TOPOLOGÍA DE LA SOLUCIÓN

1. **Implementación de Instancias Docker y Configuración de Conectividad y Seguridad con Ansible**

   Implementa dos instancias, un "Bastión" y un "Servidor de Monitoreo", implementa servicios bajo contenedores con Docker.

```yaml
- name: Configuración del Bastión y Servidor de Monitoreo
  hosts: all
  become: true
  tasks:
    - name: Configura hostname
      hostname:
        name: "{{ inventory_hostname }}"

    - name: Añade el usuario 'admin' al 'Servidor de Monitoreo'
      user:
        name: admin
        group: sudo
        createhome: yes
        state: present
      when: "'monitoring_server' in group_names"

    - name: Añade el usuario 'operator' al 'Servidor de Monitoreo' sin privilegios d
      user:
        name: operator
        createhome: yes
        state: present
      when: "'monitoring_server' in group_names"

    - name: Asegura que el usuario 'root' esté deshabilitado en el 'Servidor de Moni
      user:
        name: root
        password_lock: yes
      when: "'monitoring_server' in group_names"

    - name: Configuración de interfaces de red y rutas
      command: echo "Este paso depende de tu configuración específica"

    - name: Instala el paquete NTP
      package:
        name: ntp
        state: present

    - name: Asegura que el servicio NTP esté activo y habilitado
      service:
        name: ntp
        state: started
        enabled: yes
```

El provisionamiento de hostname, usuarios, configuracion de redes, rutas, ntp debe realizarse utilizando ansible playbooks.

Ambas instancias deben tener las siguientes características:

Bastión:

3 interfaces de red: oym, servicio, backup.

Accesible únicamente por el puerto 22 en la interfaz oym utilizando private keys.

```yaml
- name: Limita acceso SSH solo por la interfaz 'oym' en el Bastión
  iptables:
    chain: INPUT
    in_interface: oym
    protocol: tcp
    destination_port: "22"
    jump: ACCEPT
  when: inventory_hostname == 'bastion'

- name: Deniega todo el acceso SSH que no sea por la interfaz 'oym' en el Bastión
  iptables:
    chain: INPUT
    protocol: tcp
    destination_port: "22"
    jump: DROP
  when: inventory_hostname == 'bastion'

- name: Permite comunicación específica entre Bastión y Servidor de Monitoreo en la interfaz 'servicio'
  command: echo "Configura tu regla de firewall aquí"
  when: "'monitoring_server' in group_names or inventory_hostname == 'bastion'"

- name: Configura el acceso al puerto 5040/TCP solo en la interfaz 'backup'
  iptables:
    chain: INPUT
    in_interface: backup
    protocol: tcp
    destination_port: "5040"
    jump: ACCEPT
  when: "'monitoring_server' in group_names or inventory_hostname == 'bastion'"
```

Servidor de Monitoreo:
3 interfaces de red: oym, servicio, backup.
Accesible únicamente por el “Bastion” via la interfaz oym utilizando private keys definiendo los siguientes usuarios administrator con full privilegios, operator sin privilegios, y denegado de root en “Servidor de monitoreo”.
Implementa Grafana y Prometheus.
Adiciona el "Bastión" como recurso a ser monitoreado

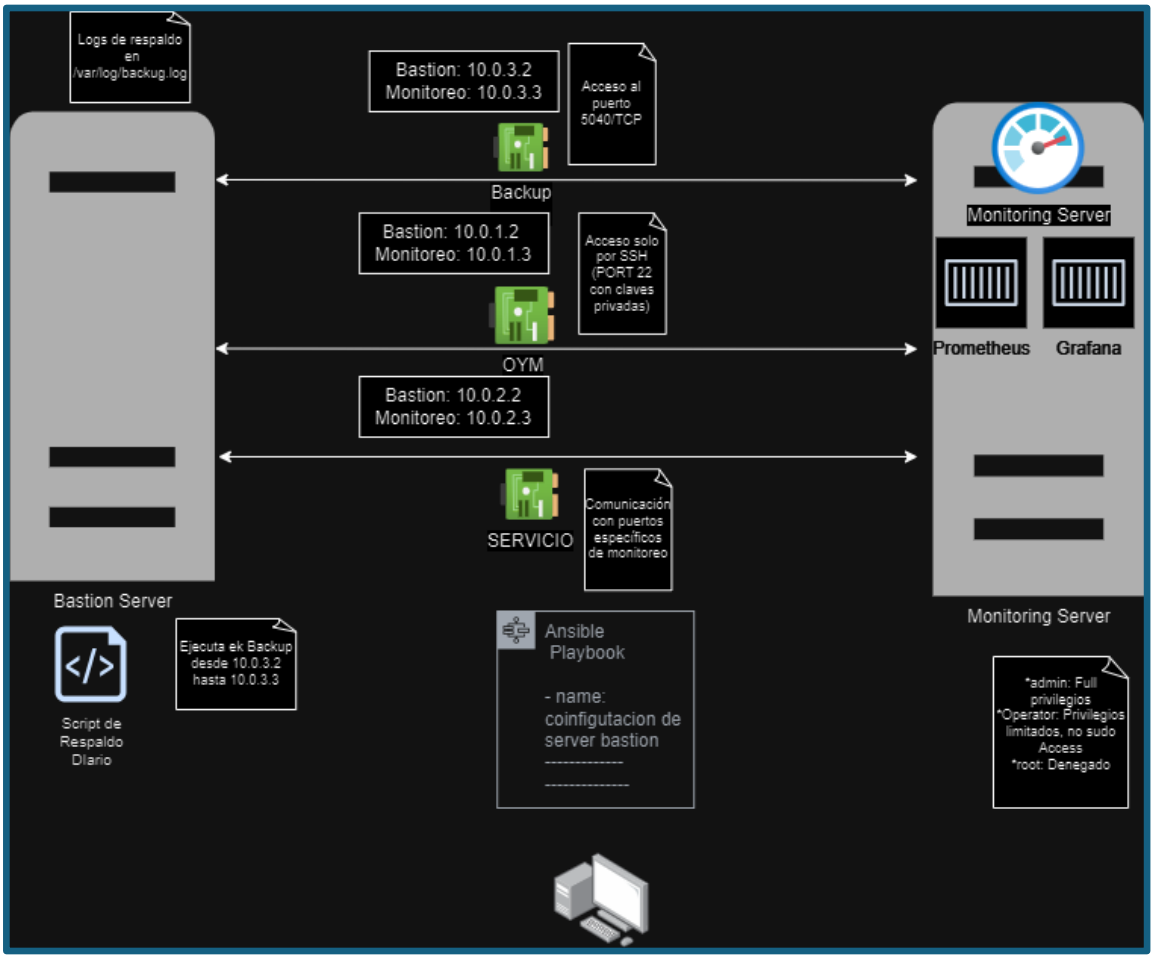
```yaml
- name: Configuración de Prometheus
  hosts: monitoring_server
  become: true
  tasks:
    - name: Crea el directorio de configuración de Prometheus
      ansible.builtin.file:
        path: /prometheus/config
        state: directory
        mode: '0755'

    - name: Copia el archivo de configuración de Prometheus
      ansible.builtin.copy:
        src: /home/christian/prometheus/config/prometheus.yml
        dest: /prometheus/config/prometheus.yml
        mode: '0644'
        remote_src: yes
```

```yaml
- name: Despliega Grafana y Prometheus en Docker en el Servidor de Monitoreo
  hosts: monitoring_server
  become: true
  tasks:
    - name: Elimina el contenedor de Prometheus si ya existe usando comando directo
      command: docker rm -f prometheus
      ignore_errors: yes

    - name: "Asegura que el contenedor de Prometheus esté corriendo"
      community.docker.docker_container:
        name: prometheus
        image: "prom/prometheus:v2.22.0"
        state: started
        restart_policy: always
        ports:
          - "9090:9090"
        volumes:
          - "/prometheus/config/prometheus.yml:/etc/prometheus/prometheus.yml"
        recreate: yes
        force_kill: true
        pull: yes

    - name: "Elimina el contenedor de Grafana si ya existe"
      community.docker.docker_container:
        name: grafana
        state: absent
      ignore_errors: yes

    - name: "Asegura que el contenedor de Grafana esté corriendo"
      community.docker.docker_container:
        name: grafana
        image: grafana/grafana
        ports:
          - "3000:3000"
        volumes:
          - "/home/christian/grafana/data:/var/lib/grafana"
        restart_policy: unless-stopped
        state: started
        recreate: yes
      ignore_errors: yes
```

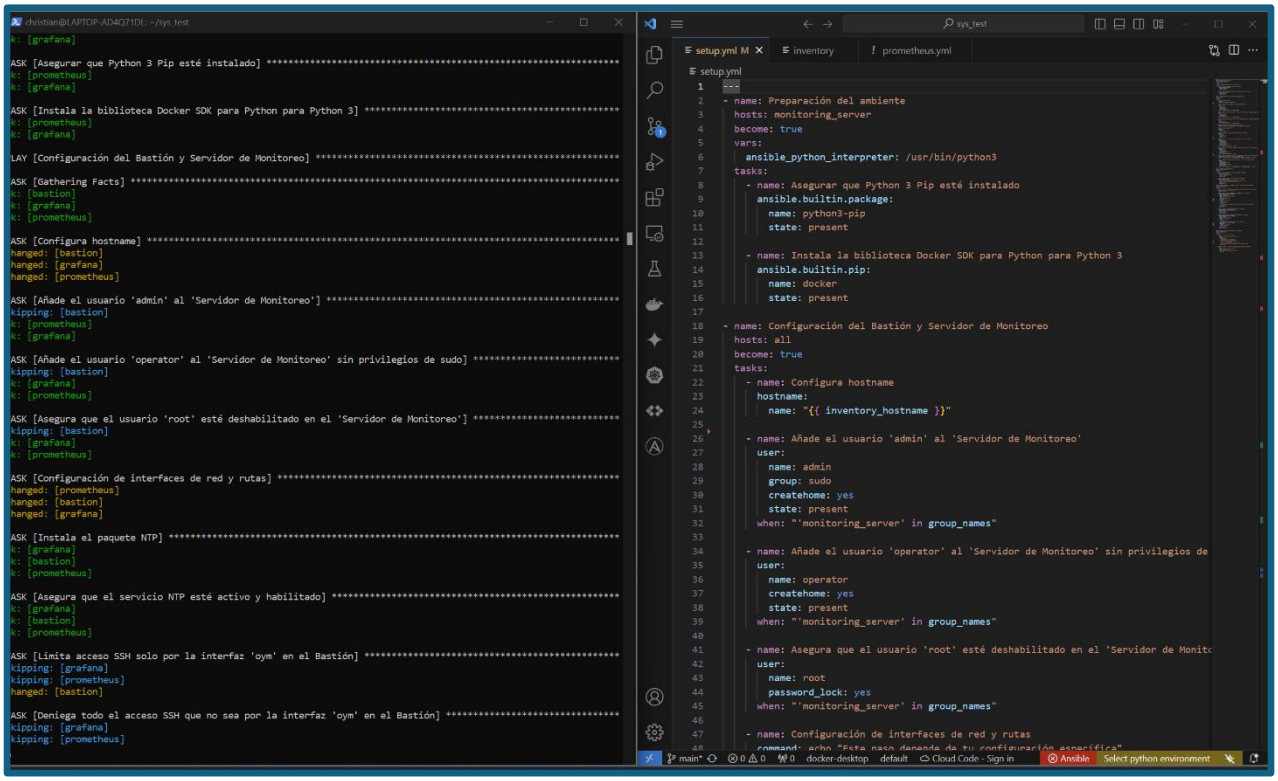
2. Implementación de Respaldos con Script y log

```yaml
- name: Configura el script de respaldo en el Bastión
  hosts: bastion_hosts
  become: true
  tasks:
    - name: Crea el script de respaldo
      copy:
        dest: "/usr/local/bin/backup_script.sh"
        content: |
          #!/bin/bash
          DATE=$(date +%Y-%m-%d-%H%M%S)
          BACKUP_DIR="/var/log/backup/$DATE"
          mkdir -p "$BACKUP_DIR"
          rsync -avz /var/log/data "$BACKUP_DIR"
          echo "Backup realizado el $DATE" >> /var/log/backup.log
        mode: '0755'

    - name: Programa el script de respaldo para ejecutarse diariamente
      cron:
        name: "Tarea de respaldo diario"
        special_time: daily
        job: "/usr/local/bin/backup_script.sh"
        user: root
```

3. Topología de la Solución



Ejecución del Playbook:



Monitoreo de la Instancia Bastion con Prometheus y Grafana: