Here is my vagrant file for AWS
Vagrant.configure("2") do |config|
  config.vm.box = "dummy"
  config.vm.provision :shell, path: "bootstrap.sh"
  config.vm.provider :aws do |aws, override|
    aws.access_key_id = ""
    aws.secret_access_key = ""
    aws.keypair_name = ""
    aws.region = "us-east-2"
    aws.ami = "ami-c5ba9fa0"
    aws.instance_type="t2.micro"
    override.ssh.username = "ubuntu"
    override.ssh.private_key_path = ""
  end
end

Vagrantfile (END)
First create an access key on AWS Website
Your Security Credentials

Use this page to manage the credentials for your AWS account. To manage credentials for AWS Identity and Access Management (IAM) users, use the IAM Console.

To learn more about the types of AWS credentials and how they’re used, see AWS Security Credentials in AWS General Reference.

- Password
- Multi-factor authentication (MFA)
- Access keys (access key ID and secret access key)
- CloudFront key pairs
- X.509 certificate
- Account identifiers
Don't create global key here, create a user
You can add multiple users at once with the same access type and permissions. Learn more

**User name**

```
sally_123
```

[Add another user]

**Select AWS access type**

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. Learn more

**Access type**

- Programmatic access
  - Enables an access key ID and secret access key for the AWS API, CLI, SDK, and other development tools.
- AWS Management Console access
  - Enables a password that allows users to sign-in to the AWS Management Console.
What permissions should that user have?
Create a new group
Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users’ permissions by job functions, AWS service access, or your custom permissions. Learn more

Group name: ec2_access_only

Filter: Policy type

<table>
<thead>
<tr>
<th>Policy name</th>
<th>Type</th>
<th>Attachments</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AmazonEC2Container</td>
<td>AWS managed</td>
<td>0</td>
<td>Provides administrative access to Amazon ECS resources.</td>
</tr>
<tr>
<td>AmazonEC2Container</td>
<td>AWS managed</td>
<td>0</td>
<td>Default policy for Amazon ECS service role.</td>
</tr>
<tr>
<td>AmazonEC2FullAccess</td>
<td>AWS managed</td>
<td>0</td>
<td>Provides full access to Amazon EC2 via the AWS Management Console.</td>
</tr>
<tr>
<td>AmazonEC2Readonly</td>
<td>AWS managed</td>
<td>0</td>
<td>Provides read only access to Amazon EC2 via the AWS Management Console.</td>
</tr>
<tr>
<td>Provides full access to all Amazon EC2 reports via the AWS Management Console.</td>
<td>0</td>
<td>Provides full access to all Amazon EC2 reports via the AWS Management Console.</td>
<td></td>
</tr>
</tbody>
</table>
Set permissions for sally_123

- Add user to group
- Copy permissions from existing user
- Attach existing policies directly

Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more

Create group  Refresh

<table>
<thead>
<tr>
<th>Group</th>
<th>Attached policies</th>
</tr>
</thead>
<tbody>
<tr>
<td>ec2_access_only</td>
<td>AmazonEC2FullAccess</td>
</tr>
<tr>
<td>foobar</td>
<td>AdministratorAccess</td>
</tr>
<tr>
<td>tuna</td>
<td>AdministratorAccess</td>
</tr>
</tbody>
</table>
Add user

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

<table>
<thead>
<tr>
<th>User name</th>
<th>sally_123</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWS access type</td>
<td>Programmatic access - with an access key</td>
</tr>
</tbody>
</table>

Permissions summary

The user shown above will be added to the following groups.

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td>ec2_access_only</td>
</tr>
</tbody>
</table>
Use the access key id and secret access key inside your vagrant file, but do not let these keys get out to unauthorized individuals
Create new ssh keys for this machine
Resources

You are using the following Amazon EC2 resources in the US East (Ohio) region:

- 1 Running Instances
- 0 DedicatedHosts
- 5 Volumes
- 4 Key Pairs
- 0 Placement Groups
- 0 Elastic IPs
- 0 Snapshots
- 0 Load Balancers
- 3 Security Groups

Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage, and networking - for a low, predictable price. Try Amazon Lightsail for free.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.
You do not have any Key Pairs in this region.

Click the "Create Key Pair" button to create your first Key Pair.

Create Key Pair
You do not have any Key Pairs in this region.

Click the "Create Key Pair" button to create your first Key Pair.

Create Key Pair

Key pair name: ec2_keypair

Cancel  Create
Mine downloaded automatically after creation.

Copy to your current directory
joe@yavin:~$ cp ~/Downloads/ec2_keypair.pem /
joe@yavin:~$ echo "Make your vagrant file use this keypair"
Make your vagrant file use this keypair
joe@yavin:~$
Vagrant.configure("2") do |config|
  config.vm.box = "dummy"
  config.vm.provision :shell, path: "bootstrap.sh"
  config.vm.provider :aws do |aws, override|
    aws.access_key_id = "AKIAI683INRYIUPUDD3CQ"
    aws.secret_access_key = "TveAkLjwkaMxZgmjDb1pS0l1x0XZrjhOaboij4if"
    aws.keypair_name = "ec2_keypair"
    aws.region = "us-east-2"
    aws.ami = "ami-c5ba9fa0"
    aws.instance_type= "t2.micro"
    override.ssh.username = "ubuntu"
    override.ssh.private_key_path = "./ec2_keypair.pem"
  end
end

"Vagrantfile" 14L, 506C written
joe@yavin:~/f17/it4200/vagrant/aws_eml_experiment

joe@yavin:~/f17/it4200/vagrant/aws_eml_experiment$ vagrant up provider=aws
Warning! The AWS provider doesn't support any of the Vagrant high-level network configurations (`config.vm.network`). They will be silently ignored.

Launching an instance with the following settings...

- Type: t2.micro
- AMI: ami-c5ba9fa0
- Region: us-east-2
- Keypair: ec2_keypair
- Block Device Mapping: []
- Terminate On Shutdown: false
- Monitoring: false
- EBS optimized: false
- Source Destination check: false
- Assigning a public IP address in a VPC: false
- VPC tenancy specification: default

Waiting for instance to become "ready"...
<table>
<thead>
<tr>
<th>Name</th>
<th>Instance ID</th>
<th>Instance Type</th>
<th>Availability Zone</th>
<th>Instance State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>i-04bc093e318771ba</td>
<td>t2.micro</td>
<td>us-east-2c</td>
<td>stopped</td>
</tr>
<tr>
<td></td>
<td>i-0663240bd5df127a9</td>
<td>t2.micro</td>
<td>us-east-2c</td>
<td>running</td>
</tr>
<tr>
<td></td>
<td>i-07615c0b7a18f1880</td>
<td>t2.micro</td>
<td>us-east-2c</td>
<td>stopped</td>
</tr>
<tr>
<td></td>
<td>i-09c287a5e88629f91</td>
<td>t2.micro</td>
<td>us-east-2a</td>
<td>stopped</td>
</tr>
<tr>
<td></td>
<td>i-0d43a1b7725b220...</td>
<td>t2.micro</td>
<td>us-east-2c</td>
<td>terminated</td>
</tr>
</tbody>
</table>
```bash
default: will be silently ignored.
default: Launching an instance with the following settings...
default:   -- Type: t2.micro
default:   -- AMI: ami-c5ba9fa0
default:   -- Region: us-east-2
default:   -- Keypair: ec2_keypair
default:   -- Block Device Mapping: []
default:   -- Terminate On Shutdown: false
default:   -- Monitoring: false
default:   -- EBS optimized: false
default:   -- Source Destination check:
default:   -- Assigning a public IP address in a VPC: false
default:   -- VPC tenancy specification: default
default: Waiting for instance to become "ready"...
default: Waiting for SSH to become available...
default: Machine is booted and ready for use!
default: Installing rsync to the VM...
default: Rsyncing folder: /home.00/joe/f17/it4200/vagrant/aws_em...
The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Last login: Fri Oct 20 19:35:55 2017 from 144.38.198.24

admin@ip-172-31-43-155:~$
admin@ip-172-31-43-155:~$ echo "Woot. I'm in"
Woot. I'm in
admin@ip-172-31-43-155:~$
vagrant destroy

default: Are you sure you want to destroy the 'default' VM? [y/N] y

>>> default: Terminating the instance...

joe@yavin:~/f17/it4200/vagrant/aws_emi_experiment$
If your machine gets hung on ssh, you may have to allow that through the AWS firewall.
Resources

You are using the following Amazon EC2 resources in the US East (Ohio) region:

<table>
<thead>
<tr>
<th>AMIs</th>
<th>Bundle Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ELASTIC BLOCK STORE</th>
<th>Volumes</th>
<th>Snapshots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volumes</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SECURITY &amp; NETWORK &amp; SECURITY</th>
<th>Security Groups</th>
<th>Elastic IPs</th>
<th>Placement Groups</th>
<th>Key Pairs</th>
<th>Volumes</th>
<th>Load Balancers</th>
<th>Elastic IPs</th>
<th>Snapshots</th>
<th>Security Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Groups</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

Account Attributes

Supported Platforms

Default VPC

vpc-802e14e9

Resource ID length management

Additional Information

Getting Started Guide

Documentation

All EC2 Resources

Forums

Pricing

Contact Us

AWS Marketplace

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Just need a simple virtual private server? Get everything you need to jumpstart your project - compute, storage, and networking – for a low, predictable price. Try Amazon Lightsail for free.
### Security Groups

<table>
<thead>
<tr>
<th>Name</th>
<th>Group ID</th>
<th>Group Name</th>
<th>VPC ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sg-396bfa31</td>
<td>launch-wizard-1</td>
<td>vpc-802e14e9</td>
<td></td>
<td>launch-wizard-1 created</td>
</tr>
<tr>
<td>sg-6733ae0f</td>
<td>Ubuntu 16.04 LTS - Xenial (...</td>
<td>vpc-802e14e9</td>
<td></td>
<td>This security group was</td>
</tr>
<tr>
<td>sg-82901bea</td>
<td>default</td>
<td>vpc-802e14e9</td>
<td></td>
<td>default VPC security group</td>
</tr>
</tbody>
</table>

### Security Group: sg-82901bea

#### Inbound

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All traffic</td>
<td>All</td>
<td>All</td>
<td>0.0.0.0/0</td>
<td></td>
</tr>
</tbody>
</table>
this is my default firewall that will apply to all EC2 instances, you can create a new security group if you want
### Edit inbound rules

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>All traffic</td>
<td>All</td>
<td>0 - 65535</td>
<td>Custom</td>
<td>e.g. SSH for Admin Desktop</td>
</tr>
<tr>
<td>Custom TCP F</td>
<td>TCP</td>
<td>22</td>
<td>Anywhere</td>
<td>e.g. SSH for Admin Desktop</td>
</tr>
</tbody>
</table>

**Add Rule**

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.
joe@yavin:~/f17/it4200/vagrant/aws_emi_experiment$ echo "that rule allowed ssh from anywhere, you probably only want to allow from a certain host or subnet"
joe@yavin:~/f17/it4200/vagrant/aws_emi_experiment$
I deleted my access key after doing these screenshots ;-(

joe@yavin:~/f17/it4200/vagrant/aws_emi_experiment$