Standard Operating Procedure (SOP)

Environmental Details:

Here we have taken 3 hosts, 192.168.31.88 (control node), 2 remote nodes 192.168.31.55/56

```
----------------------------------------
Ansible control node: 192.168.31.88

Client nodes: 192.168.31.55, 192.168.31.56
```

Prechecks:

1. **Establish Passwordless Connection (steps on control node):**

   From control node perform the ssh-keygen operation, once it's done, copy directly to the client nodes for passwordless connectivity.

   ```
   ssh-keygen
   ssh-copy-id root@192.168.31.56
   ssh-copy-id root@192.168.31.55
   ```

2. **Install Ansible On Control Node:**

   Install Ansible over control node, this is very basic configuration, many additional modules will required as per the requirement.

   ```
   sudo yum update && \
   sudo yum install -y epel-release && \
   sudo yum install -y ansible && \
   ansible --version
   ```

3. **Add Client Hosts to Control Node:**

   On control node we are adding remote node ip's over "/etc/ansible/hosts" file.

   ```
   [root@console postgresql]# cat /etc/ansible/hosts|grep -i 192.168.31
   192.168.31.56
   192.168.31.55
   ```

4. **Check Ping Status:**

   ```
   [root@console postgresql]# ansible all -m ping
   192.168.31.55 | SUCCESS => {
       "ansible_facts": {
           "discovered_interpreter_python": "/usr/bin/python"
       },
       "changed": false,
       "ping": "pong"
   }
   192.168.31.56 | SUCCESS => {
       "ansible_facts": {
           "discovered_interpreter_python": "/usr/bin/python"
       },
       "changed": false,
       "ping": "pong"
   ```

5. **Check RPM Package Availability:**

   Before our operation this is for verification that no packages are available over the remote sides,(there are more other ways to verify). Here we got error as there are no details available.

   ```
   [root@console postgresql]# ansible all -m shell -a "rpm -qa|grep -i postgres"
   [WARNING]: Consider using the yum, dnf or zypper module rather than running 'rpm'.  If you need to use command because yum, dnf or
   zypper is insufficient you can add 'warn: false' to this command task or set 'command_warnings=False' in ansible.cfg to get rid of
   this message.
   192.168.31.55 | FAILED | rc=1 >>
   non-zero return code
   192.168.31.56 | FAILED | rc=1 >>
   non-zero return code

   [root@console postgresql]# ansible all -m shell -a "id postgres"
   192.168.31.55 | FAILED | rc=1 >>
   id: postgres: no such usernon-zero return code
   192.168.31.56 | FAILED | rc=1 >>
   ```

```
id: postgres: no such usernon-zero return code
[root@console postgresql]#
```

6. **Prepare the Playbook File:**

```
[root@console postgresql]# cat 1-Instal_postgresql_utls.yml
---
- name: Install postgresql on client machine
  hosts: all


  tasks:
  - name: Task1-Create Necessary  directories
    file:
     path: /postgres/
     state: directory
     owner: root
     group: root
     mode: 0755

  - name: Task2-get the rpm download
    get_url:
     url: https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86_64/pgdg-redhat-repo-latest.noarch.rpm
     dest: /etc/yum.repos.d/pgdg-redhat-repo-latest.noarch.rpm
     owner: root
     group: root
     mode: 0755

  - name: Task3-create repository
    yum:
     name: /etc/yum.repos.d/pgdg-redhat-repo-latest.noarch.rpm
     state: present

  - name: Task4-install PostgreSQL 15
    yum:
     name: "{{item}}"
     state: present
    loop:
     - postgresql15.x86_64
     - postgresql15-contrib.x86_64
     - postgresql15-server.x86_64

  - name: Task5-Initialize PostgreSQL 15 database cluster
    shell: /usr/pgsql-15/bin/postgresql-15-setup initdb

  - name: Task-6 Start PostgreSQL service
    service:
     name: postgresql-15
     state: started

  - name: Task7- Enable Postgresql Service
    service:
     service: postgresql-15
     permanent: true
     state: enabled
```

7. **Check Syntax:**

```
[root@console postgresql]# ansible-playbook --syntax-check 1-Instal_postgresql_utls.yml

playbook: 1-Instal_postgresql_utls.yml
[root@console postgresql]#
```

8. **Run Playbook:**

```
[root@console postgresql]# ansible-playbook  1-Instal_postgresql_utls.yml

PLAY [Install postgresql on client machine] ************************************************************************

TASK [Gathering Facts] *********************************************************************************************
ok: [192.168.31.55]
ok: [192.168.31.56]

TASK [Task1-Create Necessary  directories] *************************************************************************
ok: [192.168.31.56]
ok: [192.168.31.55]

TASK [Task2-get the rpm download] **********************************************************************************
changed: [192.168.31.55]
ok: [192.168.31.56]

TASK [Task3-create repository] *************************************************************************************
ok: [192.168.31.55]
ok: [192.168.31.56]

TASK [Task4-install PostgreSQL 15] *********************************************************************************
changed: [192.168.31.56] => (item=postgresql15.x86_64)
```

```
changed: [192.168.31.56] => (item=postgresql15-contrib.x86_64)
ok: [192.168.31.56] => (item=postgresql15-server.x86_64)
changed: [192.168.31.55] => (item=postgresql15.x86_64)
changed: [192.168.31.55] => (item=postgresql15-contrib.x86_64)
ok: [192.168.31.55] => (item=postgresql15-server.x86_64)

TASK [Task5-Initialize PostgreSQL 15 database cluster] ************************************************************************
changed: [192.168.31.56]
changed: [192.168.31.55]

TASK [Task-6 Start PostgreSQL service] ****************************************************************************************
changed: [192.168.31.55]
changed: [192.168.31.56]

TASK [Task7- Enable Postgresql Service] ***************************************************************************************
changed: [192.168.31.56]
changed: [192.168.31.55]

PLAY RECAP ********************************************************************************************************************
192.168.31.55              : ok=8    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
192.168.31.56              : ok=8    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@console postgresql]#
```

9. **Verify Configurations:**

```
[root@console postgresql]# ansible all -m shell -a "id postgres"
192.168.31.55 | CHANGED | rc=0 >>
uid=26(postgres) gid=26(postgres) groups=26(postgres)
192.168.31.56 | CHANGED | rc=0 >>
uid=26(postgres) gid=26(postgres) groups=26(postgres)

[root@console postgresql]# ansible all -m shell -a "rpm -qa|grep -i postgres"
[WARNING]: Consider using the yum, dnf or zypper module rather than running 'rpm'.  If you need to use command because yum, dnf or
zypper is insufficient you can add 'warn: false' to this command task or set 'command_warnings=False' in ansible.cfg to get rid of
this message.
192.168.31.55 | CHANGED | rc=0 >>
postgresql15-libs-15.7-1PGDG.rhel7.x86_64
postgresql15-server-15.7-1PGDG.rhel7.x86_64
postgresql15-contrib-15.7-1PGDG.rhel7.x86_64
postgresql15-15.7-1PGDG.rhel7.x86_64
192.168.31.56 | CHANGED | rc=0 >>
postgresql15-contrib-15.7-1PGDG.rhel7.x86_64
postgresql15-libs-15.7-1PGDG.rhel7.x86_64
postgresql15-15.7-1PGDG.rhel7.x86_64
postgresql15-server-15.7-1PGDG.rhel7.x86_64
[root@console postgresql]#
```

10. **Verify Running Status of PostgreSQL Servers:**

```
[root@console postgresql]# cat postgresql_server_status.yml
- name: Check PostgreSQL server running status
  hosts: all
  tasks:
    - name: Check PostgreSQL service status
      systemd:
        name: postgresql-15
      register: postgresql_status

    - name: Display PostgreSQL service status
      debug:
        msg: "PostgreSQL service is {{ postgresql_status.status }}"


[root@console postgresql]# ansible-playbook postgresql_server_status.yml
```

11. **Manual Verification:**

Logged into the client machines and checked if it's installed correctly. There are more steps to perform, we just followed the basic steps over here.

```
[root@console postgresql]# ssh 192.168.31.55
Last login: Wed May 15 00:11:58 2024 from 192.168.31.88
[root@dba ~]# su - postgres
-bash-4.2$ psql
psql (15.7)
Type "help" for help.

postgres=# exit
-bash-4.2$ exit
logout
[root@dba ~]# exit
logout
Connection to 192.168.31.55 closed.


[root@console postgresql]# ssh 192.168.31.56
```

```
Last login: Wed May 15 00:11:58 2024 from 192.168.31.88
[root@dba ~]# su - postgres
-bash-4.2$ psql
psql (15.7)
Type "help" for help.

postgres=# exit
-bash-4.2$ exit
logout
[root@dba ~]#
```