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.

Agile Management: The Good, The Bad, and the Downright Ugly | Data Driven Investor
Companies are constantly reinventing themselves to gain or maintain competitive advantage and market share. This is…
www.datadriveninvestor.com

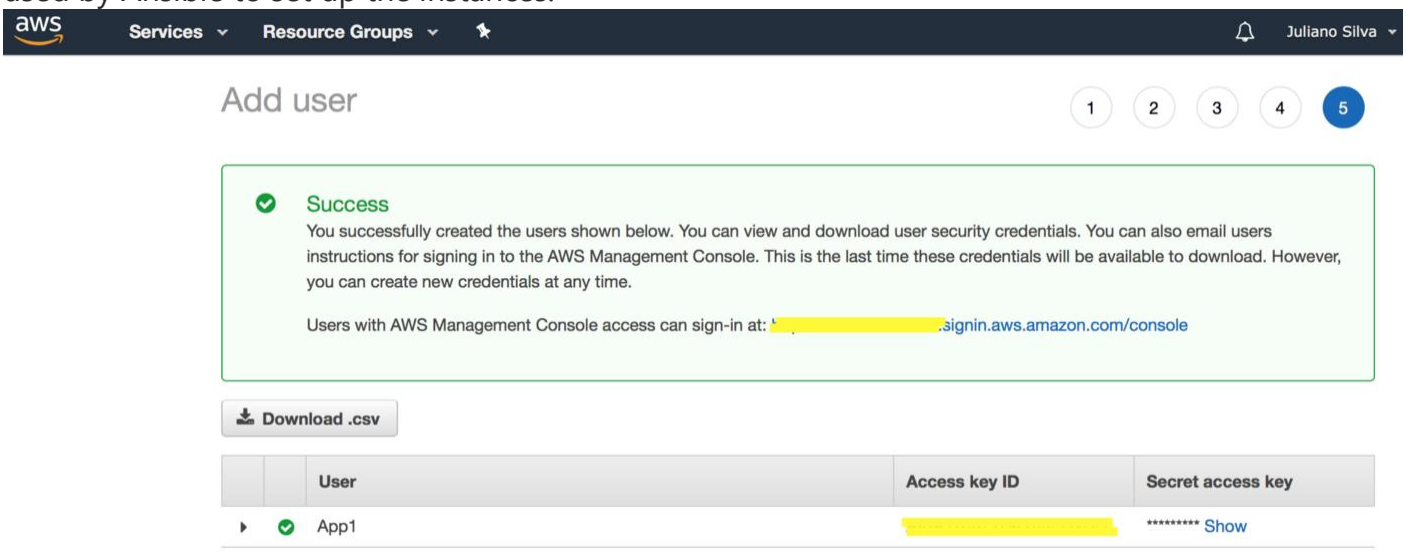
The **Ansible** is a configuration management tool widely used to provision IT environments, deploy software or be integrated to CI/CD pipelines. There are lots of Ansible **modules** developed to ease tasks related to AWS cloud management.

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, search for IAM (Identity and Access Management) and follow this **steps** 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.



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: AAAAAAAAAAAAAABBBBBBBBBBB
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

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** 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
```

```
➜  AWS_Ansible ansible-playbook playbook.yml --ask-vault-pass
Vault password:
 [WARNING]: No inventory was parsed, only implicit localhost is available

 [WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'


PLAY [localhost] ***************************************************************************************************************

TASK [Get instances facts] *****************************************************************************************************
ok: [localhost]

TASK [Instances ID] ************************************************************************************************************

PLAY RECAP *********************************************************************************************************************
localhost                  : ok=1    changed=0    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

Create the instance

```
ansible-playbook playbook.yml --ask-vault-pass --tags create_ec2
```

```
→  AWS_Ansible ansible-playbook playbook.yml --ask-vault-pass --tags create_ec2
Vault password:
 [WARNING]: No inventory was parsed, only implicit localhost is available

 [WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match 'all'


PLAY [localhost] ***************************************************************************************************************

TASK [Get instances facts] *****************************************************************************************************
ok: [localhost]

TASK [Instances ID] ************************************************************************************************************

TASK [Upload public key to AWS] ************************************************************************************************
ok: [localhost]

TASK [Create security group] ***************************************************************************************************
changed: [localhost]

TASK [Provision instance(s)] ***************************************************************************************************
changed: [localhost]

PLAY RECAP *********************************************************************************************************************
localhost                  : ok=4    changed=2    unreachable=0    failed=0    skipped=1    rescued=0    ignored=0
```

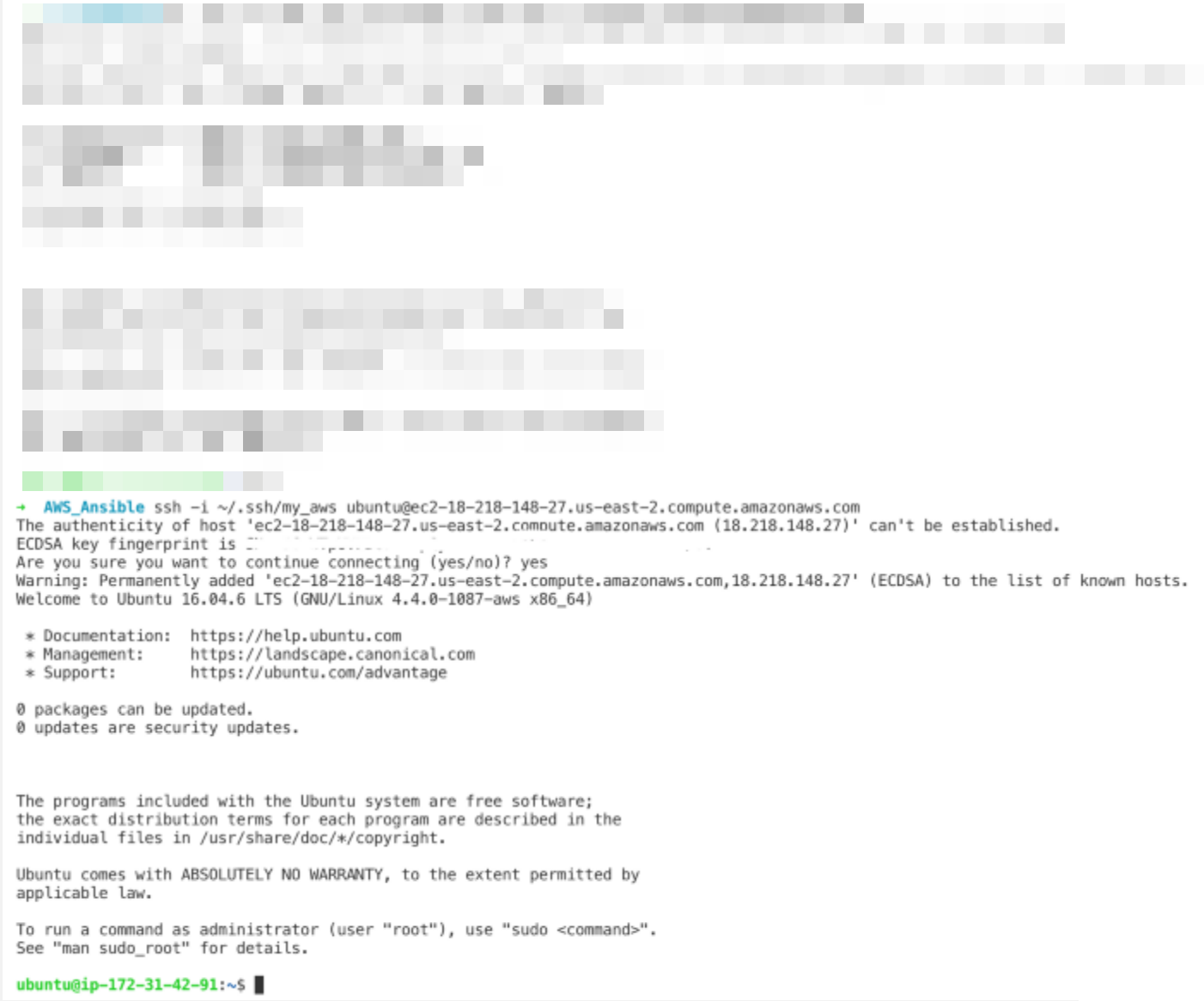
Get the public DNS

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

```
→  AWS_Ansible ansible-playbook playbook.yml --ask-vault-pass
Vault password:
 [WARNING]: No inventory was parsed, only implicit localhost is available

 [WARNING]: provided hosts list is empty, only localhost is available. Note that the implicit localhost does not match


PLAY [localhost] ***************************************************************************************************************

TASK [Get instances facts] *****************************************************************************************************
ok: [localhost]

TASK [Instances ID] ************************************************************************************************************
ok: [localhost] => (item={u'root_device_type': u'ebs', u'private_dns_name': u'ip-172-31-42-91.us-east-2.compute.intern
re': 1, u'core_count': 1}, u'security_groups': [{u'group_id': u'sg-00a7c8b501b7029a6', u'group_name': u'webapp-01-sec'
ed'}, u'subnet_id': u'subnet-cdb87d81', u'ebs_optimized': False, u'state': {u'code': 16, u'name': u'running'}, u'sourc
u'webapp-01', u'virtualization_type': u'hvm', u'root_device_name': u'/dev/sda1', u'public_ip_address': u'18.218.148.27
, u'image_id': u'ami-0f93b5fd8f220e428', u'ena_support': True, u'hibernation_options': {u'configured': False}, u'capac
acity_reservation_preference': u'open'}, u'public_dns_name': u'ec2-18-218-148-27.us-east-2.compute.amazonaws.com', u'b
atus': u'attached', u'delete_on_termination': True, u'attach_time': u'2019-08-06T04:59:20+00:00', u'volume_id': u'vol-
'/dev/sda1'}], u'placement': {u'availability_zone': u'us-east-2c', u'tenancy': u'default', u'group_name': u''}, u'ami_
', u'network_interfaces': [{u'status': u'in-use', u'description': u'', u'subnet_id': u'subnet-cdb87d81', u'ipv6_addres
ni-03eb288b0f6b4005d', u'private_dns_name': u'ip-172-31-42-91.us-east-2.compute.internal', u'attachment': {u'status':
achment_id': u'eni-attach-0d6c0c1cd3138dbda', u'delete_on_termination': True, u'attach_time': u'2019-08-06T04:59:19+00
ivate_ip_address': u'172.31.42.91', u'private_dns_name': u'ip-172-31-42-91.us-east-2.compute.internal', u'association'
ublic_dns_name': u'ec2-18-218-148-27.us-east-2.compute.amazonaws.com', u'ip_owner_id': u'amazon'}, u'primary': True}],
 u'private_ip_address': u'172.31.42.91', u'vpc_id': u'vpc-75201c1d', u'groups': [{u'group_id': u'sg-00a7c8b501b7029a6'
'association': {u'public_ip': u'18.218.148.27', u'public_dns_name': u'ec2-18-218-148-27.us-east-2.compute.amazonaws.co
ce_dest_check': True, u'owner_id': u'894657971551'}], u'launch_time': u'2019-08-06T04:59:19+00:00', u'instance_id': u'
: u't2.micro', u'architecture': u'x86_64', u'state_transition_reason': u'', u'private_ip_address': u'172.31.42.91', u'
des': []}) => {
    "msg": "ID: i-00ca7973cf3628a86 - State: running - Public DNS: ec2-18-218-148-27.us-east-2.compute.amazonaws.com"
}

PLAY RECAP *********************************************************************************************************************
localhost                  : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

Connect to the EC2 instance via SSH

ssh -i ~/.ssh/my_aws ubuntu@ec2-18-218-148-27.us-east-2.compute.amazonaws.com



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