**Ansible & promethus Grafana Questions**

1. What is the use of ansible?
2. Have you written any playbooks in ansible?
3. Explain about inventory file?
4. Explain about handlers?
5. How did you troubleshoot your playbook?
6. What is difference between ansible, chef , puppet?
7. How did you configure promethus and Grafana for monitoring Kubernetes cluster?
8. Explain promethus?
9. How did you create dashboard in Grafana?

**10.**How did you configure alert manager in Grafana?

**11.**Have you written and Queries in promql?

Here are detailed answers to your Ansible, Prometheus, and Grafana interview questions:

**1. What is the use of Ansible?**

**Ansible** is an open-source automation tool used for:

* **Configuration Management**: Automating system configurations across multiple servers.
* **Application Deployment**: Deploying applications without manual intervention.
* **Orchestration**: Managing services and ensuring smooth interoperation.
* **Provisioning**: Setting up servers in a defined state.
* **Security & Compliance**: Enforcing security policies across systems.

It is agentless and uses SSH to communicate with remote systems.

**2. Have you written any playbooks in Ansible?**

Yes! An **Ansible Playbook** is a YAML-based script used to define automation tasks.

Example of a playbook to install Apache:

- name: Install Apache

hosts: webservers

become: yes

tasks:

- name: Install Apache package

apt:

name: apache2

state: present

- name: Start Apache service

service:

name: apache2

state: started

This playbook:

* Targets the webservers group.
* Installs apache2.
* Ensures it is running.

**3. Explain about the inventory file?**

An **inventory file** lists managed nodes (hosts) and groups them.

Example:

[webservers]

192.168.1.10

192.168.1.11

[dbservers]

192.168.1.20 ansible\_user=root ansible\_ssh\_private\_key\_file=/path/to/key

Types of inventories:

* **Static Inventory** (as above)
* **Dynamic Inventory** (retrieved from cloud providers)

**4. Explain about handlers?**

**Handlers** are special tasks that only run when notified.

Example:

- name: Install Nginx

hosts: webservers

tasks:

- name: Install Nginx

apt:

name: nginx

state: present

notify: Restart Nginx

handlers:

- name: Restart Nginx

service:

name: nginx

state: restarted

Here, Restart Nginx runs **only if** the Install Nginx task makes changes.

**5. How did you troubleshoot your playbook?**

* **Syntax check**:
* ansible-playbook myplaybook.yml --syntax-check
* **Dry run**:
* ansible-playbook myplaybook.yml --check
* **Verbose mode**:
* ansible-playbook myplaybook.yml -vvv
* **Debugging inside tasks**:
* - debug:
* msg: "Variable value is {{ my\_variable }}"

**6. What is the difference between Ansible, Chef, and Puppet?**

| **Feature** | **Ansible** | **Chef** | **Puppet** |
| --- | --- | --- | --- |
| Agentless | ✅ | ❌ | ❌ |
| Language | YAML | Ruby | Puppet DSL |
| Ease of Use | Easy | Medium | Complex |
| Architecture | Push-based | Pull-based | Pull-based |

Ansible is simpler and does not require agents, making it preferable for quick automation.

**7. How did you configure Prometheus and Grafana for monitoring a Kubernetes cluster?**

1. **Deploy Prometheus**:
   * Use Helm:
   * helm install prometheus prometheus-community/kube-prometheus-stack
   * This installs Prometheus, Grafana, and Alertmanager.
2. **Deploy Grafana**:
   * Access Grafana UI:
   * kubectl port-forward svc/prometheus-grafana 3000:80
   * Login (admin/admin) and configure **Prometheus** as a data source.

**8. Explain Prometheus?**

Prometheus is an open-source **monitoring and alerting tool** that:

* Collects time-series data using **pull-based** scraping.
* Uses **PromQL** for querying data.
* Supports **service discovery** for dynamic environments.

It consists of:

* **Prometheus Server** (collects and stores data).
* **Exporters** (gather system metrics).
* **Alertmanager** (handles alerts).

**9. How did you create a dashboard in Grafana?**

1. **Add Data Source**:
   * In Grafana, go to **Settings > Data Sources**.
   * Select **Prometheus** and enter the URL (http://prometheus:9090).
2. **Create Dashboard**:
   * Click **Create > Dashboard**.
   * Add a **panel** and use PromQL queries (e.g., rate(node\_cpu\_seconds\_total[5m])).

**10. How did you configure Alertmanager in Grafana?**

* Modify **alertmanager.yml**:
* route:
* receiver: 'email-alert'
* receivers:
* - name: 'email-alert'
* email\_configs:
* - to: 'admin@example.com'
* from: 'alert@example.com'
* smarthost: 'smtp.example.com:587'
* Restart Prometheus:
* systemctl restart prometheus
* Set alerts in **Grafana > Alerting**.

**11. Have you written any queries in PromQL?**

Yes! Some useful **PromQL** queries:

1. **CPU Usage**:
2. rate(node\_cpu\_seconds\_total{mode="user"}[5m])
3. **Memory Usage**:
4. node\_memory\_Active\_bytes / node\_memory\_MemTotal\_bytes \* 100
5. **Pod Restarts**:
6. increase(kube\_pod\_container\_status\_restarts\_total[1h])