Install Grafana Prometheus Node_Exporter Using Ansible

In this tutorial we will write a playbook to install Prometheus , Grafana and node exporters in multiple ubuntu/Debian servers. to edit the code for a different server you just need to modify the package managers.

Step 1: Set up static inventory file and Ping the Target Nodes

- As a best practice, I prefer to create a new folder for each project and create a **config** file in it.
- before setting up the ansible playbook, make sure you have configured ssh connection to all your nodes with ssh-key.

> Ansible searches for the config file as shown below. that is it starts from the environment variable if any then to the current directory. since we will create our config file in the current directory, ansible will use the configurations in the current diretory.

```
ANSIBLE_CONFIG (environment variable if set)
ansible.cfg (in the current directory)
~/.ansible.cfg (in the home directory)
/etc/ansible/ansible.cfg
```

- create a directory named g*rafana-prometheus* in the home directory and navigate into it.

```
$ mkdir grafana-prometheus
$ cd grafana-prometheus
```

Create a file named `inventory.txt` in the home directory.

$ sudo vi inventory.txt

- Paste the content below into the **inventory.txt** file and edit to content to have your various servers..

```
[monitorserver]
monitor_server
[nodeservers]
server1
server2
```

> Note: don't forget to fill in the ip address of the various servers.

- Create file named `ansible.cfg` in the `grafana-prometheus` directory.
- Paste the content below into **ansible.cfg** file.

```
[defaults]
host_key_checking = False
inventory=inventory.txt
interpreter_python=auto_silent
localhost_warning=false
```

Validate and check the inventory.

```
ansible-inventory --graph
```

- Check the connectivity to the target nodes. to be sure you can ping all your servers.

$ ansible all -m ping

Step 2: Create a role to install node-exporter

As seen in the ansible tutorial, the ansible-galaxy command aids in the construction of a role's skeleton:

```
ansible-galaxy init roles/node-exporter
```

Then set some defaults variables in the default directory :

```
node_exporter_version: "1.1.2"
node_exporter_bin: /usr/local/bin/node_exporter
node_exporter_user: node-exporter
node_exporter_group: "{{ node_exporter_user }}"
node_exporter_dir_conf: /etc/node_exporter
```

In the main file of tasks directory paste the code below:

```
- name: check if node exporter exist
  stat:
    path: "{{ node_exporter_bin }}"
  register: __check_node_exporter_present
- name: create node exporter user
  user:
    name: "{{ node_exporter_user }}"
    append: true
    shell: /usr/sbin/nologin
    system: true
    create_home: false
- name: create node exporter config dir
  file:
    path: "{{ node_exporter_dir_conf }}"
    state: directory
    owner: "{{ node_exporter_user }}"
    group: "{{ node_exporter_group }}"
- name: if node exporter exist get version
  shell: "cat /etc/systemd/system/node_exporter.service | grep Version
| sed s/'.*Version '//g"
  when: __check_node_exporter_present.stat.exists == true
  changed_when: false
```

```
  register: __get_node_exporter_version

- name: download and unzip node exporter if not exist
  unarchive:
    src: "https://github.com/prometheus/node_exporter/releases/download
/v{{ node_exporter_version }}/node_exporter-{{ node_exporter_version }}.
linux-amd64.tar.gz"
    dest: /tmp/
    remote_src: yes
    validate_certs: no
- name: move the binary to the final destination
  copy:
    src: "/tmp/node_exporter-{{ node_exporter_version }}.linux-amd64
/node_exporter"
    dest: "{{ node_exporter_bin }}"
    owner: "{{ node_exporter_user }}"
    group: "{{ node_exporter_group }}"
    mode: 0755
    remote_src: yes
  when: __check_node_exporter_present.stat.exists == false or not
__get_node_exporter_version.stdout == node_exporter_version
- name: clean
  file:
    path: /tmp/node_exporter-{{ node_exporter_version }}.linux-amd64/
    state: absent
- name: install service
  template:
    src: node_exporter.service.j2
    dest: /etc/systemd/system/node_exporter.service
    owner: root
    group: root
    mode: 0755
  notify: reload_daemon_and_restart_node_exporter
- meta: flush_handlers
- name: service always started
  systemd:
    name: node_exporter
    state: started
    enabled: yes
```

We need to create node_exorter.service.j2 file in templates directory and paste the code below in it :

```
[Unit]
Description=Node Exporter Version {{ node_exporter_version }}
After=network-online.target
[Service]
User={{ node_exporter_user }}
Group={{ node_exporter_user }}
Type=simple
ExecStart={{ node_exporter_bin }}
[Install]
WantedBy=multi-user.target
```

Finally the handler file in the handler directory :

```
- name: reload_daemon_and_restart_node_exporter
  systemd:
    name: node_exporter
    state: restarted
    daemon_reload: yes
    enabled: yes
```

Step 3: Create a role for Prometheus and its configuration

Initialize a Prometheus role with the ansible-galaxy command :

```
ansible-galaxy init roles/prometheus
```

Now set some defaults variables in the default directory :

```
prometheus_dir_configuration: "/etc/prometheus"
prometheus_retention_time: "365d"
prometheus_scrape_interval: "30s"
prometheus_node_exporter: true
prometheus_node_exporter_group: "all"
prometheus_env: "production"
prometheus_var_config:
  global:
    scrape_interval: "{{ prometheus_scrape_interval }}"
    evaluation_interval: 5s
    external_labels:
      env: '{{ prometheus_env }}'
  scrape_configs:
    - job_name: prometheus
      scrape_interval: 5m
      static_configs:
        - targets: ['{{ inventory_hostname }}:9090']
```

Now lets create the prometheus tasks in the main.yml file of tasks directory :

```
- name: update and install prometheus
  apt:
    name: prometheus
    state: latest
    update_cache: yes
    cache_valid_time: 3600
- name: prometheus args
  template:
    src: prometheus.j2
    dest: /etc/default/prometheus
    mode: 0644
    owner: root
    group: root
  notify: restart_prometheus
- name: prometheus configuration file
  template:
    src: prometheus.yml.j2
    dest: "{{ prometheus_dir_configuration }}/prometheus.yml"
    mode: 0755
    owner: prometheus
    group: prometheus
  notify: reload_prometheus
- name: start prometheus
  systemd:
    name: prometheus
    state: started
    enabled: yes
```

Create the prometheus.yaml.j2 file in the template directory and paste the code below in it.

```
#jinja2: lstrip_blocks: "True"
{{ prometheus_var_config | to_nice_yaml(indent=2) }}
{% if prometheus_node_exporter_group %}
- job_name: node
  scrape_interval: 15s
  metrics_path: /metrics
  static_configs:
  - targets:
{% for server in groups[prometheus_node_exporter_group] %}
    - '{{ server }}:9100'
{% endfor %}
{% endif %}
```

And the prometheus.j2 file for the Prometheus CLI :

```
ARGS="--web.enable-lifecycle --storage.tsdb.retention.time={{
prometheus_retention_time }} --web.console.templates=/etc/prometheus
/consoles --web.console.libraries=/etc/prometheus/console_libraries
```

Finally handlers of this Prometheus role in the handlers directory; We have two handlers :

- Restart with the systemd service
- And to reload the curl services.

```
- name: restart_prometheus
  systemd:
    name: prometheus
    state: restarted
    enabled: yes
    daemon_reload: yes
- name: reload_prometheus
  uri:
    url: http://localhost:9090/-/reload
    method: POST
    status_code: 200
```

Step 4: Create a role for Grafana

Now we can create a last role to install grafana-server package and start it.

`ansible-galaxy init roles/grafana`

Just edit the main.yml file in the tasks directory and paste the code below in it :

```
- name: install gpg
  apt:
    name: gnupg,software-properties-common
    state: present
    update_cache: yes
    cache_valid_time: 3600
- name: add gpg hey
  apt_key:
    url: "https://packages.grafana.com/gpg.key"
    validate_certs: no
- name: add repository
  apt_repository:
    repo: "deb https://packages.grafana.com/oss/deb stable
main"
    state: present
    validate_certs: no
- name: install grafana
  apt:
    name: grafana
    state: latest
    update_cache: yes
    cache_valid_time: 3600
- name: start service grafana-server
  systemd:
    name: grafana-server
    state: started
    enabled: yes
- name: wait for service up
  uri:
    url: "http://127.0.0.1:3000"
    status_code: 200
  register: __result
  until: __result.status == 200
  retries: 120
  delay: 1
- name: change admin password for grafana gui
  shell : "grafana-cli admin reset-admin-password {{
grafana_admin_password }}"
  register: __command_admin
  changed_when: __command_admin.rc !=0
```

Don't forget to set your Admin password for start and you can set it in default directory:

```
grafana_admin_password: "admin"
```

Step 5: Create Ansible Playbook

Create an Ansible playbook file like below in the prometheus-grafana directory :

```
- name: install monitoring stack
  hosts: monitorserver
  become: yes
  roles:
  - prometheus
  - grafana
- name: install node-exporter
  hosts: nodeservers
  become: yes
  roles:
  - node-exporter
```

Step 6: Testing

You can run this playbook file via below command:

```
ansible-playbook -i inventory.txt playbook.yml
```

you just created a playbook to install Prometheus and grafana in the monitoring server and node_exporter in the servers to be monitored. In other to install just node exporters you can comment out the section installing Prometheus and grafana on the playbook.

```
name: install node-exporter
  hosts: nodeservers
  become: yes
  roles:
  - node-exporter
```

feel free to practice more and ask questions in case of any difficulty.