**How to Setup WinRM in Windows Machine to Prepare for Ansible**

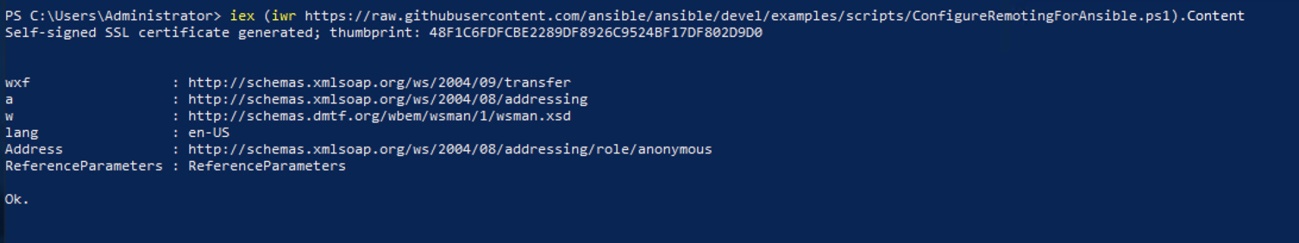
The First step for us to be able to connect to the windows machine is to install this WinRM properly on our Windows machine.

Thanks to Ansible team. they have created a PowerShell script that does the required configuration on the windows machine for us.

Do not worry about downloading the Powershell script file.  Just run the following powershell command in your PowerShell terminal

This downloads the script automatically and runs it in your terminal.

iex(iwr https://raw.githubusercontent.com/ansible/ansible/devel/examples/scripts/ConfigureRemotingForAnsible.ps1).Content



If the installation is done right. you can see that your WinRM is UP and running and would be listening in port 5986

Like in Linux, Windows has netstat command too

here is a quick command to for you to check if winrm listens on the port 5986

netstat -anp|findstr 5986

Thats all.

One more thing is pending. If you are on a cloud,

Consider opening this port to ansible control machine. So the Ansible can connect this machine from control machine. ( It is same like Opening up port 22 for linux to allow SSH)

**Ansible Configuration Changes on the Control machine.**

Hope you have already installed Ansible on your control machine ( Linux/Mac/Windows)

But there are few more extra packages you might need for Ansible to support windows modules

If you are using Ansible with Python 2 use PIP to install this package

pip install pywinrm

If you are using Ansible with Python3 use PIP3 to install pywinrm

pip3 install pywinrm

once the pywinrm package is installed we are all good and we can go and do a quick health check with ping.

But I would recommend you can use telnet or nc command (whichever available) to make sure that the network connectivity is there to the remote machine

nc -w 3 -v <remote windows server ip/hostname> 5986

(or)

telnet <remote windows server ip/hostname>:5986

This would give you an additional confidence that your connection is OK.

**Create or Update ansible hosts inventory file**

Before you can connect to the remote machine with Ansible.

you need to let Ansible know about this machine, as usual you need to add this machine to any hostgroup. In my case the host group name is win

**[**win**]**

192.9.12.122

You can keep the IP or the hostname which ever is reachable from your ansible control machine

Additionally, since this is windows, we need to provide some more variables at the hostgroup level.

**[**win:vars**]**

ansible\_connection=winrm

ansible\_user=administrator

ansible\_password=r$eBQNgc5U&A2at8kDwpWo.KzLT5NvHd

ansible\_winrm\_server\_cert\_validation=ignore

* ansible\_connection=winrm to define the connection is not SSH should use winrm
* ansible\_user what ever the username you have created in the windows machine
* ansible\_password password for that user ( the same one you use for RDP)
* ansible\_winrm\_server\_cert\_validation this is fine in DEV/TEST environment to tell ansible to ignore hostkey/server cert validation.

The complete inventory file is given below for your reference

**[**win**]**

192.9.12.122

**[**win:vars**]**

ansible\_connection=winrm

ansible\_user=administrator

ansible\_password=r$eBQNgc5U&A2at8kDwpWo.KzLT5NvHd

ansible\_winrm\_server\_cert\_validation=ignore

I have saved this file in my custom directory where I would create my playbooks and named this as ansible\_hosts

this is to keep things isolated and neat. You can directly add in the ansible global inventory if you want

Now its a time to test.

**Win\_ping - Ping the remote windows machine using Ansible.**

Even if you are a beginner in Ansible, I presume you might have come across ansible ping module.

Ansible ping is to check the connection from control machine to remote linux machine.

Likewise, Ansible win\_ping is to check the connectivity from Control machine to Windows.

It is like a Hello world of programming language we can say.

So we are going execute the following command

ansible win -m win\_ping -i ansible\_hosts

here the win is our host group name and with -m we are telling ansible to use win\_ping module

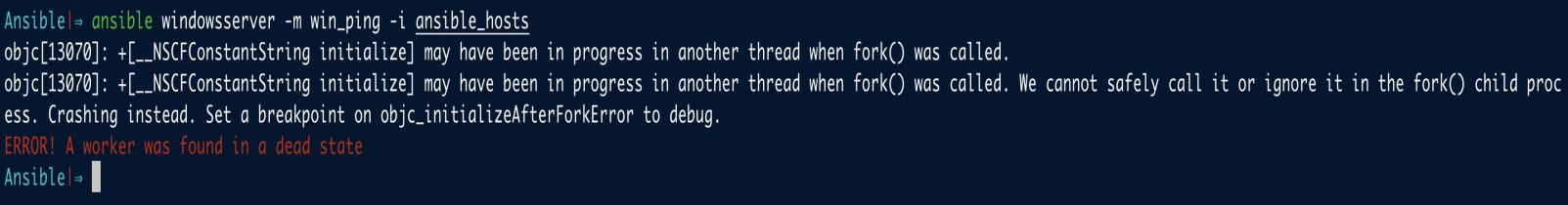
**Quick Note for Mac users - Python Crashing issue**

While you are executing this command, you might get a pop up and can see the Python is crashing.

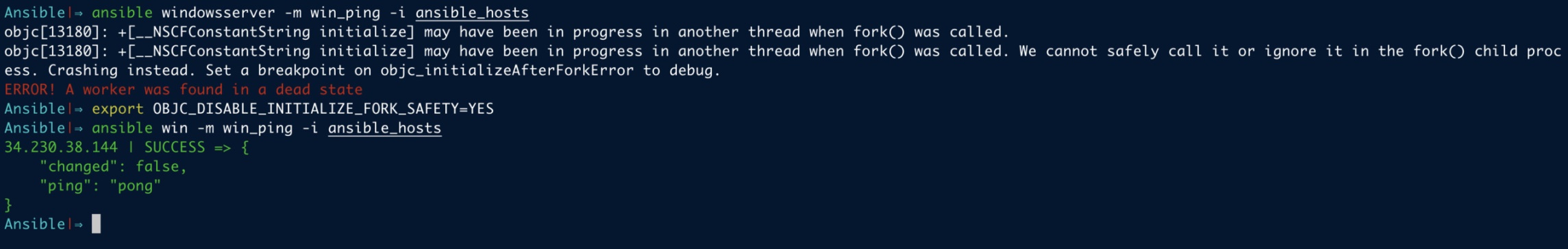
You might see either of these or both error messages which can be solved by setting this environment variable

export OBJC\_DISABLE\_INITIALIZE\_FORK\_SAFETY=YES

Error messages are given below for your reference.



Once you have got the successful ping message. You are All good to test your Playbooks and other commands.



**Validate Other Ansible AD Hoc commands and Playbooks**

Once the win\_ping is green. you can execute some other modules and commands either as ad\_hoc or as playbook to test it

here is a quick playbook you can use which executes a command on the remote server

---

- name: Windows Test Playbook

hosts: win

tasks:

- name: Remote Execute the mqsc files

win\_shell: |

hostname

Get-Date

register: scriptoutput

- name: Script output

debug: var=scriptoutput.stdout

The same playbook can be executed as two ansible ad hoc commands

ansible win -m win\_shell -a "hostname" -i ansible\_hosts

ansible win -m win\_shell -a "Get-Date" -i ansible\_hosts

Voila. You did it.