Ansible - LAMP Stack + HAProxy Load Balancer

```
# Ansible - LAMP Stack + HAProxy Load Balancer
# Source: github.com/ansible/ansible-examples
#         lamp_haproxy

---
- name: Apply common configuration to all nodes
  hosts: all
  become: true
  vars:
    ntp_server: time.google.com
    security_updates: true

  tasks:
    - name: Install common packages
      package:
        name:
          - ntp
          - wget
          - curl
          - firewalld
          - python3-pip
        state: present

    - name: Configure NTP
      template:
        src: templates/ntp.conf.j2
        dest: /etc/ntp.conf
      notify: Restart NTP

    - name: Start firewalld
      service:
        name: firewalld
        state: started
        enabled: yes

    - name: Apply security updates
      yum:
        name: '*'
        state: latest
        security: yes
      when: security_updates | bool
      tags: [security]

  handlers:
    - name: Restart NTP
      service:
        name: ntpd
        state: restarted

---
- name: Configure Database Servers
  hosts: dbservers
  become: true
  vars:
    mysql_root_pw: "{{ vault_mysql_root_pw }}"
    mysql_databases:
      - name: webapp
        encoding: utf8mb4
        collation: utf8mb4_unicode_ci
    mysql_users:
      - name: webapp_user
        password: "{{ vault_webapp_db_pw }}"
        priv: "webapp.*:ALL"
      host: "10.0.%"
  mysql_max_connections: 200
  mysql_innodb_buffer_pool: "1G"

tasks:
  - name: Install MariaDB
    yum:
      name:
        - mariadb-server
        - mariadb
        - MySQL-python
      state: present

  - name: Deploy MariaDB config
    template:
      src: templates/my.cnf.j2
      dest: /etc/my.cnf
    notify: Restart MariaDB

  - name: Start MariaDB
    service:
      name: mariadb
      state: started
      enabled: yes

  - name: Set root password
    mysql_user:
      name: root
      password: "{{ mysql_root_pw }}"
      login_unix_socket: /var/lib/mysql/mysql.sock
    no_log: true

  - name: Create databases
    mysql_db:
      name: "{{ item.name }}"
      encoding: "{{ item.encoding }}"
      collation: "{{ item.collation }}"
      state: present
      login_user: root
      login_password: "{{ mysql_root_pw }}"
    loop: "{{ mysql_databases }}"

  - name: Create database users
    mysql_user:
      name: "{{ item.name }}"
      password: "{{ item.password }}"
      priv: "{{ item.priv }}"
      host: "{{ item.host }}"
      state: present
      login_user: root
      login_password: "{{ mysql_root_pw }}"
    loop: "{{ mysql_users }}"
    no_log: true

  - name: Open MySQL port
    firewalld:
      port: 3306/tcp
      permanent: yes
      state: enabled
      source: "10.0.0.0/24"
    notify: Reload firewalld

handlers:
  - name: Restart MariaDB
    service:
      name: mariadb
```

```
        state: restarted

    - name: Reload firewalld
      service:
        name: firewalld
        state: reloaded

---
- name: Configure Web Servers
  hosts: webservers
  become: true
  vars:
    http_port: 80
    app_root: /var/www/webapp
    php_packages:
      - php
      - php-fpm
      - php-mysql
      - php-gd
      - php-mbstring
    max_children: 50

  tasks:
    - name: Install Apache and PHP
      yum:
        name: "{{ ['httpd'] + php_packages }}"
        state: present

    - name: Deploy Apache vhost
      template:
        src: templates/vhost.conf.j2
        dest: /etc/httpd/conf.d/webapp.conf
      notify: Restart Apache

    - name: Configure PHP-FPM pool
      template:
        src: templates/www.conf.j2
        dest: /etc/php-fpm.d/www.conf
      notify: Restart PHP-FPM

    - name: Create app directory
      file:
        path: "{{ app_root }}"
        state: directory
        owner: apache
        group: apache
        mode: "0755"

    - name: Deploy application
      git:
        repo: "https://github.com/example/webapp.git"
        dest: "{{ app_root }}"
        version: "{{ app_version | default('main') }}"
      notify: Restart Apache
      tags: [deploy]

    - name: Set app config
      template:
        src: templates/app-config.php.j2
        dest: "{{ app_root }}/config.php"
        owner: apache
        mode: "0640"
      tags: [deploy, config]

    - name: Start services
      service:
        name: "{{ item }}"
        state: started
        enabled: yes
      loop:
        - httpd
        - php-fpm

    - name: Open HTTP port
      firewalld:
        port: "{{ http_port }}/tcp"
        permanent: yes
        state: enabled
      notify: Reload firewalld

  handlers:
    - name: Restart Apache
      service:
        name: httpd
        state: restarted

    - name: Restart PHP-FPM
      service:
        name: php-fpm
        state: restarted

    - name: Reload firewalld
      service:
        name: firewalld
        state: reloaded

---
- name: Configure HAProxy Load Balancer
  hosts: lbservers
  become: true
  vars:
    haproxy_stats_port: 8404
    haproxy_stats_user: admin
    haproxy_stats_pw: "{{ vault_haproxy_stats_pw }}"
    backend_servers: "{{ groups['webservers'] }}"

  tasks:
    - name: Install HAProxy
      yum:
        name: haproxy
        state: present

    - name: Deploy HAProxy config
      template:
        src: templates/haproxy.cfg.j2
        dest: /etc/haproxy/haproxy.cfg
        validate: haproxy -c -f %s
      notify: Reload HAProxy

    - name: Configure HAProxy logging
      copy:
        content: |
          $ModLoad imudp
          $UDPServerAddress 127.0.0.1
          $UDPServerRun 514
          local2.* /var/log/haproxy.log
        dest: /etc/rsyslog.d/haproxy.conf
      notify: Restart rsyslog

    - name: Enable SELinux boolean
      seboolean:
        name: haproxy_connect_any
```

```yaml
        state: yes
        persistent: yes
      when: ansible_selinux.status == "enabled"

    - name: Start HAProxy
      service:
        name: haproxy
        state: started
        enabled: yes

    - name: Open LB ports
      firewalld:
        port: "{{ item }}/tcp"
        permanent: yes
        state: enabled
      loop:
        - "80"
        - "443"
        - "{{ haproxy_stats_port }}"
      notify: Reload firewalld

  handlers:
    - name: Reload HAProxy
      service:
        name: haproxy
        state: reloaded

    - name: Restart rsyslog
      service:
        name: rsyslog
        state: restarted

    - name: Reload firewalld
      service:
        name: firewalld
        state: reloaded
```