**Pre – Patching & Post - Patching**

you prefer not to give your password directly in the playbook, you can use an app password or OAuth2 for better security. Gmail allows the use of app passwords for applications that don't support OAuth2, which provides an added layer of security.  
Using an App Password with Gmail

1. **Enable 2-Step Verification**: Ensure that 2-step verification is enabled for your Google account. You can enable it from
2. .
3. **Generate an App Password**:
   * Go to your Google Account settings.
   * Navigate to the "Security" section.
   * Under "Signing in to Google," select "App passwords."
   * Follow the prompts to generate an app password for your Ansible setup.
4. **Update the Playbook to Use the App Password**: Use the generated app password in your playbook instead of your regular password.

Updated Playbook  
yaml

---

- name: Pre-Patching, Patching, and Post-Patching Process

hosts: all

become: true

vars\_files:

- gmail\_pass.yml # Ensure this file contains the app password

tasks:

- name: Fetch the list of running services

command: systemctl list-units --type=service --state=running --no-pager --no-legend

register: running\_services

- name: Parse the running services

set\_fact:

service\_list: "{{ running\_services.stdout\_lines | map('regex\_replace', '\\.service.\*', '') | list }}"

- name: Debug output to check service list

debug:

var: service\_list

- name: Stop each running service

service:

name: "{{ item }}"

state: stopped

loop: "{{ service\_list }}"

ignore\_errors: yes # Add this if you want to proceed even if a service fails to stop

- name: Update all packages

dnf:

name: "\*"

state: latest

- name: Reboot the server

reboot:

reboot\_timeout: 600

msg: "Reboot initiated by Ansible"

- name: Start each previously running service

service:

name: "{{ item }}"

state: started

loop: "{{ service\_list }}"

ignore\_errors: yes # Add this if you want to proceed even if a service fails to start

- name: Send email report via Gmail

mail:

host: smtp.gmail.com

port: 587

username: your\_email@gmail.com

password: "{{ gmail\_password }}"

to: your\_team@example.com

subject: "Patching Report"

body: |

Services have been restarted successfully.

secure: starttls

Securing Your App Password:  
As mentioned earlier, it's a good practice to use Ansible Vault to encrypt your app password. Here's a quick rundown on how to do it:

1. **Create an Encrypted File for the App Password**:
2. bash
3. ansible-vault create gmail\_pass.yml
4. Add the following content to the gmail\_pass.yml file:
5. yaml
6. gmail\_password: your\_app\_password
7. **Decrypt the Vaulted File When Running the Playbook**:
8. bash
9. ansible-playbook -i your\_inventory\_file playbook.yml --ask-vault-pass

By using an app password, you maintain better security for your personal email a