VIRTUALIZATION AND CLOUD COMPUTING

FATEMEH ZAHRA MORADI BALEF
PEYMAN ATAEI

overall structure

vmware mechines

storage 10.255.255.0/24

vm1

192.168.58.0/24 dev ens37 proto kernel scope link src 192.168.58.11

192.168.255.0/24 dev ens33 proto kernel scope link src 192.168.255.11

vm2

192.168.58.0/24 dev ens37 proto kernel scope link src 192.168.58.12

192.168.255.0/24 dev ens33 proto kernel scope link src 192.168.255.12

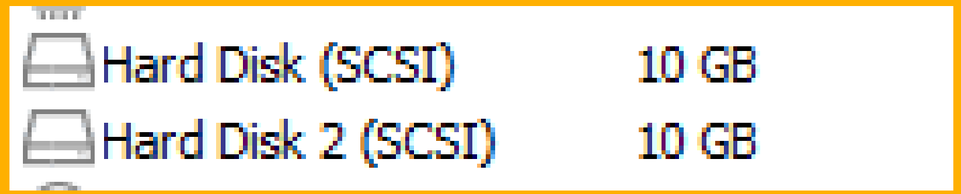
vm3

192.168.58.0/24 dev ens37 proto kernel scope link src 192.168.58.13

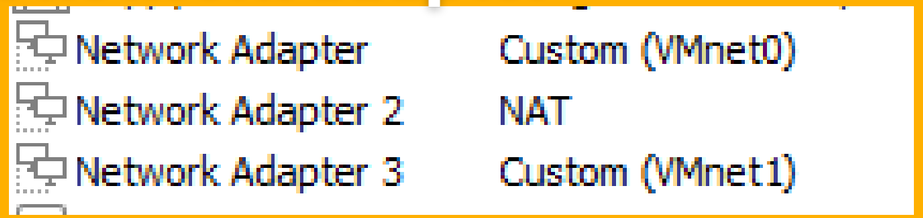
192.168.255.0/24 dev ens33 proto kernel scope link src 192.168.255.13

virtual mechines settings

2 hard disks



3 network adaptor



MY HOSTS

```
[webservers]
ansible_hostname="vm1"    ansible_host="192.168.58.11" swarm_master="True"
ansible_hostname="vm2"    ansible_host="192.168.58.12"
ansible_hostname="vm3"    ansible_host="192.168.58.13"

[all:vars]
ansible_connection=ssh
ansible_user=fatemeh
ansible_ssh_pass=123
ansible_become_pass=123
```

Ansible playbook

Ansible playbooks are lists of tasks that automatically execute against hosts

Ansible offer a repeatable, re-usable, simple configuration management and multi-machine deployment system, one that is well suited to deploying complex applications

My playbook 1

netplan configeration

```
netplan_configuration:
  network:
    version: 2
    ethernets:
      ens37:
        addresses:
          - "{{ ansible_host }}/24"
        gateway4: 192.168.58.2
        nameservers:
          addresses:
            - 10.0.2.3
            - 8.8.8.8
            - 8.8.4.4
      ens33:
        addresses:
          - "192.168.255.11/24"
      ens38:
        addresses:
          - "10.255.255.11/24"
```

Related problem

the configuration for vm1 was ok but need to update the configuration for vm2 and vm3 so...

```
- name: Adjust ens38 for vm3
  delegate_to: '192.168.58.13'
  replace:
    path: /etc/netplan/00-installer-config.yaml
    regexp: '- 10.255.255.11/24'
    replace: '- 10.255.255.13/24'

- name: Adjust ens33 for vm3.
  delegate_to: '192.168.58.13'
  replace:
    path: /etc/netplan/00-installer-config.yaml
    regexp: '- 192.168.255.11/24'
    replace: '- 192.168.255.13/24'

- name: Adjust ens38 for vm2.
  delegate_to: '192.168.58.12'
  replace:
    path: /etc/netplan/00-installer-config.yaml
    regexp: '- 10.255.255.11/24'
    replace: '- 10.255.255.12/24'

- name: Adjust ens33 for vm2.
  delegate_to: '192.168.58.12'
  replace:
    path: /etc/netplan/00-installer-config.yaml
    regexp: '- 192.168.255.11/24'
    replace: '- 192.168.255.12/24'
```

My playbook 2

Set host name

```
- name: Set the hostname.
  hostname:
    name: "{{ ansible_hostname }}.localdomain"

- name: Display the changed config
  debug:
    msg: "The new hostname is {{ ansible_hostname }}.localdomain and the OS is {{ ansible_host }}"
```

My playbook

GlusterFS

GlusterFS is a distributed, arbitrarily scalable file system that aggregates storage components from several servers into one, uniform file system

My playbook 3

install GlusterFS & enable it

```
- name: install GlusterFS and enable it.
  shell: sudo apt-get install glusterfs-server -y && sudo systemctl start glusterd && sudo systemctl er


- name: peer with other node.
  shell: sudo gluster peer probe vm2 && sudo gluster peer probe vm3
  run_once: true
```

create trusted pool

```
- name: Create a trusted storage pool
  gluster.gluster.gluster_peer:
    state: present
    nodes:
      - vm1g
      - vm2g
      - vm3g
```

My playbook 4

Gluster volume

```
- name: Create Gluster volume
  ignore_errors: yes
  shell: gluster volume create gfs replica 3 transport tcp vm1g:/gluster/bricks/1/brick vm2g:/gluster/bricks/2/brick vm3g:/gluster/bricks/3/brick
  run_once: true

- name: Start Gluster volume
  shell: gluster volume start gfs
  run_once: true

- name: Set security configration
  shell: gluster volume set gfs auth.allow 10.255.255.11,10.255.255.12,10.255.255.13
  run_once: true
```

My playbook 5

Mount point

```
- name: Create a mountpoint
  shell: mkdir /data

- name:  Add a fstab entry.
  shell: echo 'vm1g:gfs /data glusterfs defaults,_netdev,backupvolfile-server=localhost 0 0' >> /etc/fstab

- name: mount the volume
  shell: mount -a
```

Application: CMS

a docker image that automatically installs and configures Drupal

```
FROM drupal:9.2.6-php7.4-fpm-alpine3.14

# Copy the artefact (on Bitbucket) or local files into the image
COPY ./ /opt/drupal/.

# PHP Memcached which needs the libmemcached API
# Bug in Apline - creates extensions in the wrong directory so add them in correct location
RUN apk add php7-igbinary php7-pecl-memcached libmemcached \
    && echo "extension=memcached.so" > /usr/local/etc/php/conf.d/docker-php-ext-memcached.ini \
    && echo "extension=igbinary.so" > /usr/local/etc/php/conf.d/docker-php-ext-igbinary.ini \
    && mv /usr/lib/php7/modules/memcached.so /usr/local/lib/php/extensions/no-debug-non-zts-20190902 \
    && mv /usr/lib/php7/modules/igbinary.so /usr/local/lib/php/extensions/no-debug-non-zts-20190902

# APCu cache
RUN apk add --update --no-cache --virtual .build-dependencies $PHPIZE_DEPS \
    && pecl install apcu \
    && docker-php-ext-enable apcu \
    && pecl clear-cache \
    && apk del .build-dependencies
```

Application: DB

```
mariadb:
  image: mariadb:10.4.0
  container_name: "mariadb"
  environment:
    MYSQL_USER: drupal
    MYSQL_PASSWORD: drupal
    MYSQL_DATABASE: drupal
    MYSQL_ROOT_PASSWORD: ''
    MYSQL_ALLOW_EMPTY_PASSWORD: 'yes'
  restart: always
  volumes:
    - mysqlV:/var/lib/mysql
  ports:
    - '3306:3306'
  expose:
    - '3306'
  networks:
    - internal

pmy:
  image: phpmyadmin/phpmyadmin
  container_name: "pmy"
  depends_on:
    - mariadb
  environment:
    PMA_HOSTS: 192.168.58.11,192.168.58.12,192.168.58.13
    PMA_PORT: 3306
    PMA_ARBITRARY: 1
  restart: always
  labels:
    - "traefik.enable=true"
    - "traefik.http.routers.pmy.rule=Host(`pmy.localdomain`)"
    - "traefik.http.routers.pmy.entrypoints=websecure"
    - "traefik.http.routers.pmy.tls.certresolver=myresolver"
  networks:
   - internal
```

Application: Web sever

Docker image with nginx with Web Application Firewall (ModSecurity 3) and preconfigured OWASP ModSecurity Core Rule Set (CRS).

bit3/nginx-waf ☆

By bit3 • Updated an hour ago

Docker image with nginx with WAF (ModSecurity 3) and preconfigured OWASP ModSecurity CRS.

Container

```
cms:
  image: bit3/nginx-waf:latest
  volumes:
    - /gfs:/data
    - drupal-data:/var/www/html
    - ./nginx-conf:/etc/nginx/conf.d
  depends_on:
    - drupal
  container_name: "cms"
  labels:
    - "traefik.enable=true"
    - "traefik.http.routers.cms.rule=Host(`www.localdomain`)"
    - "traefik.http.routers.cms.entrypoints=websecure"
    - "traefik.http.routers.cms.tls.certresolver=myresolver"
```

Application: Centralized Logging (ELK)

Logstash

The server component of Logstash that processes incoming logs

Elasticsearch

Stores all of the logs

Kibana

Web interface for searching and visualizing logs, which will be proxied through Nginx

External Access: Traefik

Traefik is a leading modern reverse proxy and load balancer that makes deploying microservices easy

```yaml
revproxy:
# https://hub.docker.com/_/traefik
  image: "traefik:v2.5"
  container_name: "traefik"
  command:
    # - "--log.level=DEBUG"
    - "--api.insecure=true"
    - "--providers.docker=true"
    - "--providers.docker.exposedbydefault=false"
    - "--entrypoints.websecure.address=:443"
    - "--entrypoints.web.address=:80"
    - "--entrypoints.web.http.redirections.entryPoint.to=websecure"
    - "--entrypoints.web.http.redirections.entryPoint.scheme=https"
    - "--entrypoints.web.http.redirections.entrypoint.permanent=true"
    - "--certificatesresolvers.myresolver.acme.tlschallenge=true"
    #- "--certificatesresolvers.myresolver.acme.caserver=https://acme-staging-v02.api.letsencrypt.org/di
    - "--certificatesresolvers.myresolver.acme.email=pariabalf@gmail.com"
    - "--certificatesresolvers.myresolver.acme.storage=/letsencrypt/acme.json"
  ports:
    - "80:80"
    - "443:443"
    - "8080:8080"
  depends_on:
    - cms
  networks:
    - external
    - internal
```

Use 3 volumes with docker-compose

- drupal-data
- mysqlV
- elasticsearch