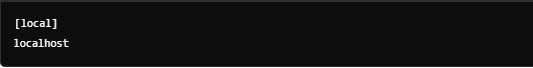
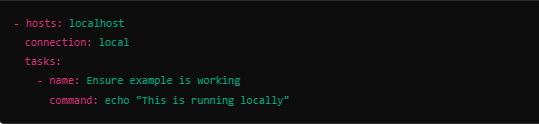
**Name: Mohammed Abdelnaby**

***Lab1 – Ansible***

1. **What inventory parameters can be used to establish a local connection instead of ssh in Ansible?**

In Ansible, when you want to establish a local connection instead of using SSH, you can specify the connection type as local. This is particularly useful if you want to run the playbooks on the same machine where Ansible is being executed (i.e., localhost). Here are the key inventory parameters you can use:

1. **Connection type**: Set ansible\_connection to local for the target host in the inventory file. 
2. **Hosts file**: Instead of adding remote hosts, you can simply define localhost. 
3. **Setting the local connection in playbooks**: Alternatively, you can define the connection type directly in the playbook: 
4. **Ansible Inventory Parameters**:

* ansible\_connection=local: Specifies that the connection should be local rather than over SSH.
* ansible\_host=localhost: Points to the local host machine.
* ansible\_python\_interpreter: If needed, you can also define the Python interpreter for the local machine, but it's often detected automatically.

**2-What value we must set for ansible\_connection parameter to connect to a Windows server?**

To connect to a Windows server using Ansible, you need to set the ansible\_connection parameter to winrm (Windows Remote Management). This is because Ansible uses WinRM as the transport mechanism for Windows hosts instead of SSH, which is typically used for Linux hosts.

windows\_host ansible\_host=192.168.1.100 ansible\_user=your\_username ansible\_password=your\_password ansible\_port=5986 ansible\_connection=winrm ansible\_winrm\_server\_cert\_validation=ignore

**3-We have a sample inventory file called inventory. It has 3 servers listed, add another server called server4.company.com in this file.**

**bash**

**web1**

**web2**

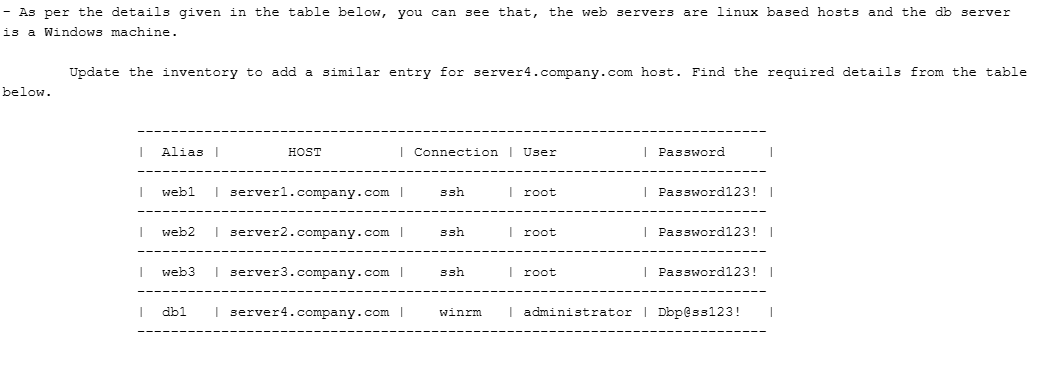
**Then added the aliases named web1, web2 and web3 for the first three hosts respectively. Update this inventory file to add an alias called db1 for server4.company.com host.**

**bash**

**web1 ansible\_host=server1.company.com**

**web2 ansible\_host=server2.company.com**

****

**4-**

Based on the table you provided, here is the updated inventory file. It includes the existing web servers (Linux hosts) and the new Windows-based database server with the appropriate WinRM connection details.

# Linux Web Servers

web1 ansible\_host=server1.company.com ansible\_connection=ssh ansible\_user=root ansible\_password=Password123!

web2 ansible\_host=server2.company.com ansible\_connection=ssh ansible\_user=root ansible\_password=Password123!

web3 ansible\_host=server3.company.com ansible\_connection=ssh ansible\_user=root ansible\_password=Password123!

# Windows Database Server

db1 ansible\_host=server4.company.com ansible\_connection=winrm ansible\_user=administrator ansible\_password=Dbp@ss123! ansible\_port=5986 ansible\_winrm\_server\_cert\_validation=ignore

We have inventory file and added a group called web\_servers for web servers. Similarly, add a group called db\_servers for database servers.

bash

# Sample Inventory File

# Web Servers

web1 ansible\_host=server1.company.com ansible\_connection=ssh ansible\_user=root ansible\_ssh\_pass=Password123!

web2 ansible\_host=server2.company.com ansible\_connection=ssh ansible\_user=root ansible\_ssh\_pass=Password123!

web3 ansible\_host=server3.company.com ansible\_connection=ssh ansible\_user=root ansible\_ssh\_pass=Password123!

# Database Servers

db1 ansible\_host=server4.company.com ansible\_connection=winrm ansible\_user=administrator ansible\_password=Password123!

[web\_servers]

web1

web2

web3

To add a group called db\_servers for database servers, you can update the inventory file as follows. This will group the database server (db1) under the new db\_servers group, just lik# Sample Inventory File

# Web Servers

web1 ansible\_host=server1.company.com ansible\_connection=ssh ansible\_user=root ansible\_ssh\_pass=Password123!

web2 ansible\_host=server2.company.com ansible\_connection=ssh ansible\_user=root ansible\_ssh\_pass=Password123!

web3 ansible\_host=server3.company.com ansible\_connection=ssh ansible\_user=root ansible\_ssh\_pass=Password123!

# Database Servers

db1 ansible\_host=server4.company.com ansible\_connection=winrm ansible\_user=administrator ansible\_password=Password123!

[web\_servers]

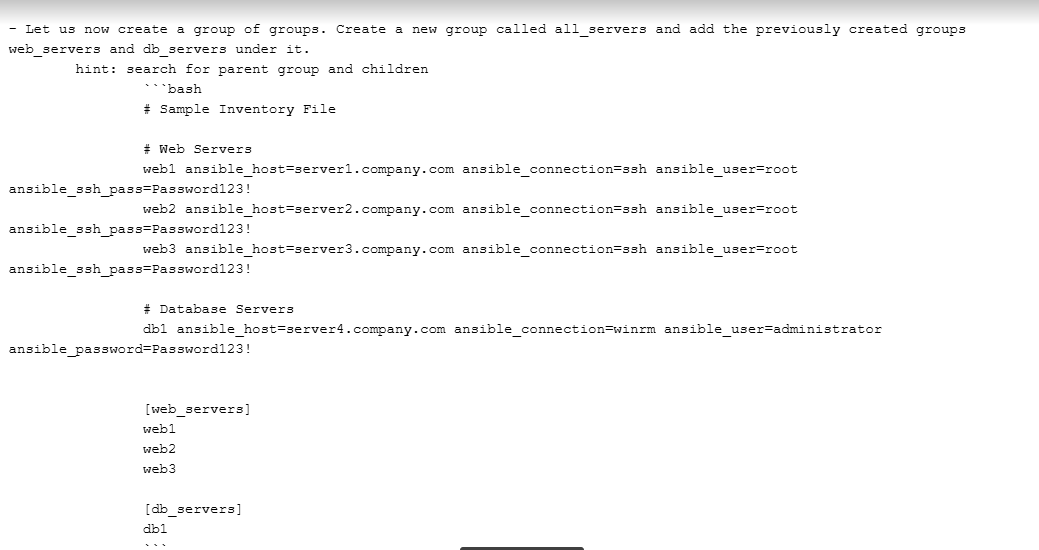
web1

web2

web3

[db\_servers]

db1e the web\_servers group for the web servers.



To create a "group of groups" in Ansible, you can define a parent group that includes multiple child groups. In your case, we will create a new group called all\_servers and include the previously created groups web\_servers and db\_servers as its children.

Here’s how you can update your inventory file:

# Sample Inventory File

# Web Servers

web1 ansible\_host=server1.company.com ansible\_connection=ssh ansible\_user=root ansible\_ssh\_pass=Password123!

web2 ansible\_host=server2.company.com ansible\_connection=ssh ansible\_user=root ansible\_ssh\_pass=Password123!

web3 ansible\_host=server3.company.com ansible\_connection=ssh ansible\_user=root ansible\_ssh\_pass=Password123!

# Database Servers

db1 ansible\_host=server4.company.com ansible\_connection=winrm ansible\_user=administrator ansible\_password=Password123!

# Group for Web Servers

[web\_servers]

web1

web2

web3

# Group for Database Servers

[db\_servers]

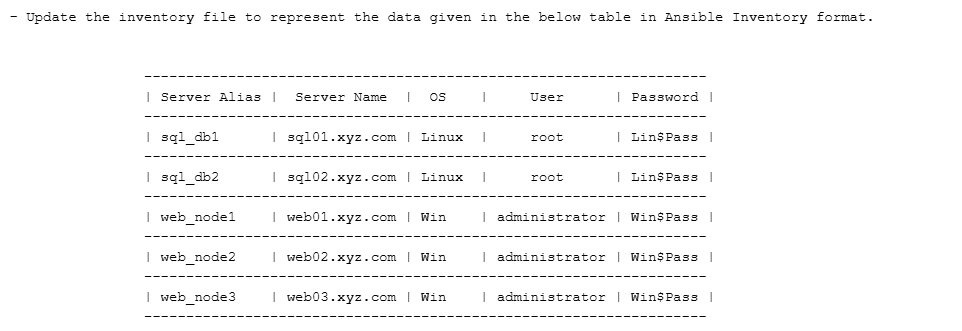
db1

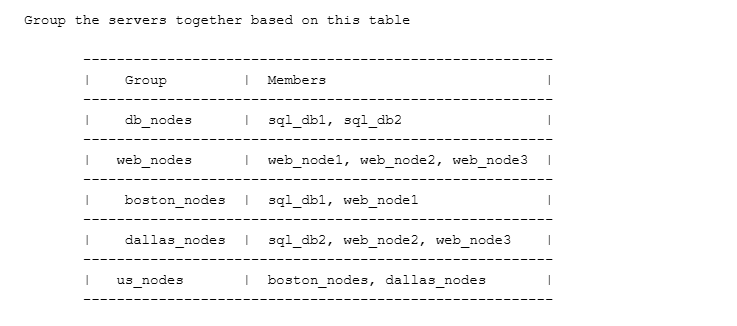
# Parent Group that includes both web\_servers and db\_servers

[all\_servers:children]

web\_servers

db\_servers





Here’s how you can represent the data from the table in an Ansible inventory file, grouping the servers together based on their respective groupings:

# Database Nodes

sql\_db1 ansible\_host=sql01.xyz.com ansible\_connection=ssh ansible\_user=root ansible\_password=Lin$Pass

sql\_db2 ansible\_host=sql02.xyz.com ansible\_connection=ssh ansible\_user=root ansible\_password=Lin$Pass

# Web Nodes (Windows)

web\_node1 ansible\_host=web01.xyz.com ansible\_connection=winrm ansible\_user=administrator ansible\_password=Win$Pass ansible\_port=5986 ansible\_winrm\_server\_cert\_validation=ignore

web\_node2 ansible\_host=web02.xyz.com ansible\_connection=winrm ansible\_user=administrator ansible\_password=Win$Pass ansible\_port=5986 ansible\_winrm\_server\_cert\_validation=ignore

web\_node3 ansible\_host=web03.xyz.com ansible\_connection=winrm ansible\_user=administrator ansible\_password=Win$Pass ansible\_port=5986 ansible\_winrm\_server\_cert\_validation=ignore

# Group for Database Nodes

[db\_nodes]

sql\_db1

sql\_db2

# Group for Web Nodes

[web\_nodes]

web\_node1

web\_node2

web\_node3

# Group for Boston Nodes

[boston\_nodes]

sql\_db1

web\_node1

# Group for Dallas Nodes

[dallas\_nodes]

sql\_db2

web\_node2

web\_node3

# Group for US Nodes (Group of Groups)

[us\_nodes:children]

boston\_nodes

dallas\_nodes