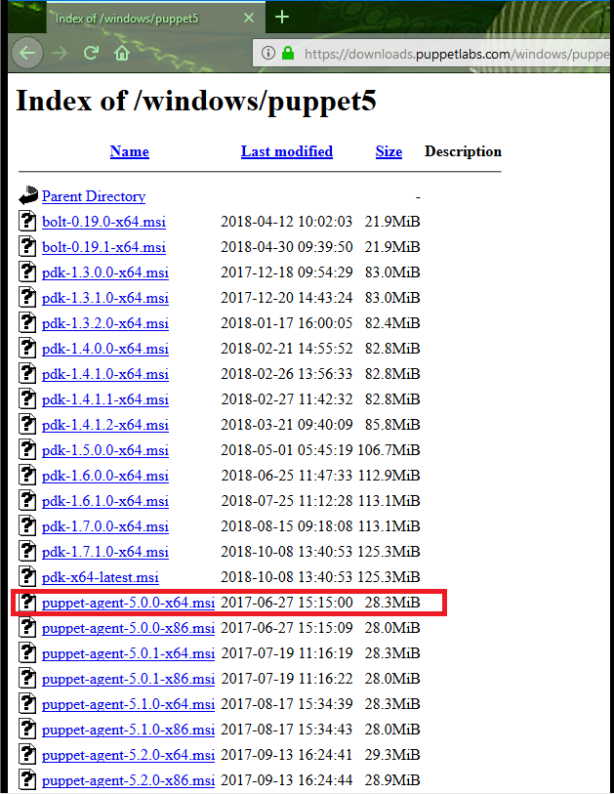
install Puppet agent on Windows server

Make sure Puppet server DNS name is resolvable from Windows host and vice-versa (skip this step if there is Host A DNS record for Puppet server):

C:\Windows\system32\drivers\etc\hosts

[192.168.1.97](https://docs.corp.hentsu.com/display/ENG/192.168.1.97) [puppetserver.example.com](https://docs.corp.hentsu.com/display/ENG/puppetserver.example.com)

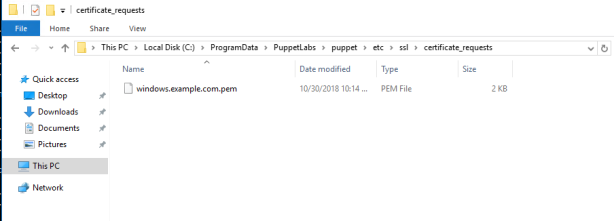
[Download](https://downloads.puppetlabs.com/windows/puppet5/) latest Puppet agent



Installation:

[puppet-agent-5.0.0-x64.msi](https://docs.corp.hentsu.com/display/ENG/puppet-agent-5.0.0-x64.msi) /qn PUPPET\_MASTER\_SERVER=[puppetserver.example.com](https://docs.corp.hentsu.com/display/ENG/puppetserver.example.com)PUPPET\_AGENT\_CERTNAME=[windows.example.com](https://docs.corp.hentsu.com/display/ENG/windows.example.com)

If all goes smooth, certificate request file should be seen in C:\ProgramData\PuppetLabs\puppet\etc\ssl\certificate\_requests

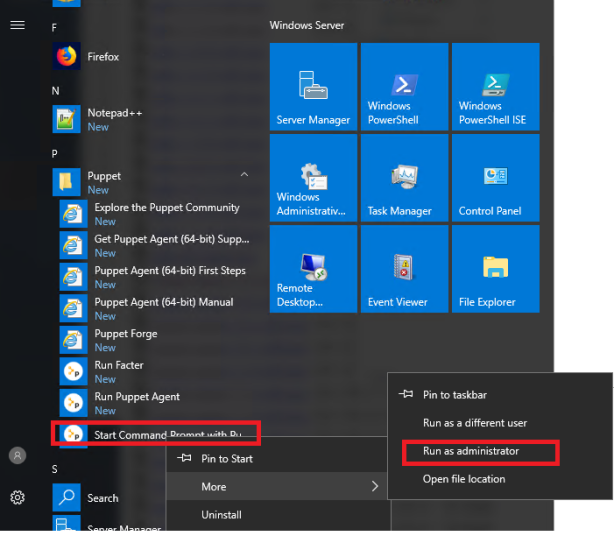


This certificate is waiting for signing, to do so, we need to move to Puppet server:

sudo puppet cert list

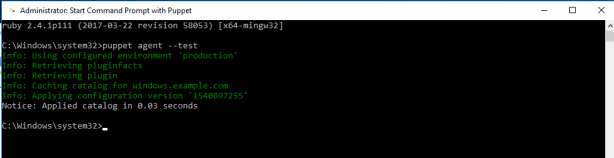
sudo puppet cert sign "[windows.example.com](https://docs.corp.hentsu.com/display/ENG/windows.example.com)"

On Windows server go to Puppet-Start Command Prompt with Puppet as Administrator



Test connection from Windows to Puppet server:

puppet agent --test

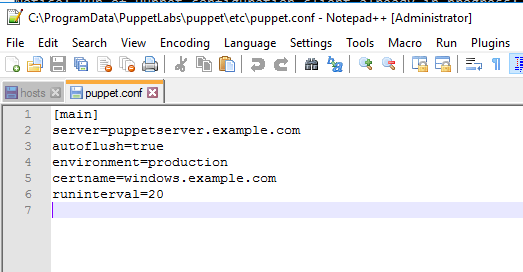


By default, Puppet agent polls Puppet server every 30 minutes.This behavior can be overridden by specifying custom polling interval by editing “C:\ProgramData\PuppetLabs\puppet\etc\puppet.conf” fileand adding line

runinterval=

One security concern is that catalog file stores credentials in plain text, to avoid this,prevent storing this file (which is in JSON format) to Puppet client

catalog\_cache\_terminus=""



**Creating  simple manifest file**

In this example we’ll make sure that IIS is installed

Install IIS module from Forge:

puppet module install puppetlabs-iis --version [4.5.0](https://docs.corp.hentsu.com/display/ENG/4.5.0)

Create manifest file and specify node and actions:

**/etc/puppetlabs/code/environments/production/manifests/site.pp**:

node '[windows.example.com'](https://docs.corp.hentsu.com/display/ENG/windows.example.com') {

$iis\_features = ['Web-WebServer','Web-Scripting-Tools','Web-Mgmt-Console']

iis\_feature { $iis\_features:

ensure => 'present',

include\_management\_tools => 'present'

}

}

Check for syntax errors:

puppet parser validate site.pp

**Run powershell scripts**

In order not to run script every time, when script runs if will create file with content “Done”

When Puppet tries to run this script again, it will check whether C:\log.txt or string “Done” exists in C:\log.txt, if both checks are true then script won’t be run (unless directive)

exec { 'configure\_gpo':

command => 'Set-ADDefaultDomainPasswordPolicy -Identity ad.contoso.com -ComplexityEnabled $true -MinPasswordLength 7 -MinPasswordAge 1 -MaxPasswordAge 30 -LockoutDuration 00:30:00 -LockoutObservationWindow 00:30:00 -LockoutThreshold 3;write-output "Done" | out-file C:\log.txt -Append',

unless => 'if (!(Test-Path C:\log.txt) -or !(Select-String -Path C:\log.txt -Pattern "Done")) {exit 1}',

provider => powershell,

logoutput => true,

}