



Installing the Denodo Solution Manager with Ansible

Revision 20220331

NOTE

This document is confidential and proprietary of **Denodo Technologies**.
No part of this document may be reproduced in any form by any means without prior
written authorization of **Denodo Technologies**.

Copyright © 2024
Denodo Technologies Proprietary and Confidential

1 INTRODUCTION

Ansible is a configuration management tool that facilitates the task of setting up and maintaining remote servers. It uses SSH and Python to communicate and execute commands on managed servers.

Ansible allows users to manage servers in two different ways: via ad hoc commands, and via playbooks. Playbooks are YAML files containing a list of ordered tasks that should be executed on remote servers to complete a task or reach a certain goal.

This document describes how to install and configure the Denodo Solution Manager with Ansible playbooks on remote servers.

2 PREREQUISITES

- **One Ansible Control Node:** The Ansible control node is the machine we will use to connect to and control the remote servers over SSH.
- **One or more Ansible Hosts:** An Ansible host is any remote server that the Ansible control node is configured to automate the installation.

3 ANSIBLE PREPARATION

3.1 INSTALL ANSIBLE ON THE CONTROL NODE

In this example we are using Ubuntu 20.04 as the operating system of all the nodes. The first step is to refresh the system's package with:

```
# sudo apt update
```

And then install the Ansible software with:

```
# sudo apt install ansible
```

The guide [How to Install and Configure Ansible on Ubuntu 20.04](#) provides more details about how to set up Ansible.

3.2 CREATE AN ANSIBLE INVENTORY FILE

The Ansible inventory contains information about all the remote servers that Ansible controls. We can edit the default inventory file created when Ansible is installed at **/etc/ansible/hosts** on the Ansible control node. The guide [How To Set Up Ansible Inventories](#) explains in detail how to create an Ansible inventory file.

Once the information about the remote servers is added in the inventory file, the next step is to run the following command to check the inventory file.

In this example, there is one remote server called server1 in the servers group.

```
# ansible-inventory -list -y
```

```
all:
  children:
    servers:
      hosts:
        server1:
          ansible_host: 172.17.0.3
          ansible_python_interpreter: /usr/bin/python3
      ungrouped: {}
```

3.3 TESTING CONNECTIVITY TO REMOTE SERVERS

To test if the connection from the control node to the remote servers over SSH works fine, we use the following command and make sure it returns Success for every remote server.

```
# ansible all -m ping
```

```
server1 | SUCCESS => {  
  "changed": false,  
  "ping": "pong"  
}
```

4 SOLUTION MANAGER INSTALLATION

4.1 DOWNLOAD THE SOLUTION MANAGER INSTALLER

Download the Solution Manager installer (denodo-install-solution manager-8.0-ga-linux64) from Denodo Support Site in the following path of the Ansible control node (/opt/denodo-install-solutionmanager-8.0-ga-linux64.zip).

4.2 CREATE THE RESPONSE FILE ON THE CONTROL NODE

To generate a response file to automate the installation, we follow the steps in our documentation [Unattended Installation of the Solution Manager](#). In this example, the response file will be generated in the path /opt/denodo-install-solutionmanager-8.0/response_file_8_0.xml on the Ansible control node.

4.3 CREATE THE ANSIBLE PLAYBOOK FILE

Create an Ansible playbook containing a list of ordered tasks that will be executed on remote servers to install the Solution Manager. The details about specific tasks can be seen under the **tasks** label.

```
---
- hosts: all
  become: true
  name: Install Denodo Solution Manager
  tasks:
    - name: Copy and unzip the Solution Manager Installer to remote servers
      unarchive:
        src: /opt/denodo-install-solutionmanager-8.0-ga-linux64.zip
        dest: /opt/

    - name: Copy the response file from control node to remote servers
      copy:
        src: /opt/denodo-install-solutionmanager-8.0/response_file_8_0.xml
        dest: /opt/denodo-install-solutionmanager-8.0/

    - name: Change the permission
      shell: chmod +x /opt/denodo-install-solutionmanager-8.0/installer_cli.sh

    - name: Start the unattended installation
      shell: /opt/denodo-install-solutionmanager-8.0/installer_cli.sh
      install --autoinstaller response_file_8_0.xml
      register: ps
```

```
# Print the shell task's stout
- debug: var=ps.stdout_lines
```

4.4 RUN THE ANSIBLE PLAYBOOK

The next step is to run the Ansible `denodo_playbook.yml` from the control node with `ansible-playbook` command to start the unattended installation on remote servers:

```
# ansible-playbook denodo_playbook.yml
```

We will see the following page after the automated installation is done:

```
TASK [Start the unattended installation] *****
changed: [server1]

TASK [debug] *****
ok: [server1] => {
  "ps.stdout_lines": [
    "Installing Denodo Platform ...",
    "[ Starting automated installation ]",
    "Read pack list from xml definition.",
    "Modify pack selection.",
    "[ Starting to unpack ]",
    "[ Processing package: Common (1/12) ]",
    "[ Processing package: Virtual DataPort (2/12) ]",
    "[ Processing package: Administration Tool (3/12) ]",
    "[ Processing package: Web Design Studio (4/12) ]",
    "[ Processing package: Virtual DataPort Server (5/12) ]",
    "[ Processing package: Diagnostic & Monitoring Tool (6/12) ]",
    "[ Processing package: Scheduler (7/12) ]",
    "[ Processing package: Administration Tool (8/12) ]",
    "[ Processing package: Solution Manager (9/12) ]",
    "[ Processing package: Solution Manager Server (10/12) ]",
    "[ Processing package: License Manager Server (11/12) ]",
    "[ Processing package: Solution Manager Web Tool (12/12) ]",
    "No update was found in /opt/denodo_update/denodo-install-solutionmanager-8.0/denodo-update/denodo-update.jar",
    "[ Unpacking finished ]",
    "[ Writing the uninstaller data ... ]",
    " - Merging executables...",
    " - Merging executables...",
    " - Merging install files...",
    " - Merging executables...",
    " - Merging install files...",
    " - Merging executables...",
    "[ Automated installation done ]"
  ]
}

PLAY RECAP *****
server1 : ok=5  changed=2  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

5 CONCLUSION

In this article, we have shown you an example to execute Ansible playbook from the control node to install the Denodo Solution Manager on remote servers. You can use the similar steps to install Denodo Platform or automate other Denodo configurations.

6 REFERENCES

[Solution Manager Installation Guide - Unattended Installation of the Solution Manager Ansible User Guide](#)
[How To Install and Configure Ansible on Ubuntu 20.04](#)
[How To Set Up Ansible Inventories](#)
[How to Write Ansible Playbooks](#)