AWS S3 & EC2

5/29/2017

galvanize

- AWS Overview
- S3
- EC2

Some Advice...

We are about to use some technologies that have a semi steep learning curve for some. Create a big data reference folder on your machine. In it, keep:

- Screen shots or files containing common actions (copying a file to an instance)
- Instructions of common workflows (setting up an EC2, EMR, etc.)
- Problems encountered and how you solved them
- GitHub repos that have useful materials (feel free to fork: https://github.com/brent-lemieux/aws_scripts, https://github.com/ewellinger/spark-talk, etc.)

Morning Objectives

galvanize

- Overview of Cloud Computing - we use Amazon Web Services (AWS)
- AWS Setup Items
  - AWS Command Line Interface
  - SSH folder / config file
- AWS Basic Workflow Established
  - Elastic Compute Cloud (EC2)
  - Simple Storage Service (S3)

Server farm with
EC2 Instance

Jobs

Clients

Cloud Computing - Pros/Cons

- Accessibility from anywhere
- Can scale easily
- Storage
- Maintenance
- *Cost* *

- Security - a third party owns the server
- *Cost* *

* Depending on your size and needs cost can be a pro or a con

galvanize

Why AWS?

- Many instance types
- A lot of traction over the years
- LIbraries and tools built around it
- Behemoth in the cloud computing sector
    - ~30% market share

Morning Objectives

galvanize

- ✅ Overview of Cloud Computing - we use Amazon Web Services (AWS)
- AWS Setup Items
  - AWS Command Line Interface
  - SSH folder / config file
- AWS Basic Workflow Established
  - Elastic Compute Cloud (EC2)
  - Simple Storage Service (S3)

AWS Command Line Interface

$ pip install awscli --upgrade --user

$ aws configure

- Enter AWS Access Key ID
- Enter AWS Secret Key
- Enter region ----> us-east-1
- Press enter

.ssh folder and config file

- Check to see if you have a **.ssh** folder in your home directory
- If not, make one

- Check to see if you have a **config** file in your .ssh folder
- If not, make one

Keep the config file open, this will save us a lot of time when we start using EC2

Morning Objectives

- ✓ Overview of Cloud Computing - we use Amazon Web Services (AWS)
- ✓ AWS Setup Items
  - AWS Command Line Interface
  - SSH folder / config file
- AWS Basic Concepts and Workflow Established
  - Elastic Compute Cloud (EC2)
  - Simple Storage Service (S3)

EC2 - (Elastic Cloud Compute)

- Virtual machine in the cloud
- Can choose type of machine to launch
  - Ubuntu (Linux)
  - RedHat (Linux)
  - … and many more
- We use our local terminal as the client to access the remote machine

Step 1: Amazon Machine Image

This is where we specify the type of Machine Image we want to launch. We can choose one of the default AMIs (Amazon Machine Images) or we can choose from the Community AMIs.

- Community AMIs are essentially templates with packages pre-installed (these may need to be updated)
- **Today**, let's use the **DSI-Template3** (only available in the N. Virginia Region)
- In the future, you may want access to a GPU… Use **DSI-DeepLearning3**

Step 2: Choose an Instance Type

This is where we will choose what kind of resources our machine will have including the number of CPUs, GPUs, and Memory. Each instance will have a varying hourly cost associated with it that fluctuates with current demand.

As a rough guideline, a m3.xlarge instance which has 4 CPUs and 15 GB of memory costs approximately $0.25 - $0.30 per hour.

galvanize

Step 3: Configuration

- Spot instances are available to reduce price starting from m3.medium
- Bid the machine with the price you set (as max)
- Takes longer to start
- Cannot stop and restart instance
- Much cheaper in general
- Otherwise expensive to use larger instances

Step 4: Storage

- By default we will have a root EBS volume (more on this later) for storing OS and other files
- We can choose to up this or add on other hard drives

galvanize

Step 5: Tagging

- First Key is by default “Name”
- This controls the name that shows up in the Instances section of your EC2 dashboard
- All this does is help you keep track of which machine is which

Step 6: Security Group

- Here we will configure the security settings of our instance
- For example we could configure our instance to only accept ssh connections from our home or work IP Addresses

galvanize

Creating/Using .pem files

In order to access our machine through ssh we need to have a pem file which is essentially a cryptographic key.

CAUTION: Make sure you have the .pem file locally, otherwise create a new key pair

$ mv key_pair.pem ~/.ssh/

$ chmod 400 key_pair.pem

Accessing Instance

```
# Example for lecture
Host my_instance_name
    Hostname ec2-53-205-213-56.compute-1.amazonaws.com
    User ubuntu
    IdentityFile ~/.ssh/key_pair.pem
```

Add to **config** file in .ssh folder

Note: sub User for whatever OS you chose and Hostname for your instances domain

This instance could be accessed by:

$ ssh my_instance_name

Or…

$ ssh -i key_pair.pem ubuntu@ec2-53-205-213-56.compute-1.amazonaws.com

galvanize

Copying files and folders to instance

In the terminal on your local machine:

$ scp -i key_pair.pem file.txt User@Domain:/path/where/file/should/go/

$ scp -i key_pair.pem -r folder User@Domain:/path/where/folder/should/go/

User = ubuntu (or other OS)

Domain = IP Address (i.e. ec2-53-205-213-56.compute-1.amazonaws.com)

galvanize

Running Scripts and Multiplexers

Check out screens -- easy to use multiplexer:

- Allows us to basically have multiple terminal windows open on instance
- We can still use our instance while it is running scripts
- We are going to use **screen** (**tmux** is a more powerful, but harder to use multiplexer)

galvanize

screen

Install: $ sudo apt-get install screen

Create a session called my-session: $ screen -S my-session

Detach the session: $ ctrl + a + d

List the running sessions: $ screen -ls

Reattach a specific session: $ screen -R my-session

screen - What is it good for?

Allows us to use our remote machine while it simultaneously:

- Trains a model
- Hosts a mongo/SQL database
- Hosts a web app
- Runs a jupyter notebook (need a special script to do this)
- etc.

Stopping/Terminate Instances

Make sure you do this!!!

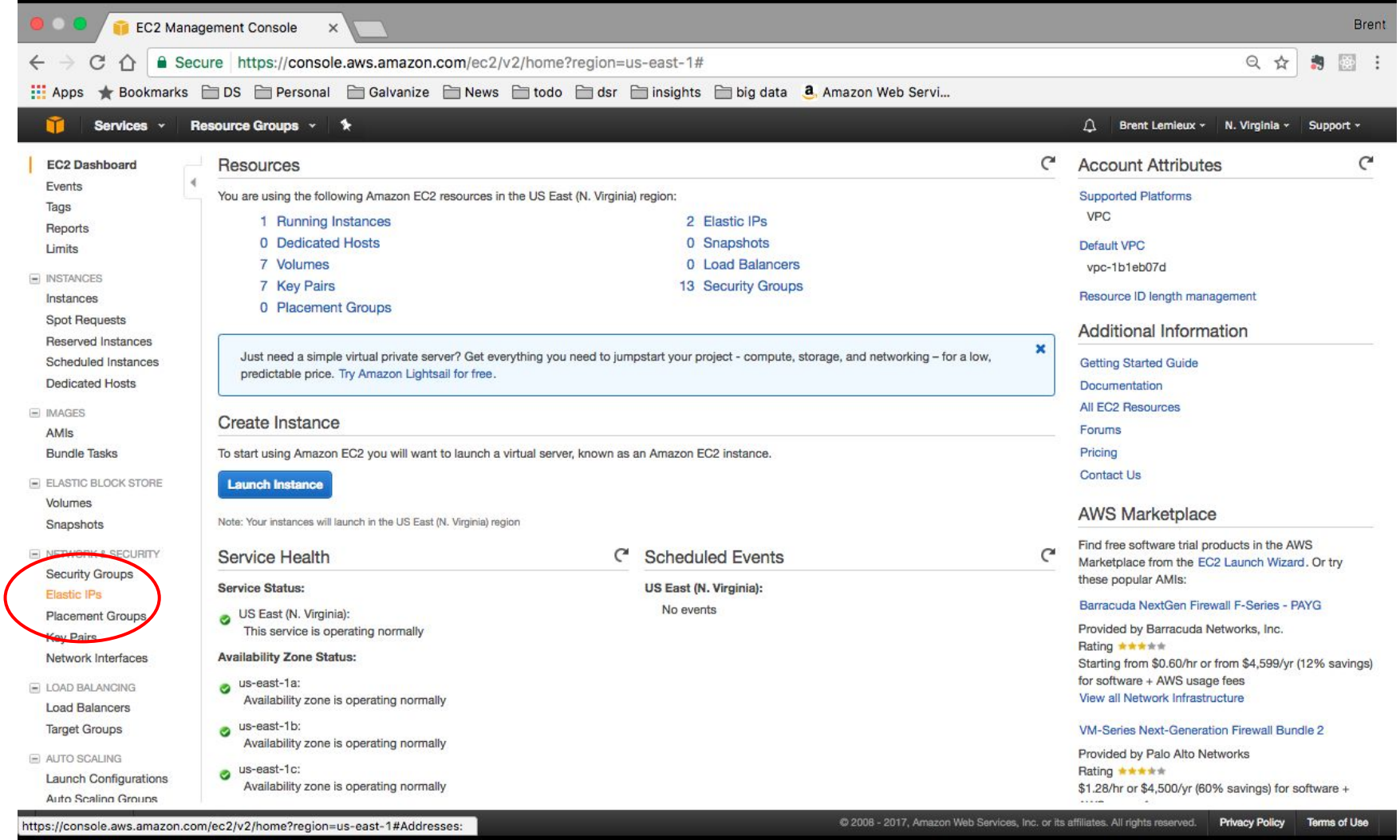
You can do this via the AWS GUI or using AWS Command Line Interface

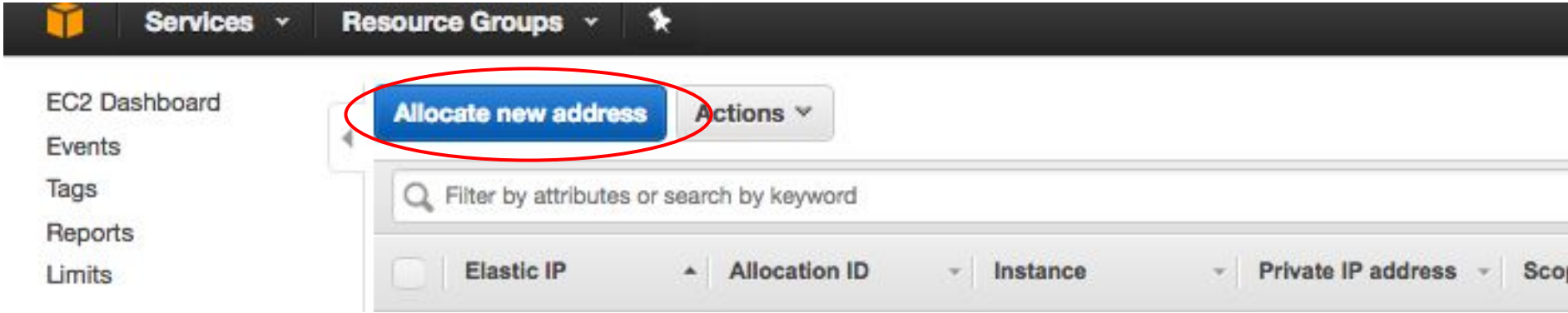
Even if you are running on the free tier, be aware that there are usage limits dictating when a free tier becomes paid

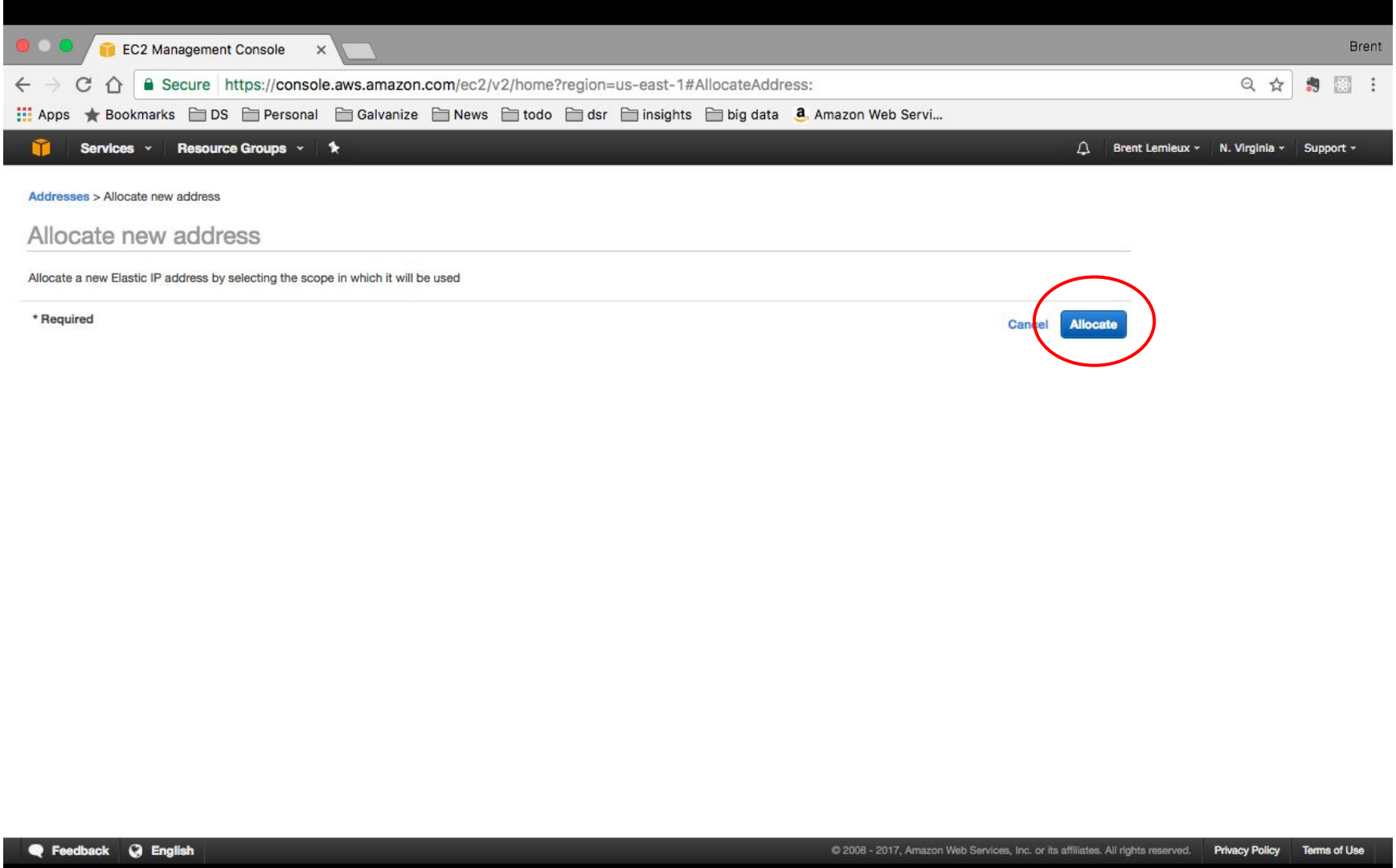
AWS Elastic IP Address

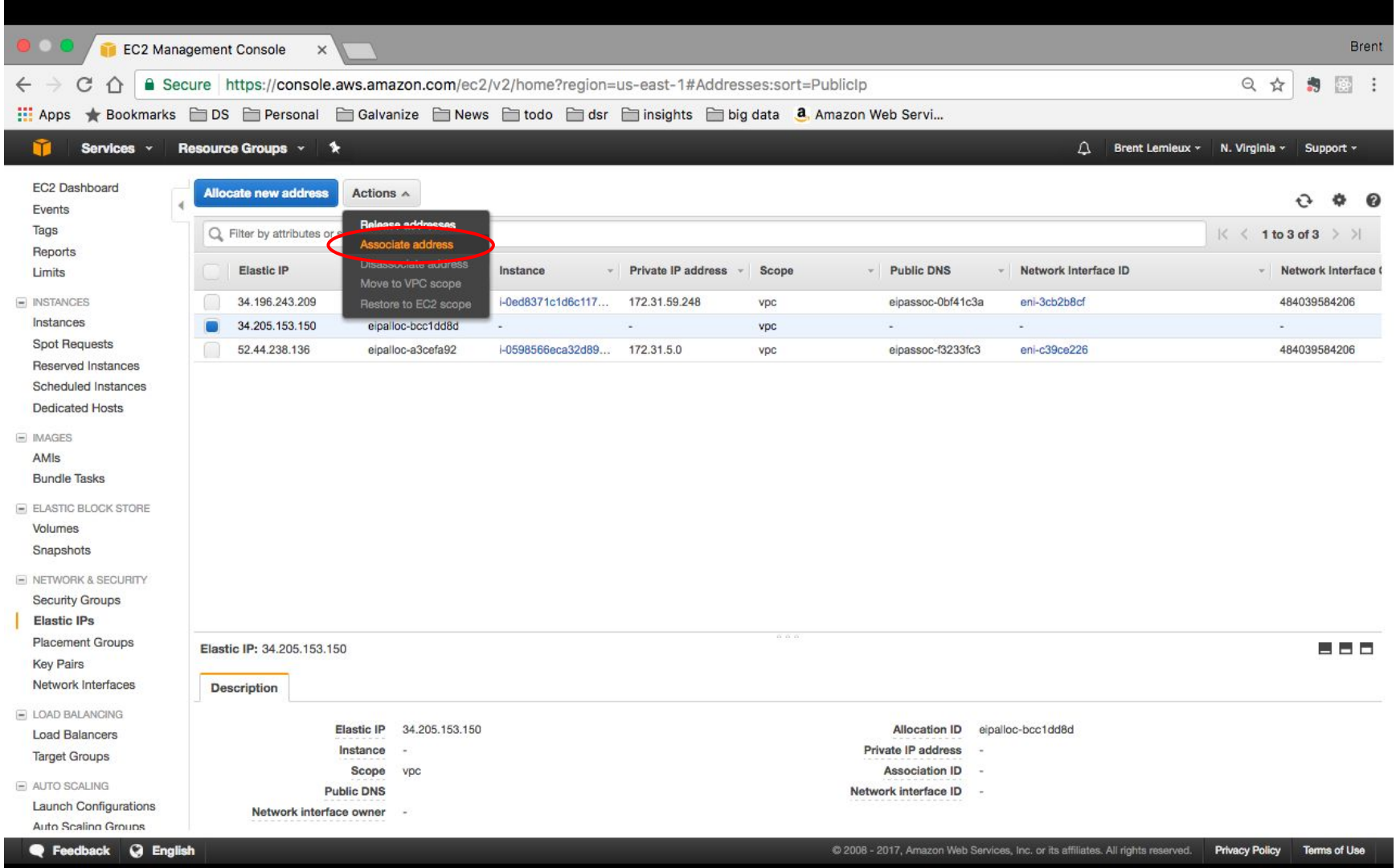
One problem with stopping and starting instances is that the IP Address changes each time… This means we have to edit our **~/.ssh/config** file each time.

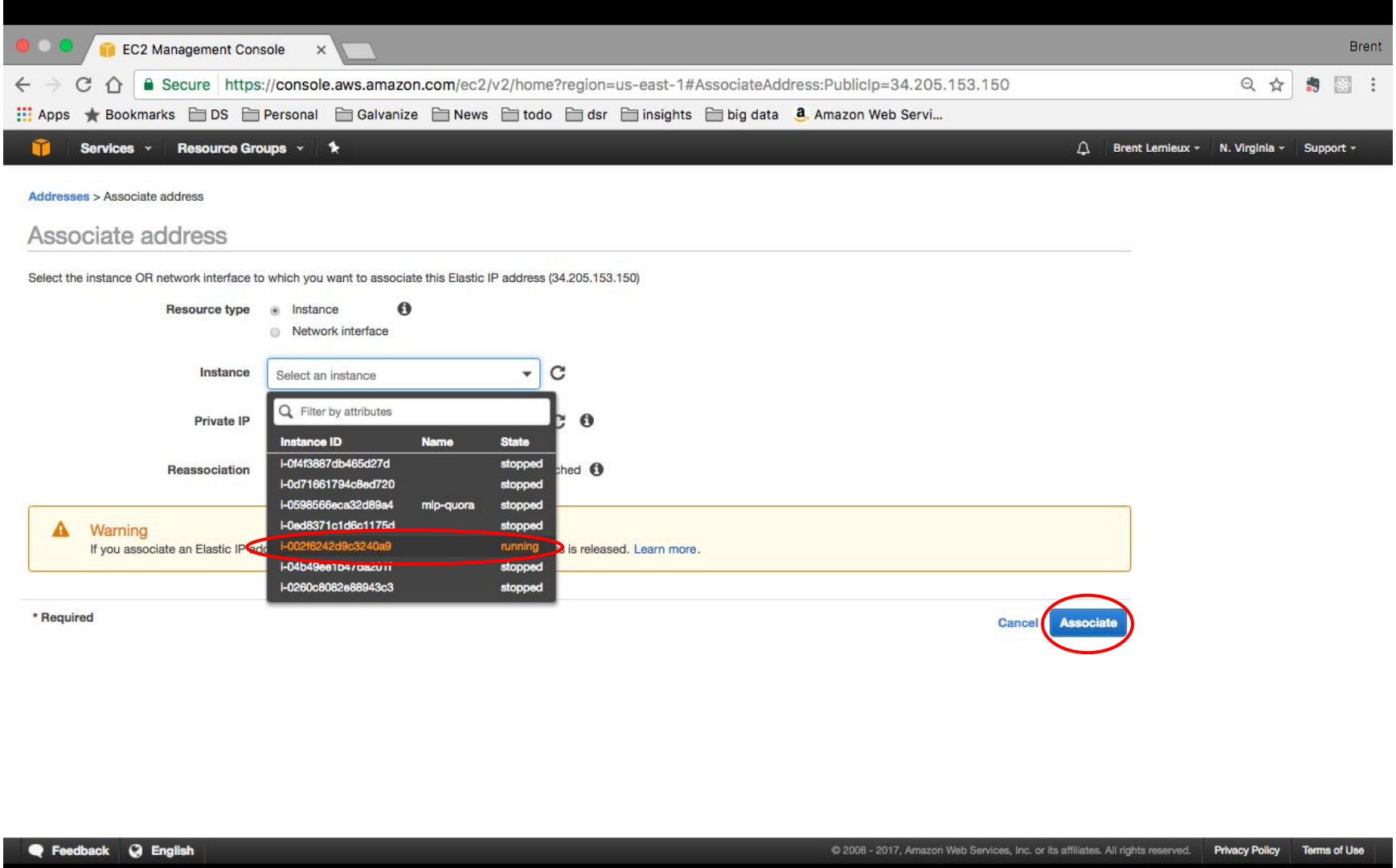
Elastic IPS are our way around this


EC2 Management Console
Brent
Secure https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#
Apps Bookmarks DS Personal Galvanize News todo dsr insights big data Amazon Web Servi...
Services Resource Groups
Brent Lemieux N. Virginia Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Spot Requests
Reserved Instances
Scheduled Instances
Dedicated Hosts
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
LOAD BALANCING
Load Balancers
Target Groups
AUTO SCALING
Launch Configurations
Auto Scaling Groups
Resources
You are using the following Amazon EC2 resources in the US East (N. Virginia) region:
1 Running Instances
0 Dedicated Hosts
7 Volumes
7 Key Pairs
0 Placement Groups
2 Elastic IPs
0 Snapshots
0 Load Balancers
13 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.
Launch Instance
Note: Your instances will launch in the US East (N. Virginia) region
Service Health
Service Status:
US East (N. Virginia):
This service is operating normally
Availability Zone Status:
us-east-1a:
Availability zone is operating normally
us-east-1b:
Availability zone is operating normally
us-east-1c:
Availability zone is operating normally
Scheduled Events
US East (N. Virginia):
No events
Account Attributes
Supported Platforms
VPC
Default VPC
vpc-1b1eb07d
Resource ID length management
Additional Information
Getting Started Guide
Documentation
All EC2 Resources
Forums
Pricing
Contact Us
AWS Marketplace
Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try these popular AMIs:
Barracuda NextGen Firewall F-Series - PAYG
Provided by Barracuda Networks, Inc.
Rating
Starting from $0.60/hr or from $4,599/yr (12% savings) for software + AWS usage fees
View all Network Infrastructure
VM-Series Next-Generation Firewall Bundle 2
Provided by Palo Alto Networks
Rating
$1.28/hr or $4,500/yr (60% savings) for software +
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Addresses:
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Services
Resource Groups
EC2 Dashboard
Events
Tags
Reports
Limits
Allocate new address
Actions
Filter by attributes or search by keyword
Elastic IP
Allocation ID
Instance
Private IP address


EC2 Management Console
Brent
Secure
https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#AllocateAddress:
Apps
Bookmarks
DS
Personal
Galvanize
News
todo
dsr
insights
big data
Amazon Web Servi...
Services
Resource Groups
Brent Lemieux
N. Virginia
Support
Addresses > Allocate new address
Allocate new address
Allocate a new Elastic IP address by selecting the scope in which it will be used
* Required
Cancel
Allocate
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


EC2 Management Console
Brent
Secure https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Addresses:sort=PublicIp
Apps Bookmarks DS Personal Galvanize News todo dsr insights big data Amazon Web Servi...
Services Resource Groups
Brent Lemieux N. Virginia Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Spot Requests
Reserved Instances
Scheduled Instances
Dedicated Hosts
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
LOAD BALANCING
Load Balancers
Target Groups
AUTO SCALING
Launch Configurations
Auto Scaling Groups
Allocate new address
Actions
Release addresses
Associate address
Disassociate address
Move to VPC scope
Restore to EC2 scope
Filter by attributes or search by keyword
1 to 3 of 3
Elastic IP
Instance
Private IP address
Scope
Public DNS
Network Interface ID
Network Interface
34.196.243.209 i-0ed8371c1d6c117... 172.31.59.248 vpc eipassoc-0bf41c3a eni-3cb2b8cf 484039584206
34.205.153.150 eipalloc-bcc1dd8d - - vpc - - -
52.44.238.136 eipalloc-a3cefa92 i-0598566eca32d89... 172.31.5.0 vpc eipassoc-f3233fc3 eni-c39ce226 484039584206
Elastic IP: 34.205.153.150
Description
Elastic IP 34.205.153.150
Instance -
Scope vpc
Public DNS
Network interface owner -
Allocation ID eipalloc-bcc1dd8d
Private IP address -
Association ID -
Network interface ID -
Feedback English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


EC2 Management Console
Brent
Secure https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#AssociateAddress:PublicIp=34.205.153.150
Apps Bookmarks DS Personal Galvanize News todo dsr insights big data Amazon Web Servi...
Services Resource Groups
Brent Lemieux N. Virginia Support
Addresses > Associate address
Associate address
Select the instance OR network interface to which you want to associate this Elastic IP address (34.205.153.150)
Resource type
Instance
Network interface
Instance
Select an instance
Private IP
Reassociation
Filter by attributes
Instance ID Name State
i-0f4f3887db465d27d stopped
i-0d71661794c8ed720 stopped
i-0598566eca32d89a4 mlp-quora stopped
i-0ed8371c1d6c1175d stopped
i-002f6242d9c3240a9 running
i-04b49ee1b47da201f stopped
i-0260c8082e88943c3 stopped
Warning
If you associate an Elastic IP
is released. Learn more.
* Required
Cancel
Associate
Feedback
English
© 2008 - 2017, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

galvanize

Elastic IP Address - Last Step

Set the new domain or IP Address in your **.config** file…

Now you can access this instance easily from the command line without changing anything

galvanize

AWS CLI

Start new instance...

```
$ aws ec2 run-instances --image-id ami-xxxxxxxx --count 1 --instance-type t1.micro --key-name MyKeyPair --security-groups my-sg
```

Start existing instance...

```
$ aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

Stop existing instance...

```
$ aws ec2 stop-instances --instance-ids i-1234567890abcdef0
```

S3 (Simple Storage Service)

- Storage space for big data
- S3 Storage –> Permanent and big data
- EBS (Elastic Block) Storage –> Temporary and small data
- Operations:
  - Upload
  - Download
  - Read
  - Write

S3 Cost

- Very cheap
- First 1TB per month is approximately $0.03 per GB

S3 Buckets

- Where your data is stored
- Can contain folders and files
- Permissions can be set to dictate access for other users
  - If data is not sensitive, you can set **read** privileges for public
  - *CAUTION: Don't give public write permissions!!!*

S3 Bucket Names

- Bucket name must be unique -- no one has used it before
- Must be all lowercase
- No underscores allowed (use dashes instead)

Buckets Uploading/Downloading Data

Can be done through AWS GUI -- but this is annoying.

Use Python instead!

- Pandas
- Boto3
  - http://boto3.readthedocs.io/en/latest/guide/migrations3.html
  - https://github.com/brent-lemieux/aws_scripts - some scripts to get you started

galvanize

Pandas S3

Pandas allows us to read files easily… Don’t forget to configure AWS CLI first...

```
import pandas as pd

df = pd.read_csv('s3a://my_bucket/my_file.csv')

# Or, if the data set is too large, use the chunksize argument

data_chunks = pd.read_csv('s3a://my_bucket/my_file.csv', chunksize=50000)
df_chunk = data_chunks.get_chunk()
```

galvanize

Boto3

$ pip install boto3

Boto3 - Allows us to manage our buckets… More functionality than Pandas.

https://github.com/brent-lemieux/aws_scripts

galvanize

When to use S3/EC2

- When data is too big to fit locally
- Scripts that take a long time to run
- Have to run continuously (Hosting Web Apps)

galvanize

Workflow

1. Upload/Access data on S3
2. Use pandas/boto to pull a subset down to local
3. Develop script locally
4. Upload script to EC2
5. Run script on EC2 on full set
6. Write results to S3

galvanize

Assignment

Use boto3 instead of boto!