Demo Project: Automate Node.js application Deployment

Project Description

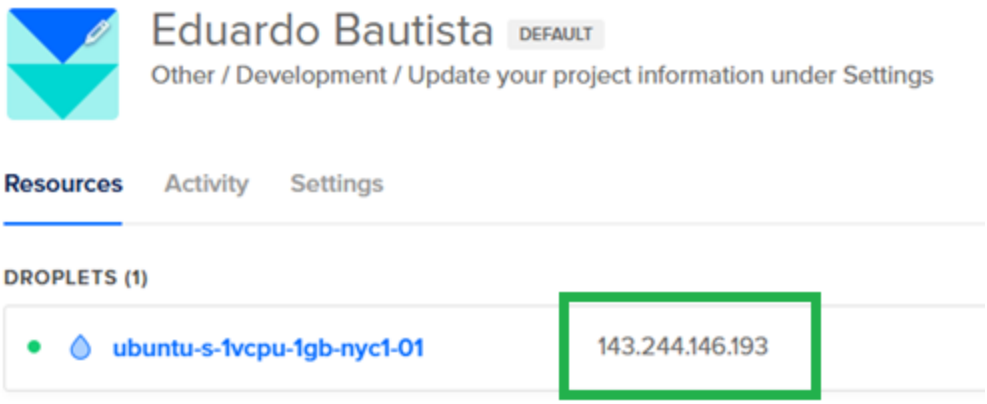
In this project, we will use **Ansible** to automate the deployment of a **Node.js** application on a server hosted on **DigitalOcean**. The Ansible Playbook will:

- Install necessary technologies like Node.js and npm.
- Create a dedicated Linux user for the application.
- Deploy and run the Node.js application using that user.
- Ensure the application is running successfully.

Step 1: Create Server on DigitalOcean

1. Go to **DigitalOcean**.
2. Create a new **Droplet**.
3. Select the **Region** closest to you.
4. Choose the **Ubuntu** distribution.
5. **Shared CPU Basic**:
    - Regular Disk type: **SSD**
    - **1 GB / 1 CPU**

6. **Authentication Method**:
    - Use your existing **SSH key**.
7. Click **Create Droplet**.
8. Copy the new **Public IP Address**.



Step 2: Configure Ansible Inventory

Create a file called `hosts` and add the following content:

```
[webserver]
<your-public-ip> ansible_ssh_private_key_file=~/.ssh/id_rsa ansible_user=root
```

Replace `<your-public-ip>` with your DigitalOcean Droplet's public IP address.

Step 3: Write Ansible Playbook

Create a file named **deploy-node.yaml** with the following content:

```yaml
---
- name: Install node and npm
  hosts: 143.244.146.193
  tasks:
    - name: Update apt repo and cache
      apt: update_cache=yes force_apt_get=yes cache_valid_time=3600
    - name: Install nodejs and npm
      apt:
        pkg:
          - nodejs
          - npm


- name: Create new linux user for node app
  hosts: 143.244.146.193
  tasks:
    - name: Create linux user
      user:
        name: NodeUser
        comment: Node User
        group: admin


- name: Deploy nodejs app
  hosts: 143.244.146.193
  become: True
  become_user: NodeUser
  tasks:
    - name: Unpack the nodejs file
      unarchive:
        src: /mnt/c/Users/eduar/devops_projects2/08-ansible/nodejs-app/nodejs-app-1.0.0.tgz
        dest: /home/NodeUser
    - name: Install dependencies
      npm:
        path: /home/NodeUser/package
    - name: Start the application
      command:
        chdir: /home/NodeUser/package/app
        cmd: node server
      async: 1000
      poll: 0
    - name: Ensure app is running
      shell: ps aux | grep node
      register: app_status
    - debug: msg={{app_status.stdout_lines}}
```

Step 4: Run Ansible Playbook

Execute the playbook using the following command: `ansible-playbook -i hosts deploy-node.yaml`

Step 5: Verify Deployment

SSH into the server: `ssh root@<your-public-ip>`

List the files in the Node.js application directory: `ls /home/NodeUser/package`

Expected Output:

```
Dockerfile Readme.md app node_modules package-lock.json package.jso
n
```

Step 6: Use Ansible Variables

Variables can be used to make the playbook more dynamic. You can find the reference for this in the following repository: https://gitlab.com/twn-devops-projects/ansible/ansible-projects/-/tree/feature/variables?ref_type=heads

Step 7: Clean Up

- Delete the server from the DigitalOcean dashboard.