

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.

```
18 - name: Configuración del Bastión y Servidor de Monitoreo
19   hosts: all
20   become: true
21   tasks:
22     - name: Configura hostname
23       hostname:
24         name: "{{ inventory_hostname }}"
25
26     - name: Añade el usuario 'admin' al 'Servidor de Monitoreo'
27       user:
28         name: admin
29         group: sudo
30         createhome: yes
31         state: present
32       when: "'monitoring_server' in group_names"
33
34     - name: Añade el usuario 'operator' al 'Servidor de Monitoreo' sin privilegios de
35       user:
36         name: operator
37         createhome: yes
38         state: present
39       when: "'monitoring_server' in group_names"
40
41     - name: Asegura que el usuario 'root' esté deshabilitado en el 'Servidor de Moni
42       user:
43         name: root
44         password_lock: yes
45       when: "'monitoring_server' in group_names"
46
47     - name: Configuración de interfaces de red y rutas
48       command: echo "Este paso depende de tu configuración específica"
49
50     - name: Instala el paquete NTP
51       package:
52         name: ntp
53         state: present
54
55     - name: Asegura que el servicio NTP esté activo y habilitado
56       service:
57         name: ntp
58         state: started
59         enabled: yes
```

El provisionamiento de hostname, usuarios, configuración 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.

```

61 - name: Limita acceso SSH solo por la interfaz 'oym' en el Bastión
62   iptables:
63     chain: INPUT
64     in_interface: oym
65     protocol: tcp
66     destination_port: "22"
67     jump: ACCEPT
68     when: inventory_hostname == 'bastion'
69
70 - name: Deniega todo el acceso SSH que no sea por la interfaz 'oym' en el Bastión
71   iptables:
72     chain: INPUT
73     protocol: tcp
74     destination_port: "22"
75     jump: DROP
76     when: inventory_hostname == 'bastion'
77
78 - name: Permite comunicación específica entre Bastión y Servidor de Monitoreo en la interfaz 'servicio'
79   command: echo "Configura tu regla de firewall aquí"
80   when: "'monitoring_server' in group_names or inventory_hostname == 'bastion'"
81
82 - name: Configura el acceso al puerto 5040/TCP solo en la interfaz 'backup'
83   iptables:
84     chain: INPUT
85     in_interface: backup
86     protocol: tcp
87     destination_port: "5040"
88     jump: ACCEPT
89     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 administrador 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

```

91 - name: Configuración de Prometheus
92   hosts: monitoring_server
93   become: true
94   tasks:
95     - name: Crea el directorio de configuración de Prometheus
96       ansible.builtin.file:
97         path: /prometheus/config
98         state: directory
99         mode: '0755'
100
101     - name: Copia el archivo de configuración de Prometheus
102       ansible.builtin.copy:
103         src: /home/christian/prometheus/config/prometheus.yml
104         dest: /prometheus/config/prometheus.yml
105         mode: '0644'
106         remote_src: yes

```

```

108 - name: Despliega Grafana y Prometheus en Docker en el Servidor de Monitoreo
109   hosts: monitoring_server
110   become: true
111   tasks:
112     - name: Elimina el contenedor de Prometheus si ya existe usando comando directo
113       command: docker rm -f prometheus
114       ignore_errors: yes
115
116     - name: "Asegura que el contenedor de Prometheus esté corriendo"
117       community.docker.docker_container:
118         name: prometheus
119         image: "prom/prometheus:v2.22.0"
120         state: started
121         restart_policy: always
122         ports:
123           - "9090:9090"
124         volumes:
125           - "/prometheus/config/prometheus.yml:/etc/prometheus/prometheus.yml"
126         recreate: yes
127         force_kill: true
128         pull: yes
129
130     - name: "Elimina el contenedor de Grafana si ya existe"
131       community.docker.docker_container:
132         name: grafana
133         state: absent
134       ignore_errors: yes
135
136     - name: "Asegura que el contenedor de Grafana esté corriendo"
137       community.docker.docker_container:
138         name: grafana
139         image: grafana/grafana
140         ports:
141           - "3000:3000"
142         volumes:
143           - "/home/christian/grafana/data:/var/lib/grafana"
144         restart_policy: unless-stopped
145         state: started
146         recreate: yes
147       ignore_errors: yes

```

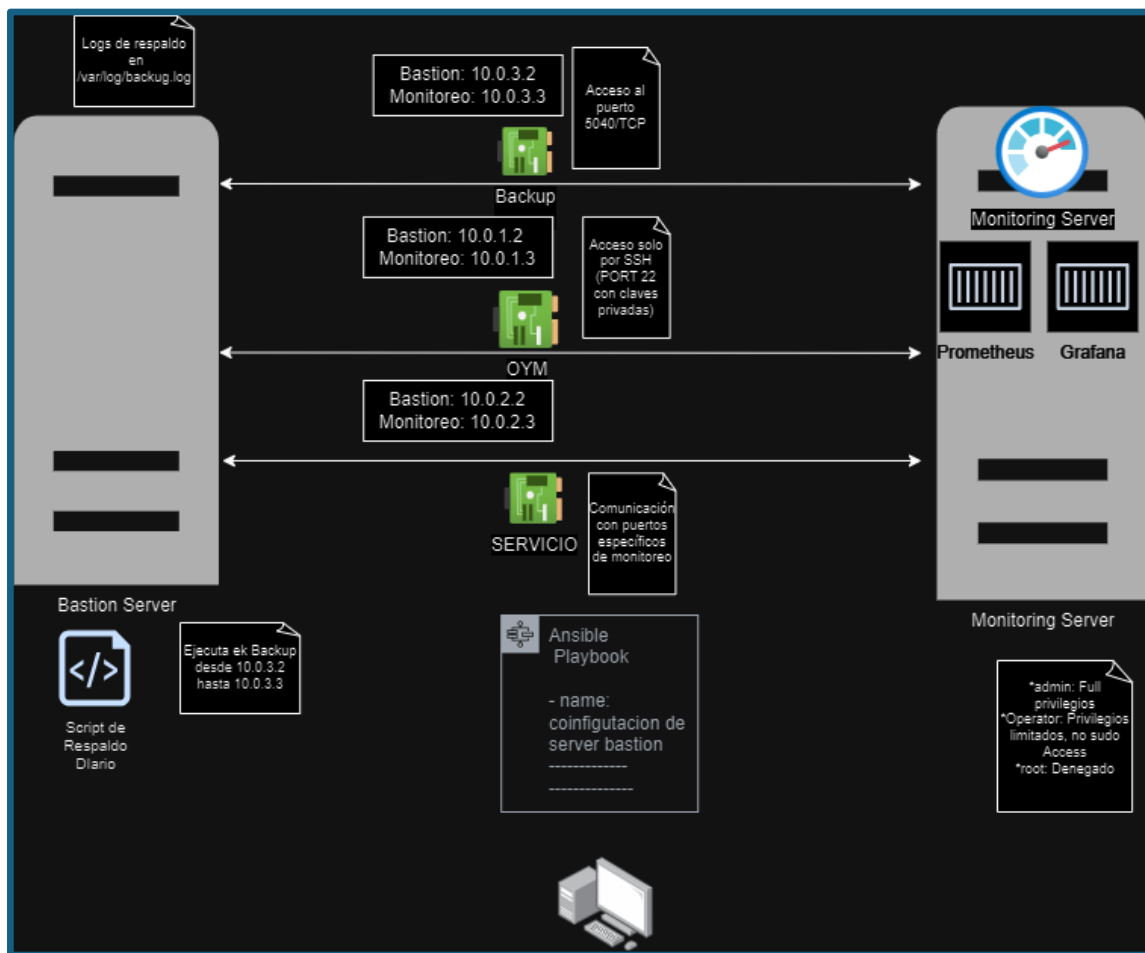
2. Implementación de Respallos con Script y log

```

150 - name: Configura el script de respaldo en el Bastión
151   hosts: bastion_hosts
152   become: true
153   tasks:
154     - name: Crea el script de respaldo
155       copy:
156         dest: "/usr/local/bin/backup_script.sh"
157         content: |
158           #!/bin/bash
159           DATE=$(date +%Y-%m-%d-%H%M%S)
160           BACKUP_DIR="/var/log/backup/$DATE"
161           mkdir -p "$BACKUP_DIR"
162           rsync -avz /var/log/data "$BACKUP_DIR"
163           echo "Backup realizado el $DATE" >> /var/log/backup.log
164         mode: '0755'
165
166     - name: Programa el script de respaldo para ejecutarse diariamente
167       cron:
168         name: "Tarea de respaldo diario"
169         special_time: daily
170         job: "/usr/local/bin/backup_script.sh"
171         user: root

```

3. Topología de la Solución



Ejecución del Playbook:

```
christian@LAPTOP-AD4Q71DL: ~/sys_test
k: [grafana]
ASK [Asegurar que Python 3 Pip esté instalado] *****
k: [prometheus]
k: [grafana]
ASK [Instala la biblioteca Docker SDK para Python para Python 3] *****
k: [prometheus]
k: [grafana]
LAY [Configuración del Bastión y Servidor de Monitoreo] *****
ASK [Gathering Facts] *****
k: [bastion]
k: [grafana]
k: [prometheus]
ASK [Configura hostname] *****
k: [bastion]
k: [grafana]
k: [prometheus]
ASK [Añade el usuario 'admin' al 'Servidor de Monitoreo'] *****
kipping: [bastion]
k: [prometheus]
k: [grafana]
ASK [Añade el usuario 'operator' al 'Servidor de Monitoreo' sin privilegios de sudo] *****
kipping: [bastion]
k: [grafana]
k: [prometheus]
ASK [Asegura que el usuario 'root' esté deshabilitado en el 'Servidor de Monitoreo'] *****
kipping: [bastion]
k: [grafana]
k: [prometheus]
ASK [Configuración de interfaces de red y rutas] *****
hanged: [prometheus]
hanged: [bastion]
hanged: [grafana]
ASK [Instala el paquete NTP] *****
k: [grafana]
k: [bastion]
k: [prometheus]
ASK [Asegura que el servicio NTP esté activo y habilitado] *****
k: [grafana]
k: [bastion]
k: [prometheus]
ASK [Limita acceso SSH solo por la interfaz 'oym' en el Bastión] *****
kipping: [grafana]
kipping: [prometheus]
hanged: [bastion]
ASK [Deniega todo el acceso SSH que no sea por la interfaz 'oym' en el Bastión] *****
kipping: [grafana]
kipping: [prometheus]
```

```
1
2
3 - name: Preparación del ambiente
4   hosts: monitoring_server
5   become: true
6   vars:
7     ansible_python_interpreter: /usr/bin/python3
8   tasks:
9     - name: Asegurar que Python 3 Pip esté instalado
10       ansible.builtin.package:
11         name: python3-pip
12         state: present
13
14     - name: Instala la biblioteca Docker SDK para Python para Python 3
15       ansible.builtin.pip:
16         name: docker
17         state: present
18
19 - name: Configuración del Bastión y Servidor de Monitoreo
20   hosts: all
21   become: true
22   tasks:
23     - name: Configura hostname
24       hostname:
25         name: "{{ inventory_hostname }}"
26
27     - name: Añade el usuario 'admin' al 'Servidor de Monitoreo'
28       user:
29         name: admin
30         group: sudo
31         createhome: yes
32         state: present
33         when: "'monitoring_server' in group_names"
34
35     - name: Añade el usuario 'operator' al 'Servidor de Monitoreo' sin privilegios de sudo
36       user:
37         name: operator
38         createhome: yes
39         state: present
40         when: "'monitoring_server' in group_names"
41
42     - name: Asegura que el usuario 'root' esté deshabilitado en el 'Servidor de Monit
43       user:
44         name: root
45         password_lock: yes
46         when: "'monitoring_server' in group_names"
47
48     - name: Configuración de interfaces de red y rutas
49       command: echo "Esta es una demanda de configuración específica"
```

Monitoreo de la Instancia Bastion con Prometheus y Grafana:

← → ↺ ⓘ localhost:9090/targets

Prometheus Alerts Graph Status ▾ Help

Targets

All Unhealthy Collapse All

bastion (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://172.23.59.3:9100/metrics	UP	instance="172.23.59.3:9100" job="bastion"	11.592s ago	11.22ms	

prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://host.docker.internal:9090/metrics	UP	instance="host.docker.internal:9090" job="prometheus"	10.092s ago	5.365ms	

