Ansible playbook

Hello! And welcome back to the Terraform and Ansible for AWS course on Linux Academy, my name is derek, and in this lesson, we’re going to create the ansible playbook that will deploy wordpress on our dev server!

It is best if you have watched the Introduction to Ansible course on Linux Academy before continuing this lesson, as I will not go into much depth for this one as it is pretty self-explanatory if you have completed the Ansible lessons.

Here is the Ansible playbook I have created:

|  |
| --- |
| --- |
|  | - hosts: dev |
|  | become: yes |
|  | remote\_user: ec2-user |
|  | tasks: |
|  | - name: wait for server to start |
|  | local\_action: |
|  | module: wait\_for |
|  | host=dev |
|  | port=22 |
|  | delay=1 |
|  | timeout=300 |
|  | - name: Install Apache. |
|  | yum: name={{ item }} state=present |
|  | with\_items: |
|  | - httpd |
|  | - php |
|  | - php-mysql |
|  | - name: Make sure Apache is started now and at boot. |
|  | service: name=httpd state=started enabled=yes |
|  | - name: Download WordPress |
|  | get\_url: url=http://wordpress.org/wordpress-latest.tar.gz dest=/var/www/html/wordpress.tar.gz force=yes |
|  | - name: Extract WordPress |
|  | command: "tar xzf /var/www/html/wordpress.tar.gz -C /var/www/html --strip-components 1" |
|  | - name: Make my directory tree readable |
|  | file: |
|  | path: /var/www/html/ |
|  | mode: u=rwX,g=rX,o=rX |
|  | recurse: yes |
|  | owner: apache |
|  | group: apache |
|  | - name: Make sure Apache is started now and at boot. |
|  | service: name=httpd state=started enabled=yes |

As you can see, we are connecting to our dev host. The IP for this host will be added to the aws\_hosts file when we create the dev instance in terraform.

After we have connected to our instance, we install Apache, php, and php-mysql. We then start and enable the services.

Following the Apache installation, we download and extract wordpress, then we apply proper permissions.

If you want to have this script add the database information as well, you can certainly do that. I recommend checking out the “lineinfile” ansible command to get started. This is beyond the scope of this lesson as we are going to manually enter the database details to ensure everything is up and running, but I definitely recommend experimenting with it!

That’s it for this lesson! Please join me in the next one as we continue building our infrastructure!