

REA-TASK

Requirements: AWS Account. Ansible Server . OS: Linux

AWS Portal

1. Create an AWS user with Programmatic Access
2. Grant Full Administrator access for this user (Please delete the user after testing this playbook)
3. The ACCESS KEY ID AND SECRET will be used in the */root/.boto file in step 11 of HowTO*
4. *Save and quit the file*

Ansible Installation

1. Login to your AWS portal
2. Create a Instance with **ami-02fd0b06f06d93dfc** AMI (Preferred as its tested)
3. Once its booted and up, login to the Instance with the ssh key you generated.
4. Use the below commands to install Ansible:

```
sudo pip install ansible
sudo pip install boto3
sudo pip install botocore
```

HowTO

1. Download the hosts and sinatra-final.yml file and place it in your working directory, where you will execute ansible from.
2. git clone <https://github.com/amnotadeveloper/rea.git>
3. You may need to use sudo if using non-root account
4. Download your AWS account key and place it in the file ~/.ssh/aws
5. Make sure to correct permissions: `chmod 400 ~/.ssh/aws`
6. If the ansible.cfg is not created by default, please create one.
7. Create / etc/ansible directory if not already created.
8. Open / etc/ansible/ansible.cfg with favourite editor and paste the below lines:

```
[defaults]
host_key_checking = False
```

9. If the file ansible.cfg exists, there would already be an entry for `host_key_checking = False` which you may just uncomment.
10. create ~/.boto file
11. Paste your AWS credentials which you created earlier in this format ~/.boto

```
[Credentials]
AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
```

After all the steps have been completed: run

go to the git directory where the clone was create e.g ~/rea and run

```
ansible-playbook -vv -i hosts sinatra-final.yml --private-key=~/.ssh/aws -u ec2-user
```

Please go to the IP of the newly created Instance which should say Hello World!

Explanation:

1. boto and botocore: Python sdk for AWS CLI
2. host_key_checking = False (This will not halt the play while running to accept the host key because it uses ssh to configure the packages)
3. The .boto file is used to provide AWS credentials to create the Instance.

Design Choices

- The application its self can run on port 80, however, that was not scalable
- nginx is lightweight and better at handling more traffic.
- In case of multiple upstreams to the app, nginx is more flexible
- The current design can be made more flexible by using 2 EC2 instances and an ELB on a domain.
- I could have done in a number of ways, however, Ansible is a tool I have been using daily for quite a while now, so reason for my choice.