

Job Interview Exercise - DevOps Lab



Scopo del documento

Descrivere gli step volti all'implementazione di un esercizio di laboratorio propedeutico al colloquio tecnico con Sourcesense per l'area CNI (Cloud Native Infrastructure).

Tecnologie a perimetro



Step

1. Creazione di un repository in GitHub denominato "**sou-lab-cni**".
2. Creazione di un progetto Vagrant multinodo (2 nodi: soufe1 e soube2) con comunicazione su interfacce network locali (scegliere una subnet a piacere in 192.168.0.0/16).
3. Scegliere la distribuzione tra Centos, Almalinux, RockyLinux e Oracle Linux.
4. Creazione di un ruolo Ansible denominato **sou_podman**. Nei punti successivi sono espresse le azione che deve compiere l'automatismo.
5. Installare podman su soufe1 e soube2.
6. Installare HaProxy su soufe1 tramite podman e utilizzare un template Ansible per haproxy.cfg.
7. Installare Prometheus e Grafana tramite podman su soube2. Utilizzare volumi ad-hoc per dati e file di configurazione. I file di configurazione grafana.ini e prometheus.yml occorre che siano gestiti tramite template Ansible.
8. Poter raggiungere Grafana e Prometheus via http o https (bonus per https con creazione certificati anche self-signed).
9. HaProxy deve quindi operare come Reverse proxy per Prometheus e Grafana.

10. Versionare tutto in sou-lab-cni => Progetto Vagrant, Ansible Role, Templates e documentazione in Markdown su README.md. Il progetto deve automatizzare tutti i punti all'esecuzione di "vagrant up o vagrant provision".

Consigli per Ansible e Podman

```
- name: Create data dir
  ansible.builtin.file:
    path: /containers_vols
    state: directory
    mode: '0755'

- name: Create Prometheus data dir
  ansible.builtin.file:
    path: /containers_vols/prometheus/data
    state: directory
    owner: "{{ prometheus_id }}"
    group: "{{ prometheus_id }}"
    mode: '0755'

- name: Create Prometheus Container
  containers.podman.podman_container:
    name: prometheus
    image: prom/prometheus:{{ prometheus_tag }}
    volume:
      -
/containers_vols/prometheus/config/prometheus.yml:/etc/prometheus/prometheus.
yml:z
    - /containers_vols/prometheus/data:/prometheus:z
    expose:
      - 9090
    ports:
      - "9090:9090"
```

Consigli per HaProxy

```
global
  maxconn 20000
  ulimit-n 16384
  log 127.0.0.1 local0
  daemon

frontend fe_https
  bind *:8443 ssl crt /usr/local/etc/haproxy/certs/foo.pem
  mode http
  log global
  option httplog
  option dontlognull
  option nolinger
  maxconn 8000
  timeout client 30s
```

```

acl is_grafana hdr(host) -i foobar.local
use_backend grafana if is_grafana

acl is_prometheus hdr(host) -i foo.local
use_backend prometheus if is_prometheus

backend vault
  mode http
  balance roundrobin
  option httpchk
  server vault1 192.168.50.111:8200 check check-ssl ssl verify none

backend grafana
  mode http
  balance roundrobin
  option httpchk
  server grafana1 192.168.50.111:3000 check

backend prometheus
  mode http
  balance roundrobin
  option httpchk
  server prometheus1 192.168.50.111:9090 check

```

Consigli per Vagrant

```

Vagrant.configure("2") do |config|

  config.vm.box = "generic/oracle8"
  config.vbguest.auto_update = false

  config.vm.network "private_network", ip: "192.168.50.111"

  config.vm.provision "ansible" do |ansible|
    ansible.playbook = "../deploy.yml"
    ansible.become = true
  end

  config.vm.provider "virtualbox" do |v|
    v.memory = 2048
    v.cpus = 1
  end

end

```