**DevOps: Using Ansible to provision AWS EC2 instances**

**A practical way to provision instances on Amazon Web Service EC2 with Ansible.**

**Welcome, this article shows a simple approach to use Ansible for provisioning an AWS EC2 instance.**

**The**[**Ansible**](https://docs.ansible.com/ansible/latest/index.html)**is a configuration management tool widely used to provision IT environments, deploy software or be integrated to CI/CD pipelines. There are lots of Ansible**[**modules**](https://docs.ansible.com/ansible/latest/modules/list_of_cloud_modules.html#amazon)**developed to ease tasks related to AWS cloud management.**

**The following steps will be performed along the article to demonstrate the power around the integration of Ansible and AWS Cloud:**

* **Create AWS user**
* **Install Ansible and Ansible EC2 module dependencies**
* **Create SSH keys**
* **Create Ansible structure**
* **Run Ansible to provision the EC2 instance**
* **Connect to the EC2 instance via SSH**

## **Create AWS user**

**Open the AWS**[**Console**](https://console.aws.amazon.com/)**, search for IAM (Identity and Access Management) and follow this**[**steps**](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html#id_users_create_console)**to create a user and take note of the Access Key and Secret Key that will be used by Ansible to set up the instances.**

**Graphical user interface, text, application, Teams

Description automatically generated**

## **Install Ansible and the EC2 module dependencies**

**sudo apt install python  
sudo apt install python-pip  
pip install boto boto3 ansible**

**This article was written with Ansible version 2.8.0 and Python version 2.7.**

## **Create SSH keys to connect to the EC2 instance after provisioning**

**ssh-keygen -t rsa -b 4096 -f ~/.ssh/my\_aws**

## **Create the Ansible directory structure**

**mkdir -p AWS\_Ansible/group\_vars/all/  
cd AWS\_Ansible  
touch playbook.yml**

**Optionally, you can use Git (or SVN) to keep the version control of this directory.**

## **Create Ansible Vault file to store the AWS Access and Secret keys.**

**ansible-vault create group\_vars/all/pass.yml  
New Vault password:  
Confirm New Vault password:**

**The password provided here will be asked every time the playbook is executed or when editing the pass.yml file.**

**This article will follow the approach above, however, if you don’t want to provide the password every time, an insecure approach can create the pass.yml file by specifying a hashed password file:**

**openssl rand -base64 2048 > vault.passansible-vault create group\_vars/all/pass.yml --vault-password-file vault.pass**

**With hashed password file you must specify the vault-password-file argument when running Ansible playbook and won’t be asked for the password:**

**ansible-playbook playbook.yml --vault-password-file vault.pass**

## **Edit the pass.yml file and create the keys global constants**

**Create the variables ec2\_access\_key and ec2\_secret\_key and set the values gathered after user creation (IAM).**

**ansible-vault edit group\_vars/all/pass.yml   
Vault password:  
ec2\_access\_key: AAAAAAAAAAAAAABBBBBBBBBBBB   
ec2\_secret\_key: afjdfadgf$fgajk5ragesfjgjsfdbtirhf**

## **Directory structure**

**➜ AWS\_Ansible tree  
.  
├── group\_vars  
│ └── all  
│ └── pass.yml  
└── playbook.yml2 directories, 2 files**

## **Open the playbook.yml file and past the following content**

|  |
| --- |
| # AWS playbook |
|  | --- |
|  |  |
|  | - hosts: localhost |
|  | connection: local |
|  | gather\_facts: False |
|  |  |
|  | vars: |
|  | key\_name: my\_aws |
|  | region: us-east-2 |
|  | image: ami-0f93b5fd8f220e428 # https://cloud-images.ubuntu.com/locator/ec2/ |
|  | id: "web-app" |
|  | sec\_group: "{{ id }}-sec" |
|  |  |
|  | tasks: |
|  |  |
|  | - name: Facts |
|  | block: |
|  |  |
|  | - name: Get instances facts |
|  | ec2\_instance\_facts: |
|  | aws\_access\_key: "{{ec2\_access\_key}}" |
|  | aws\_secret\_key: "{{ec2\_secret\_key}}" |
|  | region: "{{ region }}" |
|  | register: result |
|  |  |
|  | - name: Instances ID |
|  | debug: |
|  | msg: "ID: {{ item.instance\_id }} - State: {{ item.state.name }} - Public DNS: {{ item.public\_dns\_name }}" |
|  | loop: "{{ result.instances }}" |
|  |  |
|  | tags: always |
|  |  |
|  |  |
|  | - name: Provisioning EC2 instances |
|  | block: |
|  |  |
|  | - name: Upload public key to AWS |
|  | ec2\_key: |
|  | name: "{{ key\_name }}" |
|  | key\_material: "{{ lookup('file', '/Users/julianosilva/.ssh/{{ key\_name }}.pub') }}" |
|  | region: "{{ region }}" |
|  | aws\_access\_key: "{{ec2\_access\_key}}" |
|  | aws\_secret\_key: "{{ec2\_secret\_key}}" |
|  |  |
|  | - name: Create security group |
|  | ec2\_group: |
|  | name: "{{ sec\_group }}" |
|  | description: "Sec group for app {{ id }}" |
|  | # vpc\_id: 12345 |
|  | region: "{{ region }}" |
|  | aws\_access\_key: "{{ec2\_access\_key}}" |
|  | aws\_secret\_key: "{{ec2\_secret\_key}}" |
|  | rules: |
|  | - proto: tcp |
|  | ports: |
|  | - 22 |
|  | cidr\_ip: 0.0.0.0/0 |
|  | rule\_desc: allow all on ssh port |
|  | register: result\_sec\_group |
|  |  |
|  | - name: Provision instance(s) |
|  | ec2: |
|  | aws\_access\_key: "{{ec2\_access\_key}}" |
|  | aws\_secret\_key: "{{ec2\_secret\_key}}" |
|  | key\_name: "{{ key\_name }}" |
|  | id: "{{ id }}" |
|  | group\_id: "{{ result\_sec\_group.group\_id }}" |
|  | image: "{{ image }}" |
|  | instance\_type: t2.micro |
|  | region: "{{ region }}" |
|  | wait: true |
|  | count: 1 |
|  | # exact\_count: 2 |
|  | # count\_tag: |
|  | # Name: App |
|  | # instance\_tags: |
|  | # Name: App |
|  |  |
|  | tags: ['never', 'create\_ec2'] |

## **Notes about the playbook**

* **For security, the playbook will execute by default just the tasks to collect information on AWS. The tasks responsible for provisioning the instance will be performed if specified the tag create\_ec2.**
* **The first step to create the user (IAM) can also be performed with the Ansible [iam](https://docs.ansible.com/ansible/latest/modules/iam_module.html" \l "iam-module" \t "_blank) module, but here was demonstrated on the AWS Console to show the interaction.**

## **Running Ansible to provision instances**

**If you execute Ansible without the tags argument the creation tasks won’t be performed.**

**ansible-playbook playbook.yml --ask-vault-pass**

**A picture containing application

Description automatically generated**

## **Create the instance**

**ansible-playbook playbook.yml --ask-vault-pass --tags create\_ec2**

**Text

Description automatically generated with medium confidence**

## **Get the public DNS**

**ansible-playbook playbook.yml --ask-vault-pass**

**Text

Description automatically generated**

## **Connect to the EC2 instance via SSH**

**ssh -i ~/.ssh/my\_aws** [**ubuntu@ec2-18-218-148-27.us-east-2.compute.amazonaws.com**](mailto:ubuntu@ec2-18-218-148-27.us-east-2.compute.amazonaws.com)

**Text, letter

Description automatically generated**

**Congratulations, you’ve automated the EC2 instance provisioning process with Ansible.**