Windows Managed Node Setup

Setup a Windows host - local UI

Setting up a Windows Server to be managed by Ansible involves a few key steps. Ansible communicates with Windows servers over WinRM (Windows Remote Management), which is a Windows-native remote management protocol based on WS-Management (Web Services-Management). The setup process generally includes configuring WinRM on the Windows server and preparing the Ansible control machine to manage Windows hosts.

Here are the steps to prepare a Windows Server for management with Ansible:

1. Configure WinRM on the Windows Server

The easiest way to configure WinRM for Ansible is to use the `ConfigureRemotingForAnsible.ps1` script, which is provided in the Ansible documentation. This script sets up WinRM to use basic authentication and configures it to allow connections from Ansible.

1. **Download the Script**: On the Windows Server, open PowerShell as an Administrator and run the following command to download the script:

   ```
   Invoke-WebRequest -Uri "https://raw.githubusercontent.com/ansible/ansible-documentation/devel/examples/scripts/ConfigureRemotingForAnsible.ps1" -OutFile "ConfigureRemotingForAnsible.ps1"
   ```

2. **Run the Script**: Execute the script you just downloaded:

   ```
   .\ConfigureRemotingForAnsible.ps1
   ```

   This script will configure WinRM to use HTTP (port 5985), enable basic authentication, and create a firewall rule to allow WinRM traffic.

3. **Note**: For a production environment, it's recommended to use HTTPS (port 5986) with certificate-based authentication for increased security. This setup is more complex and requires installing a valid certificate on the Windows Server and additional WinRM configuration.

2. Prepare the Ansible Control Machine

On the Ansible control machine, which is typically a Linux system, you need to install `pywinrm` to enable WinRM support. This can be done using pip:

```
pip install pywinrm
```

3. Configure Ansible Inventory

Edit your Ansible inventory file to include your Windows hosts. You can define them under a specific group `[windows]` and specify the necessary variables:

```
[windows]
windows-server.example.com

[windows:vars]
ansible_user=Administrator
ansible_password=YourPassword
ansible_connection=winrm
ansible_winrm_server_cert_validation=ignore
```

Security Note: Storing passwords in plaintext in the inventory file is not secure. Consider using Ansible Vault to encrypt sensitive data.

4. Test the Configuration

Now, test your setup by running a simple Ansible command to ping the Windows server:

```
ansible windows -m win_ping
```

If everything is configured correctly, the `win_ping` module should return a success message.

Additional Notes

- Ensure network connectivity between the Ansible control machine and the Windows Server, specifically that the required WinRM port (5985 for HTTP, 5986 for HTTPS) is open.
- The setup process might vary slightly depending on the specific version of Windows Server you are using.
- For production environments, it's highly recommended to use Kerberos or NTLM with WinRM over HTTPS for secure authentication and encryption.

By following these steps, you should have a Windows Server ready to be managed by Ansible.

Remote install / fleet deployments

To remotely set up a Windows Server to be managed by Ansible, you need to configure WinRM (Windows Remote Management) on the target server. This process can be challenging since it ideally requires remote execution of a configuration script on the Windows server. If you have physical access or remote desktop (RDP) access to the server, it's usually easier to set up WinRM directly. However, if you need to do this entirely remotely, here are some methods you can consider:

1. Using PowerShell Remoting

If PowerShell Remoting is already enabled on the target server, you can use it to configure WinRM for Ansible.

1. **Connect to the Windows Server via PowerShell Remoting**:

   ```
   $credential = Get-Credential
   Enter-PSSession -ComputerName <Target-Server-IP-or-Hostname> -Credential $credential
   ```

2. **Run the Ansible WinRM Configuration Script**: Download and execute the `ConfigureRemotingForAnsible.ps1` script within the remote session.

   ```
   Invoke-WebRequest -Uri "https://raw.githubusercontent.com/ansible/ansible-documentation/devel/examples/scripts/ConfigureRemotingForAnsible.ps1" -OutFile "ConfigureRemotingForAnsible.ps1"
   .\ConfigureRemotingForAnsible.ps1
   ```

3. **Exit the Remote Session**:

   ```
   Exit-PSSession
   ```

2. Using Group Policy (For Domain-Joined Servers)

If the server is part of an Active Directory domain, you can use Group Policy to configure WinRM on multiple servers at once.

1. **Create a new GPO** in your Group Policy Management Console.
2. **Edit the GPO** to include the WinRM service configuration. Typically, you need to set up the service to start automatically and configure the listener for HTTP and/or HTTPS.
3. **Link the GPO** to an OU that contains your servers.

3. Using a Configuration Management Tool

If you have a configuration management tool like SCCM (System Center Configuration Manager), you can use it to push out a script or configuration to enable and configure WinRM on Windows servers.

4. Using a Remote Execution Tool

If you have access to a remote execution tool like PSExec (part of Sysinternals), you can use it to run commands or scripts on the remote Windows server.

For example:

```
psexec \\target-server -u username -p password -h powershell.exe -ExecutionPolicy Bypass -File
ConfigureRemotingForAnsible.ps1
```

Security Considerations

- When setting up WinRM, especially over HTTP, be aware of security implications. HTTP traffic is not encrypted, which can expose sensitive data. For production environments, HTTPS with certificate-based authentication is recommended.
- Ensure that the WinRM service is properly secured and accessible only from trusted networks or hosts.

Testing the Setup

After setting up WinRM, test the connection from your Ansible control machine:

```
ansible windows -i inventory_file -m win_ping
```

Replace `inventory_file` with the path to your Ansible inventory file where your Windows host is defined.

Conclusion

The method you choose depends on your current infrastructure, the tools you have available, and your access level to the Windows Server. For security and simplicity, direct access (like RDP) to set up WinRM is generally preferred, but in environments where this is not feasible, remote methods are necessary.

Revision #3
Created 22 March 2019 21:03:31 by Michael Cleary
Updated 6 February 2024 13:25:52 by Michael Cleary