Amazon Web Services (AWS) Account Set-up and Start Instance

Fan Wang

2020-09-13

Contents

1 AWS Setup

Go to the **RMD**, **PDF**, or **HTML** version of this file. Go back to Python Code Examples Repository (bookdown site) or the pyfan Package (API).

1.1 Installation on Local Machine

First install anaconda, git, and associated programs.

1. Putty
2. access to .pem key
3. conda aws environment below

1.2 Conda AWS Environment

Can Start and Stop instances from Conda Prompt after this.

```
conda deactivate
conda list env
conda env remove -n wk_aws
conda create -n wk_aws -y
conda activate wk_aws

# Install External Tools
conda install -c anaconda pip -y

# Command line interface
conda install -c conda-forge awscli -y
# Programmatically send JSON instructions with boto3
conda install -c anaconda boto3 -y
```

1.3 AWS Account set-up

1. Sign-up for AWS web services account (can be the same as your Amazon shopping account)
2. Register for AWS Educate to get student or faculty voucher.

- The University of Houston is a part of AWS Educate, choose educator or student, should hear back within 24 hours with coupon code.
- UH students can get $100, faculty can get $200.

1.4 Start a AWS Instance and Link Local to Remote

Amazon has a lot of tutorials. Here is an outline.

1. Generate keypair on AWS, aws guide
   - this gives you a .pem file which you download and Amazon also remembers
   - local computers with the right .pem file can talk to your AWS instances
   - You might need to invoke the chmod command below to set permission:

```
chmod 400 "C:/Users/fan/Documents/Dropbox (UH-ECON)/Programming/AWS/fan_wang-key-pair-us_east_nv.pe
```

2. *Launching Instance*: Go to your console, choose EC2, choose launch instance, select Amazon Linux Instance (review and launch)
3. *Instance security*: select VPC security group: I have for example: fan_wang_SG_us_east_nv_VPC (edit security group and submit)
   - Security group can allow any IP address to access your instance or just specific ones.
   - AWS has a tool here that just allows your current IP to access the EC2 instance
4. *Instance access key*: Select right keypair (your .pem key), fan_wang-key-pair-us_east_nv (prompted after submitting)
5. For SSH in, you can use Putty. aws guide
   - tell Putty your AWS instance DNS address and where your pem key is
   - Can use a Putty client to enter an EC2 instance
6. For SSH, can also do the process below:
   - open git bash (install putty before)

```
ssh-agent -s
eval $(ssh-agent -s)
```

   - Tell SSH where pem key is:

```
ssh-add "C:/Users/fan/Documents/Dropbox (UH-ECON)/Programming/AWS/fan_wang-key-pair-us_east_nv.pem"
```

   - You will find a public DNS address for your aws instance on the AWS user interface page

```
# ssh git bash command line
# for ubuntu machine
ssh ubuntu@ec2-54-197-6-153.compute-1.amazonaws.com
# for aws linux
ssh ec2-user@ec2-52-23-218-117.compute-1.amazonaws.com
# quit aws instance
# ctrl + D
```

   - if get: Permission denied (publickey), see:
     1. Trying to connect with the wrong key. Are you sure this instance is using this keypair?
     2. Trying to connect with the wrong username. ubuntu is the username for the ubuntu based AWS distribution, but on some others it's ec2-user (or admin on some Debians, according to Bogdan Kulbida's answer)(can also be root, fedora, see below)
     3. Trying to connect the wrong host. Is that the right host you are trying to log in to?
   - You can log in generally like this, note the instance gets new public DNS IP address every time you restart it:

```
LOCALPEM="C:/Users/fan/Documents/Dropbox (UH-ECON)/Programming/AWS/fan_wang-key-pair-us_east_nv.pem
IPADD=34.207.250.160
REMOTEIP=ec2-user@$IPADD
ssh-keygen -R $IPADD
ssh -i "$LOCALPEM" $REMOTEIP
```

1.5 Use AWSCLI to Start and Stop an Instance

1. Install AWS CLI
2. Create individual IAM users
3. Follow instructions to Configure your awscli, and profit access key id and secrete access key when prompted.
   - do not copy and paste the Key ID and Access Key. They are example, type these in as answers given config prompt:

```
# aws configure
AWS Access Key ID [None]: XXXXIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wXalrXXtnXXXX/X7XXXXX/bXxXfiCXXXXXXXXXXX
Default region name [None]: us-west-1
Default output format [None]: json
```

   - this creates under a folder like this: C:/Users/fan/.aws, inside the folder these info will be stored in a configuration file.

```
# the credentials file
[default]
aws_access_key_id = XXXXIOSFODNN7EXAMPLE
aws_secret_access_key = wXalrXXtnXXXXX7XXXXXbXxXfiCXXXXXXXXXX
```

   - then when you use aws cli, you will automatically be authenticated
4. Start an instance in console first (or directly in command line). Stop it. do not terminate. Now this instance will have a fixed instance ID. Its DNS IP address will change every time you restart it, but its instance ID is fixed. Instance ID is found easily in the EC2 Console.
   - Launch an instance

```
aws ec2 run-instances --image-id ami-xxxxxxxx --count 1 --instance-type t2.micro --key-name MyKeyPa
```

   - Start an instance

```
aws ec2 start-instances --instance-ids i-XXXXXXXX
aws ec2 start-instances --instance-ids i-040c856530b2619bc
```

   - Stop an instance

```
aws ec2 stop-instances --instance-ids i-XXXXXXXX
aws ec2 stop-instances --instance-ids i-040c856530b2619bc
```

1.6 Set-up SSM on EC2 Instance

To execute commandlines etc remote on EC2, need to set up SSM: AWS Systems Manager Agent (SSM Agent)

SSM-agent is already installed in Amazon Linux.

Error Message regarding InvalidInstanceId. The following scenarios can result in this error message:

- Instance id is invalid (in the comments you have verified it isn't)
- Instance is in a different region (in the comments you have verified it isn't)
- Instance is not currently in the Running state
- Instance does not have the AWS SSM agent installed and running.

"You have to create and attach the policy AmazonSSMFullAccess to the machine (thats maybe more broad than you need) but that was why it wasn't working for me... You do that by clicking on (when selected on the ec2 instance) Action > Instance Settings > Attach/Replace IAM Role then create a role for ec2 that has that permission then attach, should take like 5-10 mins to pop up in SYSTEMS MANAGER SHARED RESOURCES - Managed Instances as mark mentions. – Glen Thompson Sep 20 '18 at 16:31"

```
# Start SSM Agent with
sudo systemctl start amazon-ssm-agent
```