To deploy the Grafana, Prometheus, Node Exporter, and Alert Manager stack using Ansible, follow these steps:

**1. Set Up Your Ansible Inventory**

Define the hosts where you want to deploy the stack in your hosts file:

ini

Copy code

[monitoring]

monitoring-server ansible\_host=<your\_server\_ip> ansible\_user=<your\_username>

**2. Create an Ansible Playbook**

Create an Ansible playbook named monitoring\_stack.yml:

yaml

Copy code

---

- name: Deploy Prometheus, Node Exporter, Alert Manager, and Grafana

hosts: monitoring

become: yes

vars:

prometheus\_version: "2.33.4"

node\_exporter\_version: "1.3.1"

alertmanager\_version: "0.24.0"

grafana\_version: "9.0.0"

tasks:

- name: Install dependencies

apt:

name:

- wget

- tar

- unzip

state: present

- name: Create directories for Prometheus, Node Exporter, Alert Manager, and Grafana

file:

path: "/opt/{{ item }}"

state: directory

mode: '0755'

loop:

- prometheus

- node\_exporter

- alertmanager

- grafana

- name: Download and extract Prometheus

ansible.builtin.get\_url:

url: "https://github.com/prometheus/prometheus/releases/download/v{{ prometheus\_version }}/prometheus-{{ prometheus\_version }}.linux-amd64.tar.gz"

dest: "/tmp/prometheus.tar.gz"

notify: extract\_prometheus

- name: Download and extract Node Exporter

ansible.builtin.get\_url:

url: "https://github.com/prometheus/node\_exporter/releases/download/v{{ node\_exporter\_version }}/node\_exporter-{{ node\_exporter\_version }}.linux-amd64.tar.gz"

dest: "/tmp/node\_exporter.tar.gz"

notify: extract\_node\_exporter

- name: Download and extract Alert Manager

ansible.builtin.get\_url:

url: "https://github.com/prometheus/alertmanager/releases/download/v{{ alertmanager\_version }}/alertmanager-{{ alertmanager\_version }}.linux-amd64.tar.gz"

dest: "/tmp/alertmanager.tar.gz"

notify: extract\_alertmanager

- name: Download and install Grafana

ansible.builtin.get\_url:

url: "https://dl.grafana.com/oss/release/grafana-{{ grafana\_version }}.linux-amd64.tar.gz"

dest: "/tmp/grafana.tar.gz"

notify: extract\_grafana

handlers:

- name: extract\_prometheus

unarchive:

src: "/tmp/prometheus.tar.gz"

dest: "/opt/prometheus"

remote\_src: yes

- name: extract\_node\_exporter

unarchive:

src: "/tmp/node\_exporter.tar.gz"

dest: "/opt/node\_exporter"

remote\_src: yes

- name: extract\_alertmanager

unarchive:

src: "/tmp/alertmanager.tar.gz"

dest: "/opt/alertmanager"

remote\_src: yes

- name: extract\_grafana

unarchive:

src: "/tmp/grafana.tar.gz"

dest: "/opt/grafana"

remote\_src: yes

- name: Configure Prometheus

copy:

content: |

global:

scrape\_interval: 15s

scrape\_configs:

- job\_name: 'prometheus'

static\_configs:

- targets: ['localhost:9090']

- job\_name: 'node\_exporter'

static\_configs:

- targets: ['localhost:9100']

rule\_files:

- /opt/prometheus/rules.yml

dest: /opt/prometheus/prometheus.yml

- name: Configure Alertmanager

copy:

content: |

global:

resolve\_timeout: 5m

route:

group\_by: ['alertname']

group\_wait: 30s

group\_interval: 5m

repeat\_interval: 12h

receiver: 'email'

receivers:

- name: 'email'

email\_configs:

- to: '<your\_email@example.com>'

from: 'alertmanager@example.com'

smarthost: 'smtp.example.com:587'

auth\_username: 'alertmanager@example.com'

auth\_password: 'password'

dest: /opt/alertmanager/alertmanager.yml

- name: Create systemd service for Prometheus

copy:

content: |

[Unit]

Description=Prometheus

Wants=network-online.target

After=network-online.target

[Service]

User=root

ExecStart=/opt/prometheus/prometheus --config.file=/opt/prometheus/prometheus.yml --storage.tsdb.path=/opt/prometheus/data

[Install]

WantedBy=multi-user.target

dest: /etc/systemd/system/prometheus.service

- name: Create systemd service for Node Exporter

copy:

content: |

[Unit]

Description=Node Exporter

Wants=network-online.target

After=network-online.target

[Service]

User=root

ExecStart=/opt/node\_exporter/node\_exporter

[Install]

WantedBy=multi-user.target

dest: /etc/systemd/system/node\_exporter.service

- name: Create systemd service for Alert Manager

copy:

content: |

[Unit]

Description=Alert Manager

Wants=network-online.target

After=network-online.target

[Service]

User=root

ExecStart=/opt/alertmanager/alertmanager --config.file=/opt/alertmanager/alertmanager.yml --storage.path=/opt/alertmanager/data

[Install]

WantedBy=multi-user.target

dest: /etc/systemd/system/alertmanager.service

- name: Create systemd service for Grafana

copy:

content: |

[Unit]

Description=Grafana

Wants=network-online.target

After=network-online.target

[Service]

User=root

ExecStart=/opt/grafana/bin/grafana-server --homepath=/opt/grafana --config=/opt/grafana/conf/defaults.ini

[Install]

WantedBy=multi-user.target

dest: /etc/systemd/system/grafana.service

- name: Reload systemd daemon

systemd:

daemon\_reload: yes

- name: Start and enable Prometheus

systemd:

name: prometheus

state: started

enabled: yes

- name: Start and enable Node Exporter

systemd:

name: node\_exporter

state: started

enabled: yes

- name: Start and enable Alert Manager

systemd:

name: alertmanager

state: started

enabled: yes

- name: Start and enable Grafana

systemd:

name: grafana

state: started

enabled: yes

**3. Run the Playbook**

Execute the playbook using the following command:

bash

Copy code

ansible-playbook -i hosts monitoring\_stack.yml

**Explanation:**

* **Tasks**:
  + Install necessary dependencies (e.g., wget, tar).
  + Download and extract the required binaries for Prometheus, Node Exporter, Alert Manager, and Grafana.
  + Create and configure systemd services for each component.
  + Copy configuration files to their respective directories.
* **Handlers**:
  + Extract the downloaded tar.gz files.

**4. Access the Services**

* **Prometheus**: http://<your\_server\_ip>:9090
* **Node Exporter**: http://<your\_server\_ip>:9100
* **Alertmanager**: http://<your\_server\_ip>:9093
* **Grafana**: http://<your\_server\_ip>:3000

This setup will allow you to monitor your infrastructure with Prometheus, visualize data with Grafana, and manage alerts with Alert Manager.