# AWS Cloud for Atmospheric Scientists

Fei Yao

School of GeoSciences

The University of Edinburgh

https://www.geos.ed.ac.uk/homes/s1855106/

# Contents

- What is Cloud Computing?
- What is AWS Cloud and Why?
- Access & Use AWS Cloud with a Joint Account
  - Basic usage
  - Advanced topics
- Summary & Prospects

# Contents

- What is Cloud Computing?
- What is AWS Cloud and Why?
- Access & Use AWS Cloud with a Joint Account
  - Basic usage
  - Advanced topics
- Summary & Prospects

# What is Cloud Computing?

Cloud Computing is on-demand delivery of IT resources and applications via the Internet with pay-as-you-go pricing.

- Usually costs, but always cost-effective
- Elastic and scalable (dynamically adjust hardware as needed)
- Replication, replication, replication!
- Globally available in seconds

# Contents

- What is Cloud Computing?
- What is AWS Cloud and Why?
- Access & Use AWS Cloud with a Joint Account
  - Basic usage
  - Advanced topics
- Summary & Prospects

# What is AWS Cloud and Why?

## Cloud Computing with Amazon Web Services

Trade capital expense for variable expense

Increase speed and agility

Benefit from massive economies of scale

Stop spending money on running and maintaining data centers

Stop guessing capacity

Go global in minutes

# What is AWS Cloud and Why?

All services

**Compute**
- EC2
- Lightsail
- ECR
- ECS
- EKS
- Lambda
- Batch
- Elastic Beanstalk

**Storage**
- S3
- EFS
- FSx
- S3 Glacier
- Storage Gateway
- AWS Backup

**Security, Identity, & Compliance**
- IAM
- Resource Access Manager
- Cognito
- Secrets Manager
- GuardDuty
- Inspector
- Amazon Macie
- AWS Organizations
- AWS Single Sign-On
- Certificate Manager
- Key Management Service

Identity and Access Management

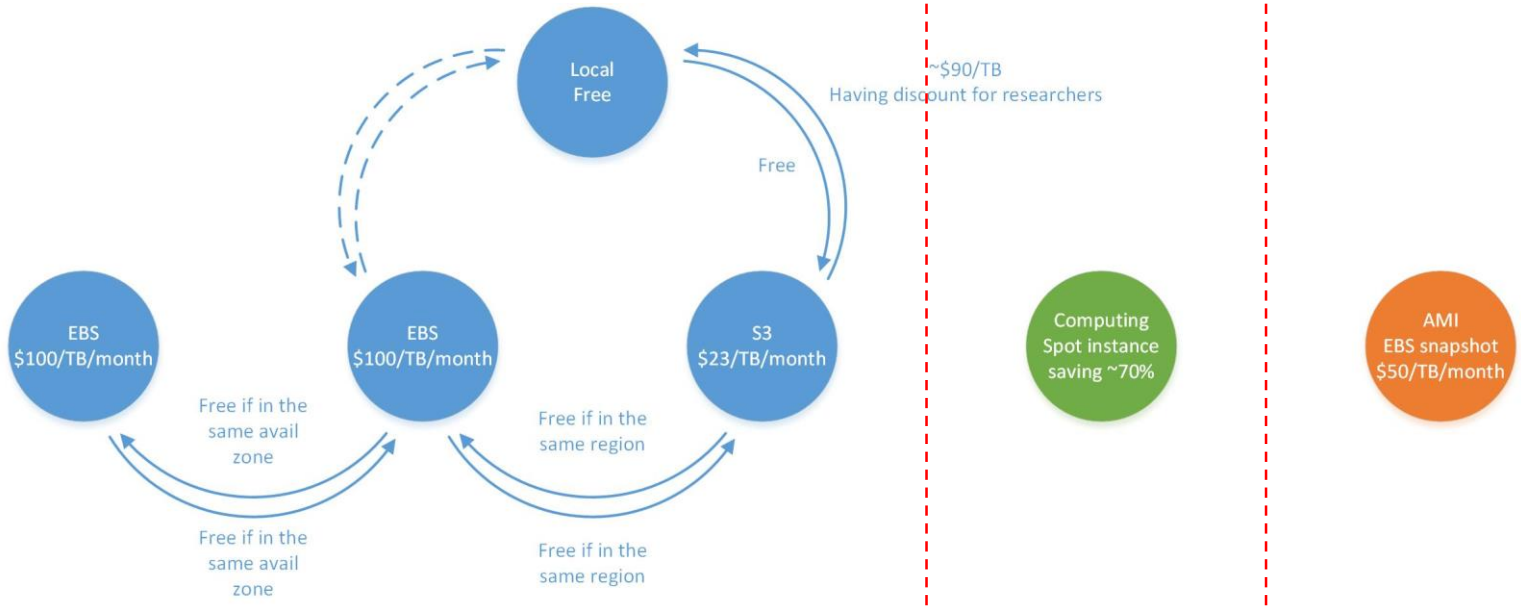
Elastic Compute Cloud

Simple Storage Service

# What is AWS Cloud and Why?

## Cloud Computing with Amazon Web Services

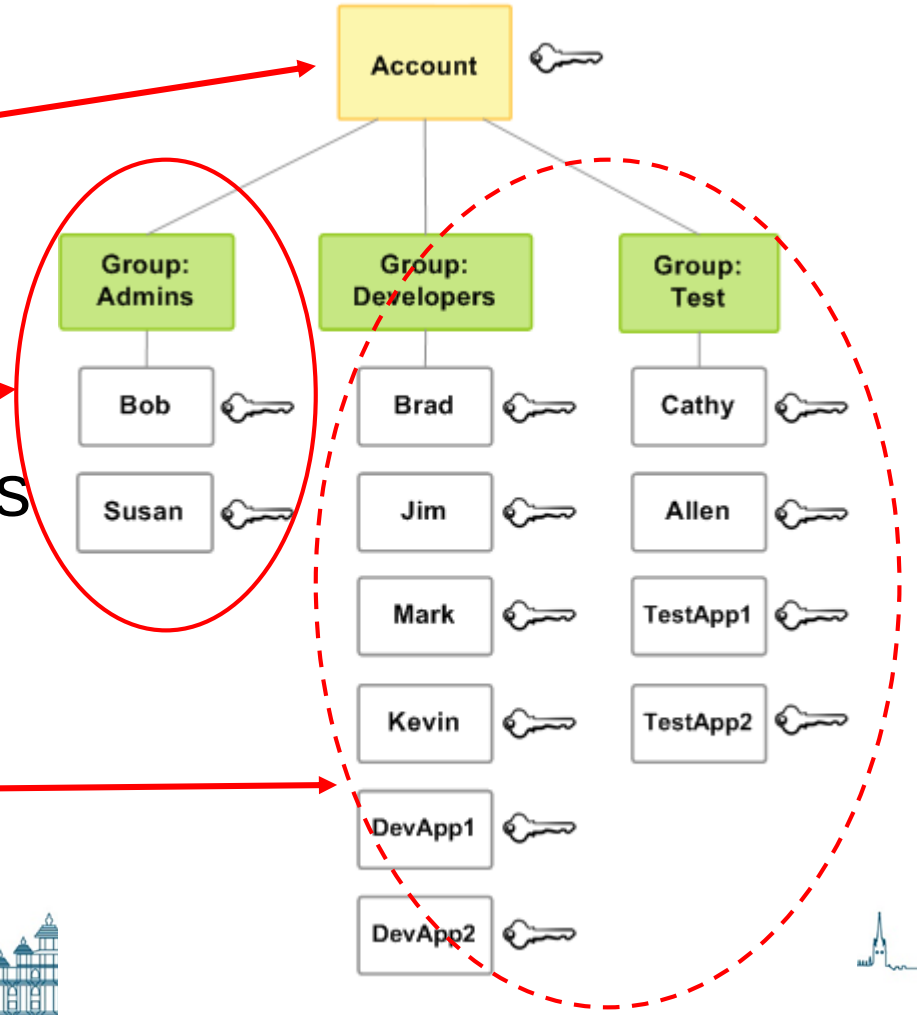
- Pricing management



# What is AWS Cloud and Why?

## Group AWS Management via IAM

- Root user has complete permissions
- Admins have almost complete permissions except for billing and credit
- Coming in the future, hopefully

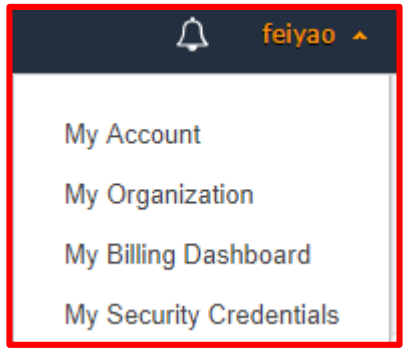


# Contents

- What is Cloud Computing?
- What is AWS Cloud and Why?
- Access & Use AWS Cloud with a Joint Account
  - Basic usage
  - Advanced topics
- Summary & Prospects

# Access & Use AWS Cloud with a joint account – Basic Usage

## Sign In to the Console as root user




- Console login link

  https://signin.aws.amazon.com/console

- User name

- Password

# Access & Use AWS Cloud with a joint account – Basic Usage

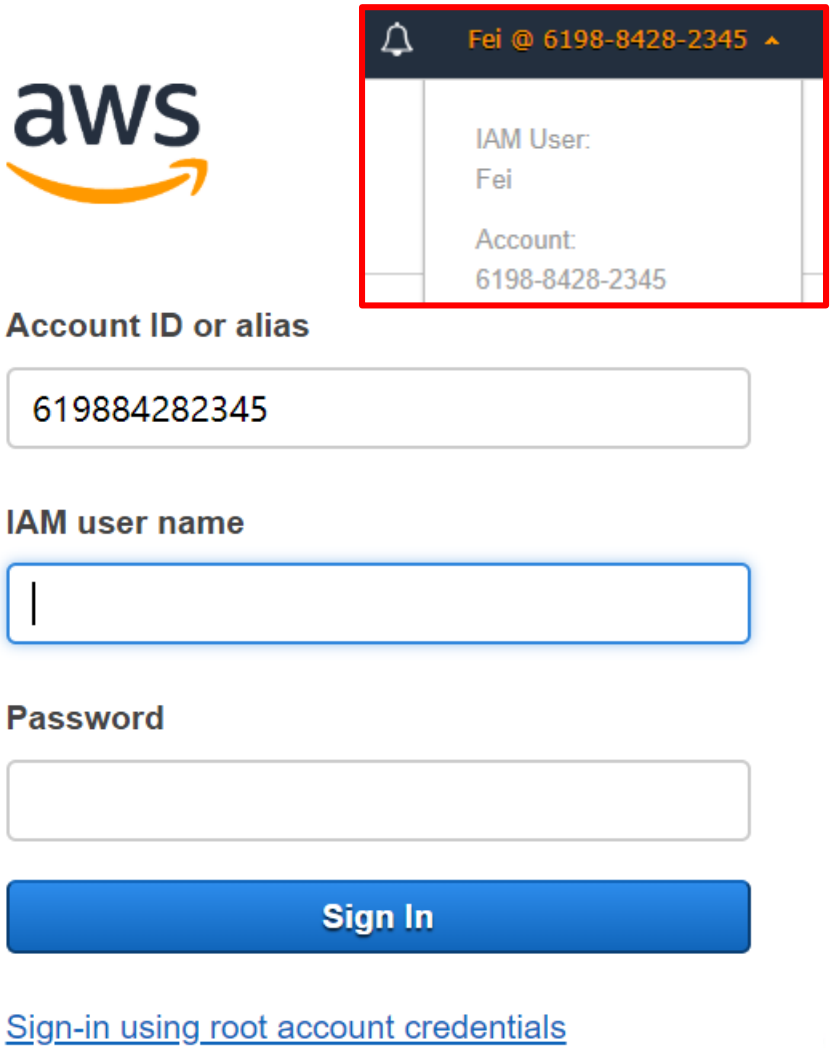
## Sign In to the Console as an IAM user

- Console login link

  https://619884282345.signin.aws.amazon.com/console

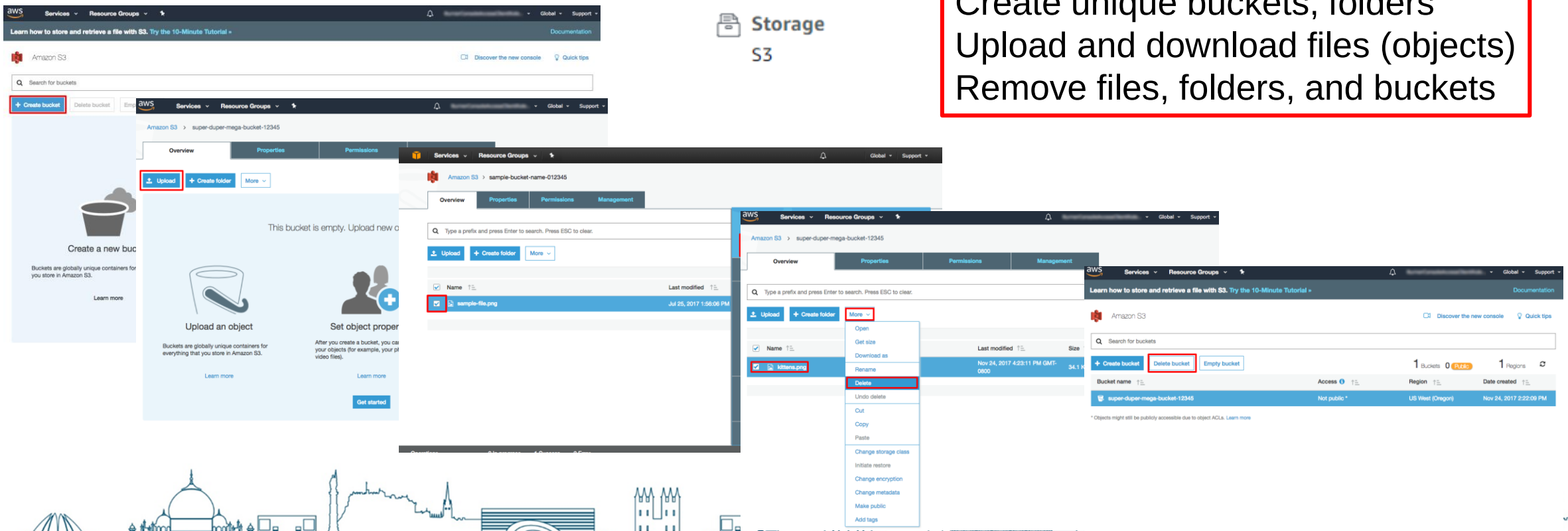
- User name

- Password



# Access & Use AWS Cloud with a joint account – Basic Usage – S3

## Manipulate S3 in Graphical Console



# Access & Use AWS Cloud with a joint account – Basic Usage – EC2

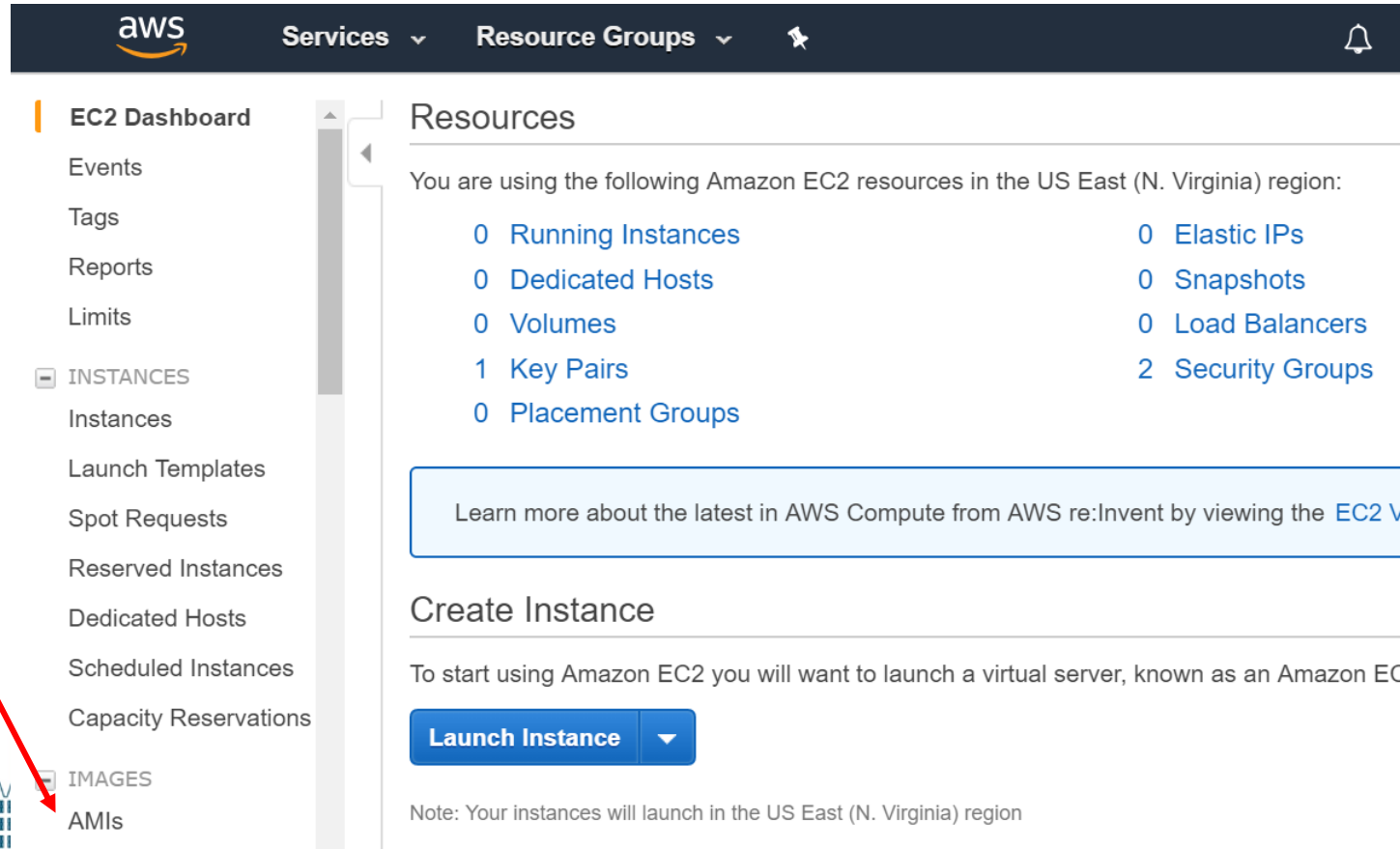
## Step 1. Switch to EC2 Dashboard

Compute

EC2

Click

Amazon Machine Image specifies the software side of an EC2 instance (computer). By replicating it, you avoid installing operating system, software, libraries, etc.



# Access & Use AWS Cloud with a joint account – Basic Usage – EC2

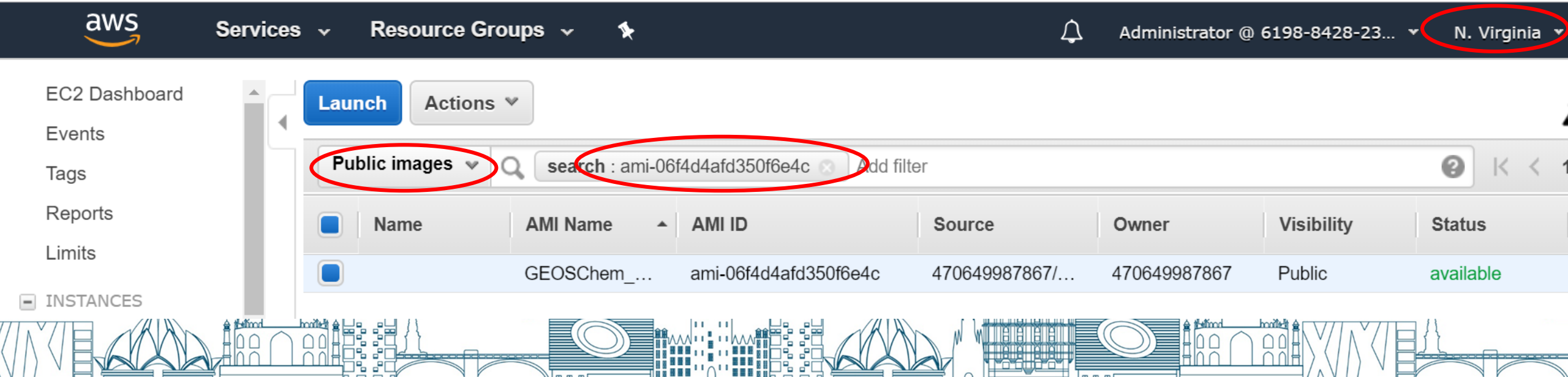
IMAGES

AMIs

## Step 2. Search for a specific AMI such as GEOS-Chem

- Search for ami-06f4d4afd350f6e4c from Public images in the US East (N. Virginia) region – that's the system with both the classic and the High-Performance versions of GEOS-Chem installed.

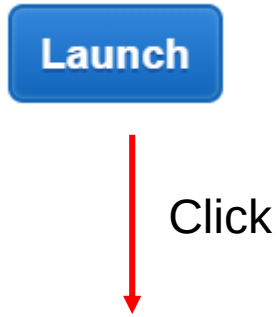
Click



# Access & Use AWS Cloud with a joint account – Basic Usage – EC2

## Step 3. Launch from that AMI and choose an Instance Type

- Instance Type specifies hardware side of an EC2 instance, mostly about CPUs.
- For longer-term, higher-resolution runs, consider bigger ones like c5.9xlarge and c5.18xlarge.



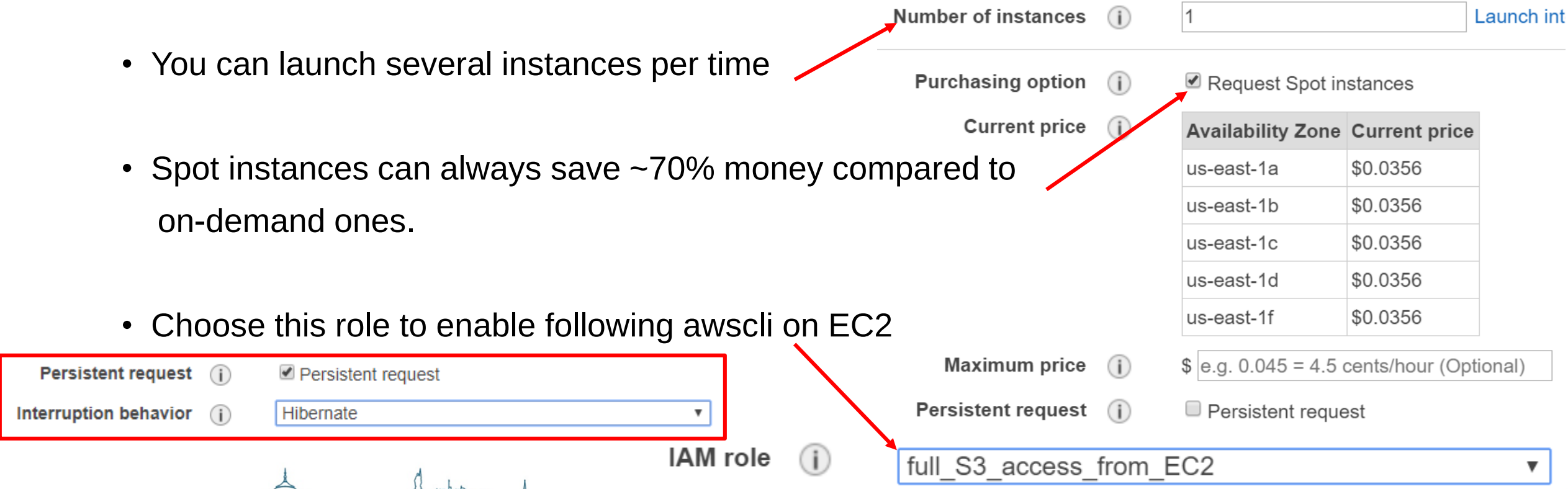

**Currently selected:** r5.large (10 ECUs, 2 vCPUs, 2.5 GHz, Intel Xeon Platinum 8175, 16 GiB memory, EBS only)

| | Family | Type | vCPUs | Memory (GiB) | Instance Storage (GB) | EBS-Optimized Available | Network Performance | IPv6 Support |
|---|---|---|---|---|---|---|---|---|
| ☑ | Memory optimized | r5.large | 2 | 16 | EBS only | Yes | Up to 10 Gigabit | Yes |

# Access & Use AWS Cloud with a joint account – Basic Usage – EC2

## Step 4. Configure Instance Details

- You can launch several instances per time
- Spot instances can always save ~70% money compared to on-demand ones.
- Choose this role to enable following awscli on EC2

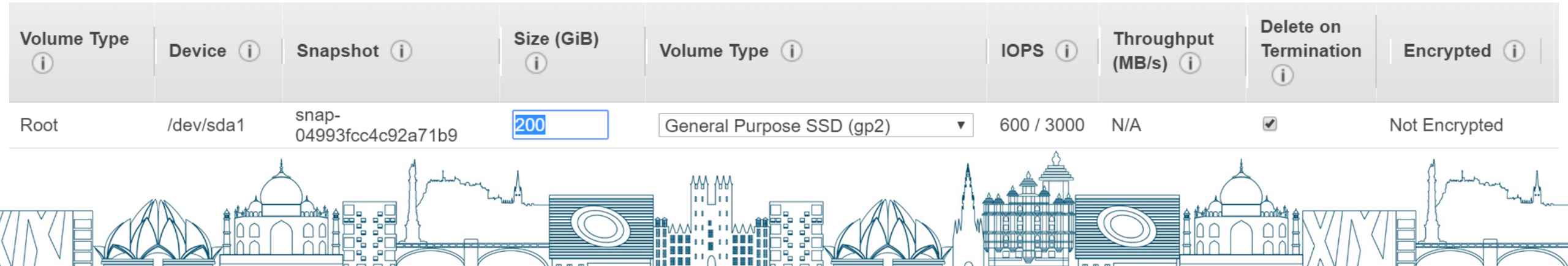


# Access & Use AWS Cloud with a joint account – Basic Usage – EC2

## Step 5. Add Storage

- The volume attached to EC2 instance is called Elastic Block Storage (EBS) and can be used by EC2 instance directly. S3 is independent of any EC2 instances. AWS provides a series of commands to help transfer data between S3 and EC2 very efficiently (>100MiB/s).

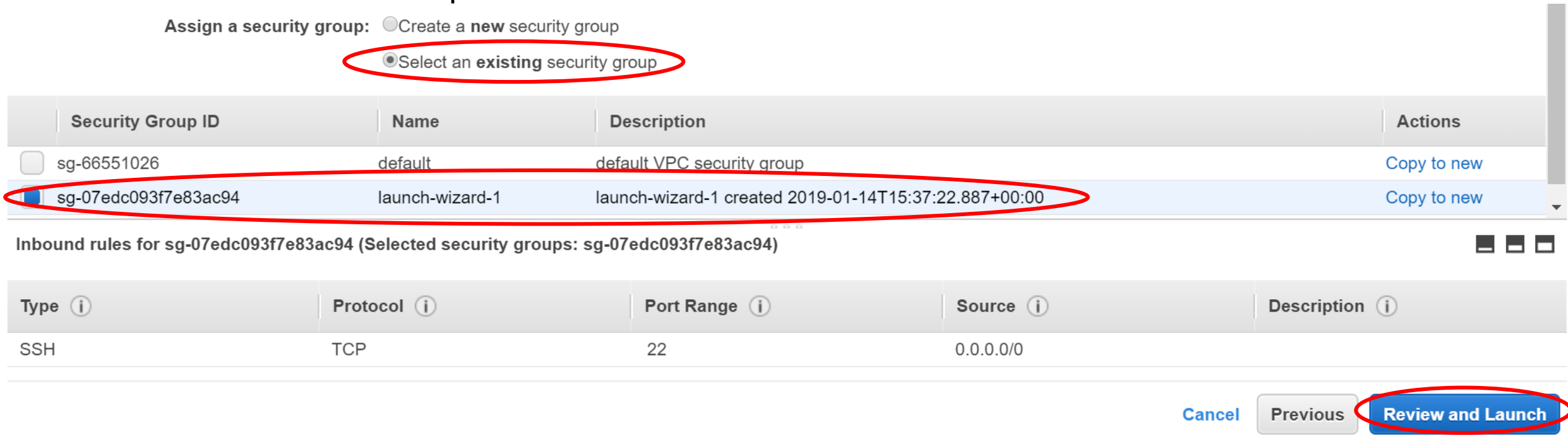
- You need to use up/down arrow key to adjust volume size attached to the EC2 instance.



# Access & Use AWS Cloud with a joint account – Basic Usage – EC2

## Step 6. Configure Security Group

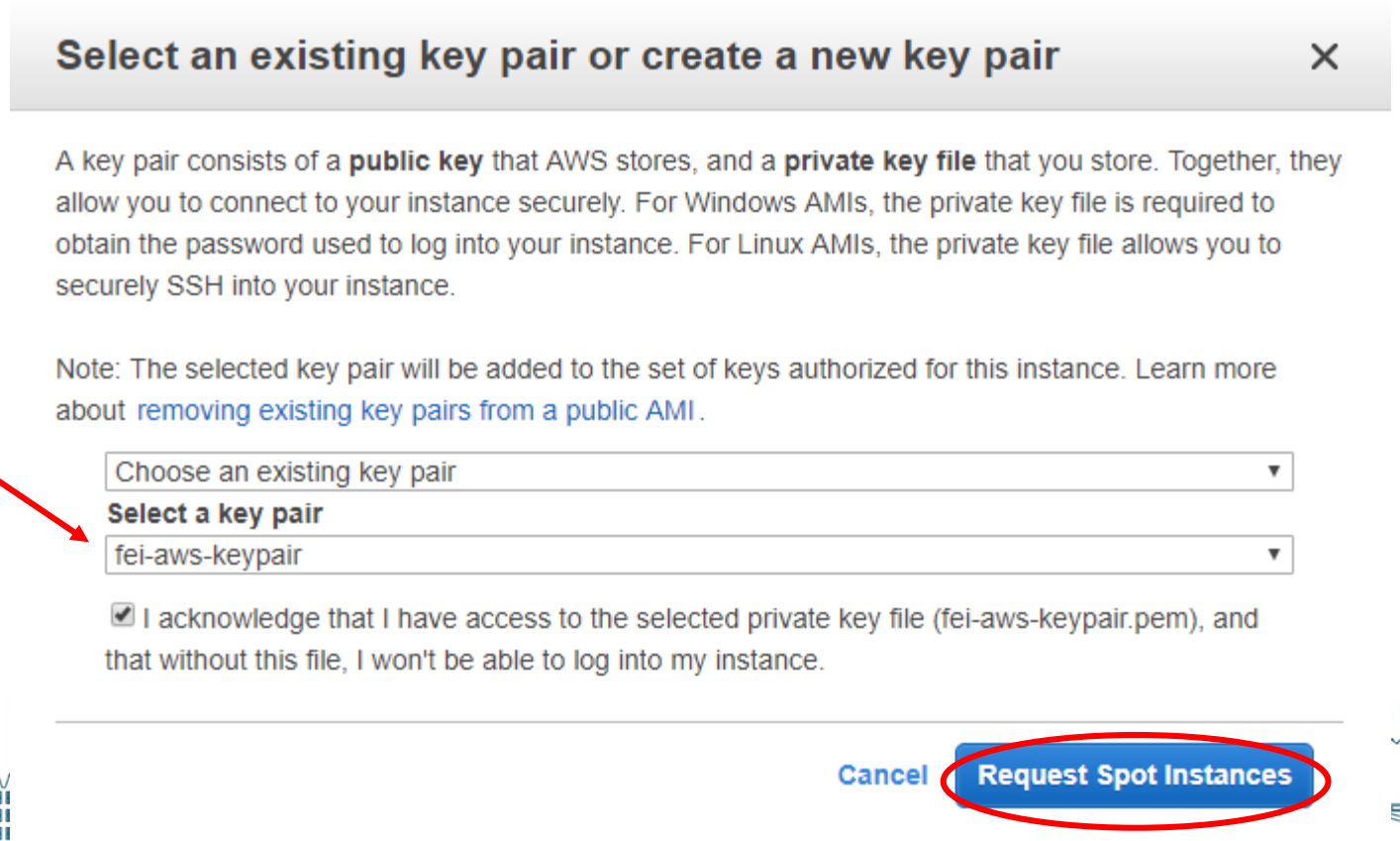
- “Security group” controls what IPs are allowed to access your server. Use an pre-defined one allowing all IPs to access is generally fine because we also need to have the EC2 Key Pair in order to access to a specific server.



# Access & Use AWS Cloud with a joint account – Basic Usage – EC2

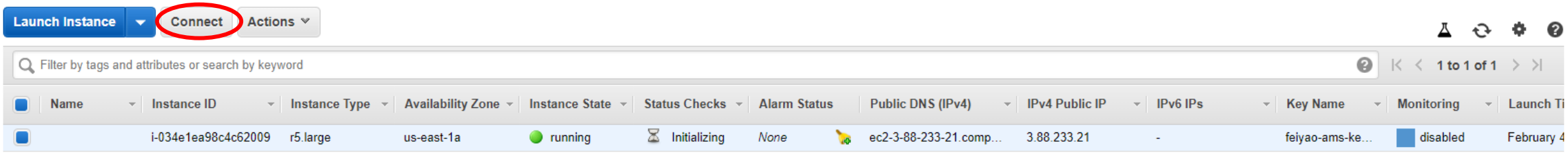
## Step 7. Review & Launch

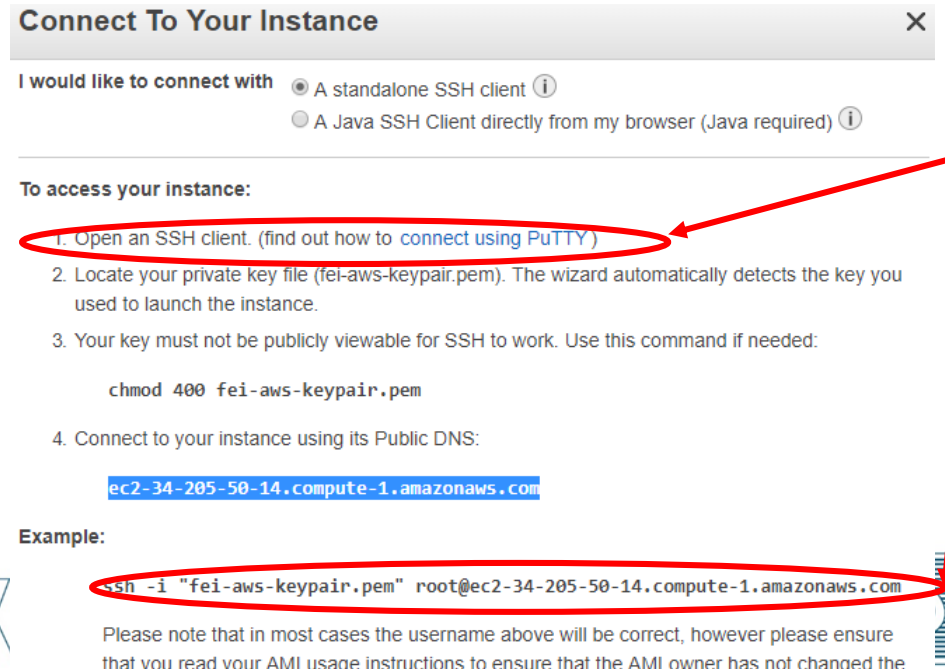
EC2 Key Pair is equivalent to the password you enter to ssh to your local server, but much more secure.



# Access & Use AWS Cloud with a joint account – Basic Usage – EC2

## Step 8. Connect



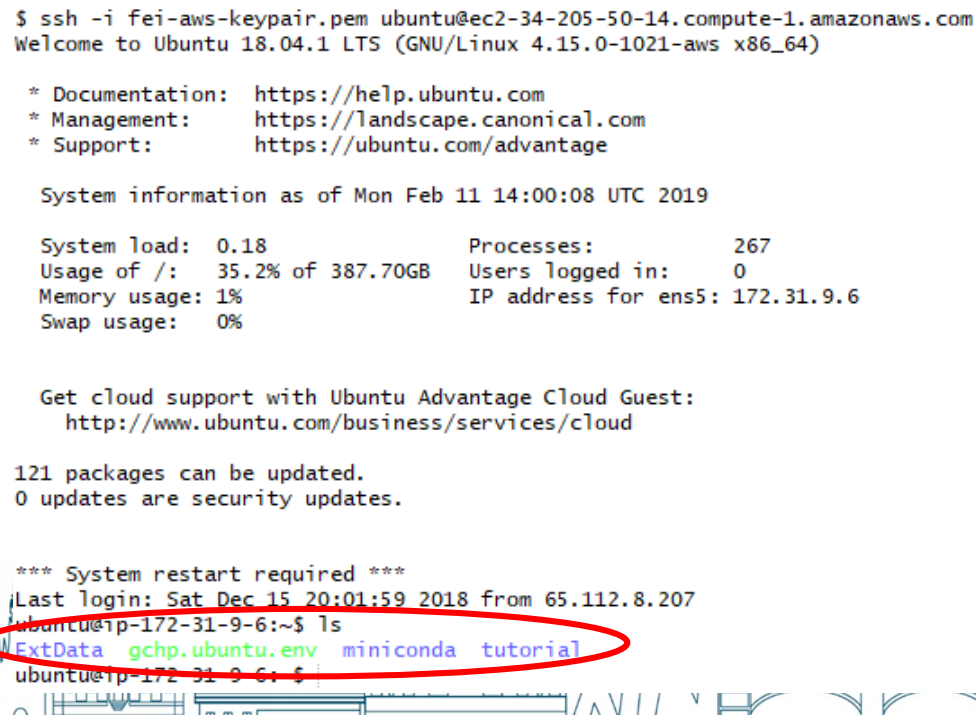




Recommend

Git Bash

root -> ubuntu

Something already here

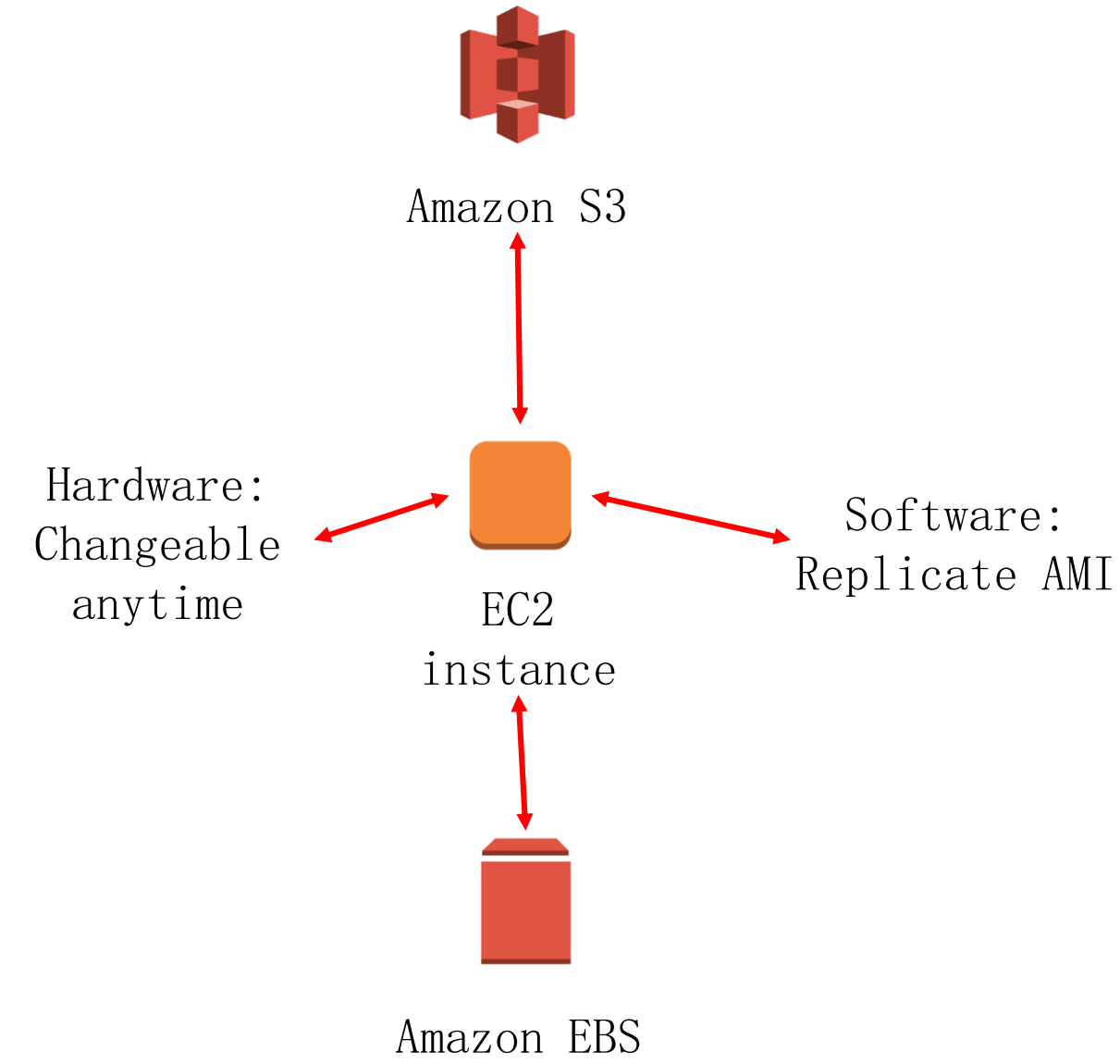


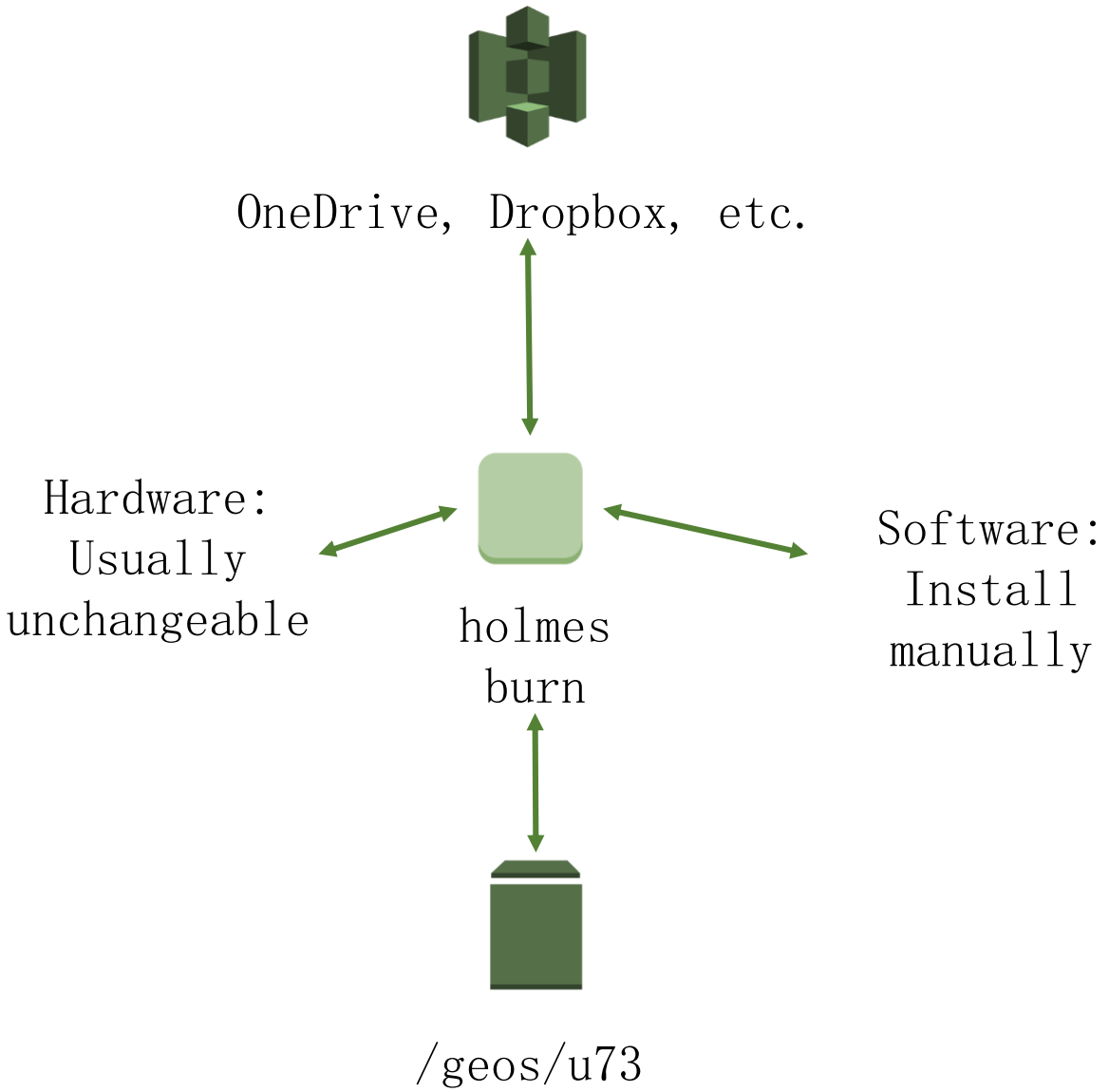
# Access & Use AWS Cloud with a joint account – Basic Usage

## Manipulate S3 with AWSCLI in EC2

| | |
|---|---|
| • Create bucket | $ aws s3 mb s3://bucket-name |
| • See bucket | $ aws s3 ls s3://bucket-name |
| • Upload/download files/directories | $ aws s3 cp [--recursive] source target |
| • Remove files/directories | $ aws s3 rm [--recursive] target |

- Source & target can be any combination of S3, EC2, Local and S3.
- These commands together with aws ec2 … are also applicable in local machines provided you properly install and configure awscli.


Amazon S3
Hardware:
Changeable
anytime
Software:
Replicate AMI
EC2
instance
Amazon EBS


OneDrive, Dropbox, etc.
Hardware:
Usually
unchangeable
Software:
Install
manually
holmes
burn
/geos/u73
/geos/d21
...

# Access & Use AWS Cloud with a joint account – Basic Usage – GEOS-Chem

## Get source code and checkout model versions

- Make a separate folder

```
$ mkdir ~/GC
```

- Enter that folder

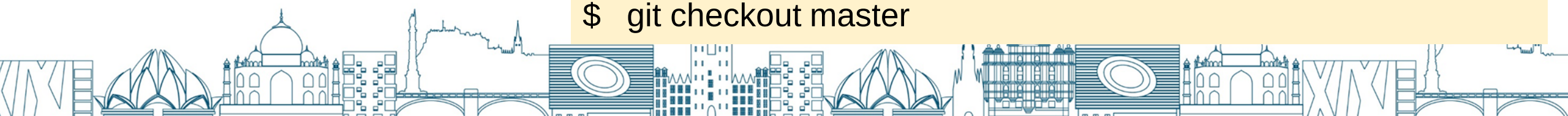
```
$ cd ~/GC
```

- Download GEOS-Chem source code & Unit tester

```
$ git clone https://github.com/geoschem/geos-chem Code.GC
$ git clone https://github.com/geoschem/geos-chem-unittest.git UT
```

- Do version control if needed

```
$ cd Code.GC (UT)
$ git log --tags --simplify-by-decoration --pretty="format:%ci %d“
$ git checkout 12.1.0; git branch
$ git checkout master
```



# Access & Use AWS Cloud with a joint account – Basic Usage – GEOS-Chem

## Configure unit tester and generate run directory

- Change GCGRID_ROOT
- Change DATA_ROOT
- Change UT_TEST_ROOT
- Change COPY_PATH
- Un-comment the desired run directory

$ ./gcCopyRunDirs

```
# %%% Data path and HEMCO settings %%%
#
   GCGRID_ROOT     : /home/ubuntu
   DATA_ROOT       : {GCGRIDROOT}/ExtData
   VERBOSE         : 0
   WARNINGS        : 1
#
# %%% Unit tester path names %%%
#
   UNIT_TEST_ROOT : {HOME}/GC/UT
   RUN_ROOT        : {UTROOT}/runs
   RUN_DIR         : {RUNROOT}/{RUNDIR}
   PERL_DIR        : {UTROOT}/perl
#
# %%% Target directory and copy command %%%
#
   COPY_PATH       : {HOME}/GC
   COPY_CMD        : cp -rfL
#
# !RUNS:
#  Specify the runs directories that you want to copy below.
#  Here we provide a few examples, but you may copy additional entries from
#  UnitTest.input and modify the dates as needed. You can deactivate copying
#  run certain directories by commenting them out with "#".
#
#--------|-----------|------|----------------|------------|--------------|-----|
# MET    | GRID      | NEST | SIMULATION     | START DATE | END DATE     |EXTRA|
#--------|-----------|------|----------------|------------|--------------|-----|
# ======= Standard ============================================================
# geosfp   4x5         -      standard         2016070100   2016080100     -
# merra2   4x5         -      standard         2016070100   2016080100     -
  geosfp   2x25        -      standard         2016070100   2016080100     -
# merra2   2x25        -      standard         2016070100   2016080100     -
# ======= GEOS-Chem benchmark ================================================
```

# Access & Use AWS Cloud with a joint account – Basic Usage – GEOS-Chem

## Configure Makefile and Compile the source code

- Change source code path

```
# Source code location (you can modify as necessary)
ifndef CODE_DIR
        CODE_DIR    :=$(HOME)/GC/Code.GC
endif
```

- Compile

```
# Remove all pre-compiled ones
# -j4 denotes 4 jobs simultaneously
# mpbuild denotes using multiple processors
# NC_DIAG denotes exporting netCDF diagnostics
# BPCH_DIAG denotes exporting BPCH diagnostics
# TIMERS?
```

```
$ make realclean
$ make -j4 mpbuild NC_DIAG=y BPCH_DIAG=n TIMERS=1
```

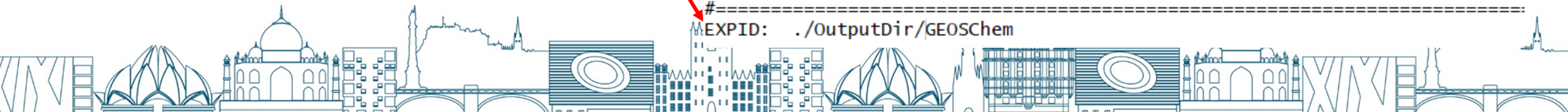
# Access & Use AWS Cloud with a joint account – Basic Usage – GEOS-Chem

## Configure input.geos and HISTORY.rc

- Change simulating period
- Create a directory named OutputDir under the run directory

```
%%% SIMULATION MENU %%% :
Start YYYYMMDD, hhmmss   : 20160701 000000
End   YYYYMMDD, hhmmss   : 20160701 010000
Run directory            : ./
Root data directory      : /home/ubuntu/ExtData
Global offsets I0, J0    : 0 0
```

```
#============================================================================
# EXPID allows you to specify the beginning of the file path corresponding
# to each diagnostic collection.  For example:
#
#   EXPID: ./GEOSChem
#      Will create netCDF files whose names begin "GEOSChem",
#      in this run directory.
#
#   EXPID: ./OutputDir/GEOSChem
#      Will create netCDF files whose names begin with "GEOSChem"
#      in the OutputDir sub-folder of this run directory.
#
#============================================================================
EXPID:   ./OutputDir/GEOSChem
```



# Access & Use AWS Cloud with a joint account – Basic Usage – GEOS-Chem

## Configure input.geos and HISTORY.rc

- Remove # to collect
- Specify collection
  - Frequency
  - Duration
  - Mode

```
# To enable a collection, remove the "#" character in front of its name. The
# Restart collection should always be turned on.
#
# NOTE: These are the "default" collections for GEOS-Chem, but you can create
# your own customized diagnostic collections as well.
#============================================================================
COLLECTIONS: 'Restart',
             #'SpeciesConc',
             #'Budget',
             'AerosolMass',
             #'Aerosols',
             #'CloudConvFlux',
             #'ConcAfterChem'
# Aerosol and PM2.5 mass
#
# Available for full-chemistry and aerosol-only simulations
#============================================================================
  AerosolMass.template:       '%y4%m2%d2_%h2%n2z.nc4',
  AerosolMass.format:         'CFIO',
  AerosolMass.frequency:      00000000 002000
  AerosolMass.duration:       00000000 002000
  AerosolMass.mode:           'time-averaged'
  AerosolMass.fields:         'AerMassBC                ', 'GIGCchem',
                              'AerMassNH4               ', 'GIGCchem',
                              'AerMassNIT               ', 'GIGCchem',
                              'AerMassPOA               ', 'GIGCchem',
                              'AerMassSAL               ', 'GIGCchem',
                              'AerMassSO4               ', 'GIGCchem',
                              'AerMassSOAGX             ', 'GIGCchem',
                              'AerMassSOAMG             ', 'GIGCchem',
                              'PM25                     ', 'GIGCchem',
                              'TotalOA                  ', 'GIGCchem',
                              'TotalOC                  ', 'GIGCchem',
::
```

# Access & Use AWS Cloud with a joint account – Basic Usage – GEOS-Chem

## Pull shared data from S3 to EC2

- s3://gcgrid is where GEOS-Chem shared data reside.

```
$   for month in {07 08}
>   do
>   aws s3 cp --request-payer=requester --recursive \
>     s3://gcgrid/GEOS_2x2.5/GEOS_FP/2016/$month \
>     ~/ExtData/GEOS_2x2.5/GEOS_FP/2016/$month
>   done
```

# Access & Use AWS Cloud with a joint account – Basic Usage – GEOS-Chem

## Obtain additional files

```
# GEOSFP 2x2.5 CN metfield
$ aws s3 cp --request-payer=requester --recursive s3://gcgrid/GEOS_2x2.5/GEOS_FP/2011/01/ ~/ExtData/GEOS_2x2.5/GEOS_FP/2011/01/
# 2x2.5 restart file
$ aws s3 cp --request-payer=requester s3://gcgrid/GEOSCHEM_RESTARTS/v2018-11/initial_GEOSChem_rst.2x25_standard.nc ~/ExtData/GEOSCHEM_RESTARTS/v2018-11/
# fix the softlink in run directory
$ ln -s ~/ExtData/GEOSCHEM_RESTARTS/v2018-11/initial_GEOSChem_rst.2x25_standard.nc ~/GC/geosfp_2x25_standard/GEOSChem.Restart.20160701_0000z.nc4
```

# Access & Use AWS Cloud with a joint account – Basic Usage – GEOS-Chem

## Run GEOS-Chem

```
#  run GEOS-Chem (To monitor running time, type time ./geos.mp)
$  ./geos.mp
```

- Subject to:
  Instance type
  time steps
  "Warm-up"

```
**************  E N D   O F   G E O S -- C H E M  **************

real    8m51.286s
user    49m27.746s
sys     0m55.446s

**************  E N D   O F   G E O S -- C H E M  **************

real    4m24.605s
user    49m20.661s
sys     0m54.415s
```

- Denotes time-averaged value between 20160701-0000z to 20160701-0020z

```
ubuntu@ip-172-31-9-6:~/GC/geosfp_2x25_standard$ ll ./OutputDir/
total 718020
drwxrwxr-x 2 ubuntu ubuntu      4096 Feb 11 16:29 ./
drwxrwxr-x 3 ubuntu ubuntu      4096 Feb 11 16:29 ../
-rw-rw-r-- 1 ubuntu ubuntu  25402406 Feb 11 16:28 GEOSChem.AerosolMass.20160701_0000z.nc4
-rw-rw-r-- 1 ubuntu ubuntu  25346481 Feb 11 16:29 GEOSChem.AerosolMass.20160701_0020z.nc4
-rw-rw-r-- 1 ubuntu ubuntu  25312359 Feb 11 16:29 GEOSChem.AerosolMass.20160701_0040z.nc4
-rw-rw-r-- 1 ubuntu ubuntu 659173261 Feb 11 16:29 GEOSChem.Restart.20160701_0100z.nc4
```

# Access & Use AWS Cloud with a joint account – Basic Usage – GEOS-Chem

## Analyze model outputs using Jupyter

```
#  Re-connect using port-forwarding
$  ssh -i "your-key-name.pem" ubuntu@xxx.amazonaws.com -L 8999:localhost:8999
#  Activate geo python environment
$  source activate geo
#  Start a jupyter
$  jupyter notebook --NotebookApp.token='' --no-browser --port=8999 --notebook-dir ~/
#  Open the url and write analyzing codes
$  http://localhost:8999/
#  Deactivate geo environment
$  source deactivate
```

# Access & Use AWS Cloud with a joint account – Basic Usage – GEOS-Chem

## Save files to S3, terminate server, and start over whenever needed

```
#  Save files (most time run directories) to S3
$  aws s3 cp --recursive ~/GC s3://fei-geoschem-run-directory/GC
#  Terminate server (to do this in the graphical console is convenient enough)
#  Start an EC2 Instance again (steps have been described before)
#  Retrieve run directories from S3 back to EC2
$  aws s3 cp --recursive s3://fei-geoschem-run-directory/GC ~/GC
#  Restore execution permission for geos.mp
$  chmod u+x ~/GC/geosfp_2x25_standard/geos.mp
#  Obtain shared data and GEOSFP 2x2.5 CN metfield (refer to p.28, 29)
```

# Access & Use AWS Cloud with a joint account – Basic Usage

Satellite remotely sensed data processing

- Launch an EC2 instance
- Upload satellite data to S3 (bear in mind that some satellite data are already in S3)
- Pull satellite data from S3 to EC2
- Write Python or similar codes to process satellite data
- Export processed data to S3 and shut down EC2

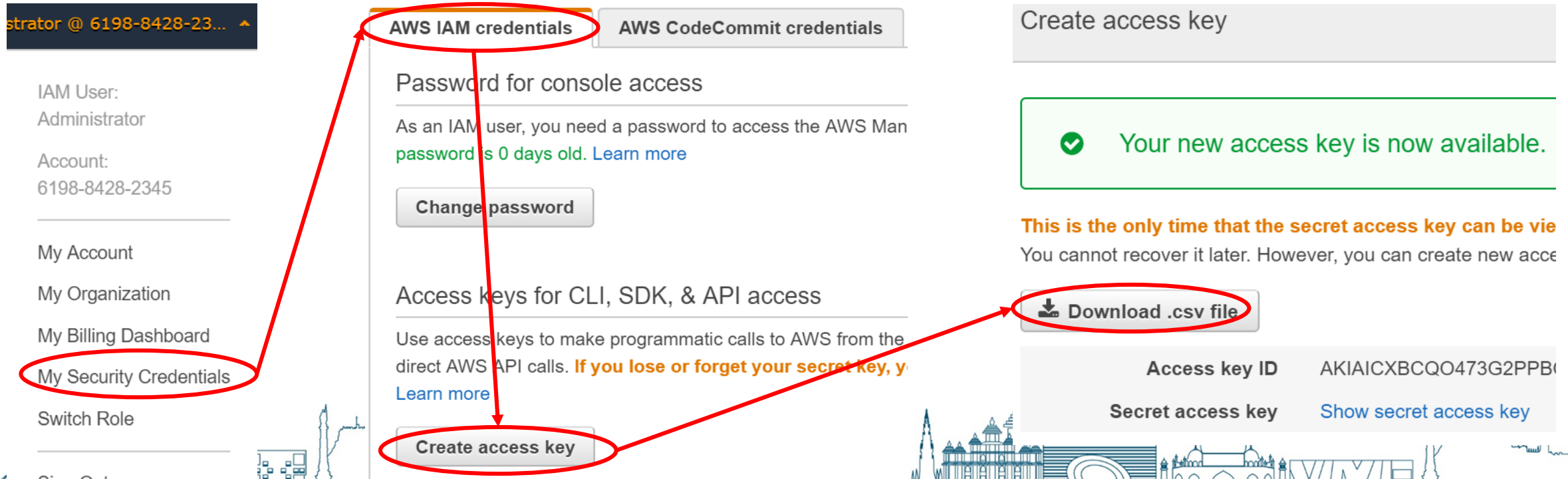
# Contents

- What is Cloud Computing?
- What is AWS Cloud and Why?
- Access & Use AWS Cloud with a Joint Account
  - Basic usage
  - Advanced topics
- Summary & Prospects

# Access & Use AWS Cloud with a joint account – Advanced Topics

## Simplify login process via scripts

- Create Access Key under My security credentials dashboard



# Access & Use AWS Cloud with a joint account – Advanced Topics

## Simplify login process via scripts

- Install AWSCLI on your own computer (i.e. pip install awscli)
- Run aws configure

```
[s1855106@burn ~]$ aws configure
AWS Access Key ID [None]: AKIAICXBCQO
AWS Secret Access Key [None]: 9pRSkCA
Default region name [None]: us-east-1
Default output format [None]: json
```

- It’s in the accessKeys file you downloaded
- us-east-1 is an alias to the “US East (N. Virginia)” region
- Specify configure output format as json

- Configure files are stored in ~/.aws directory

# Access & Use AWS Cloud with a joint account – Advanced Topics

## Simplify login process via scripts

- Write a script and store it in a directory contained in $PATH
- Run the script from terminal
- Export the AWSCLI path under Windows
- If require spot instance

```
#!/bin/bash

# == often change ==
TYPE=c5.4xlarge # EC2 instance type

# == set it once and seldom change ==
AMI=ami-06f4d4afd350f6e4c # AMI to launch from
COUNT=1 # how many instances to launch
IAM=full_S3_access_from_EC2 #EC2 IAM role name
EBS_SIZE=400 # root EBS volume size (GiB)
SG=sg-07edc093f7e83ac94 # security group ID
KEY=fei-aws-keypair # EC2 key pair name

# == almost never change; just leave it as is ==
aws ec2 run-instances --image-id $AMI \
    --instance-type $TYPE \
    --count $COUNT \
    --iam-instance-profile Name=$IAM \
    --block-device-mapping DeviceName="/dev/sda1",Ebs={VolumeSize=$EBS_SIZE} \
    --security-group-ids $SG \
    --key-name $KEY \
    --instance-market-options '{"MarketType":"spot"}'
```

# Access & Use AWS Cloud with a joint account – Advanced Topics

## Keep program running after logging out using tmux

- Create a new session
- Split a panel
- Switch between panels
- Detach from a session
- List all sessions
- Attach to a session

$ tmux new –s session_name

- Ctrl-b %/”
- Ctrl-b ←/→/↑/↓
- Ctrl-b d

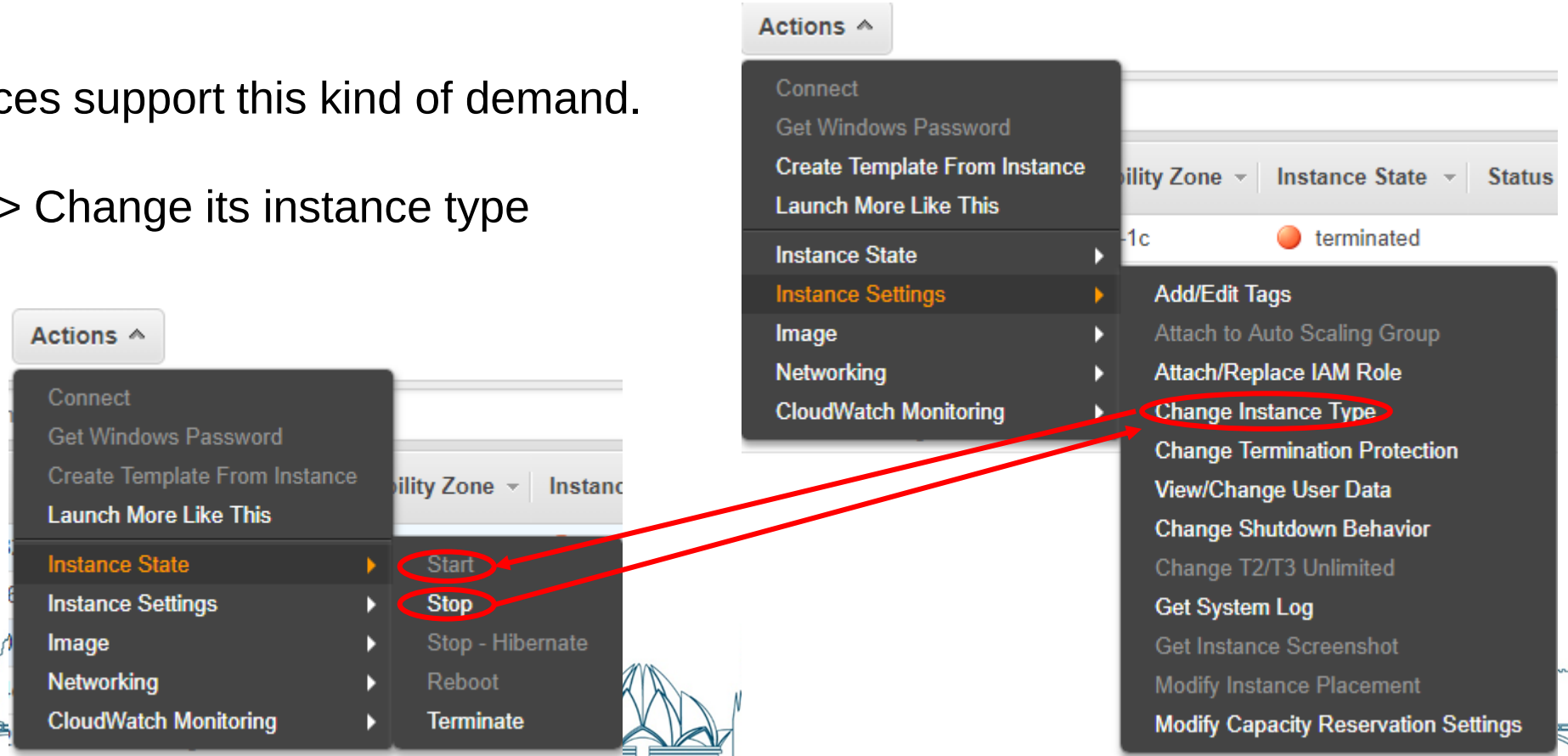
$ tmux list

$ tmux attach –t session_name

# Access & Use AWS Cloud with a joint account – Advanced Topics

## What if I want to change Instance Type of a launched EC2 instance?

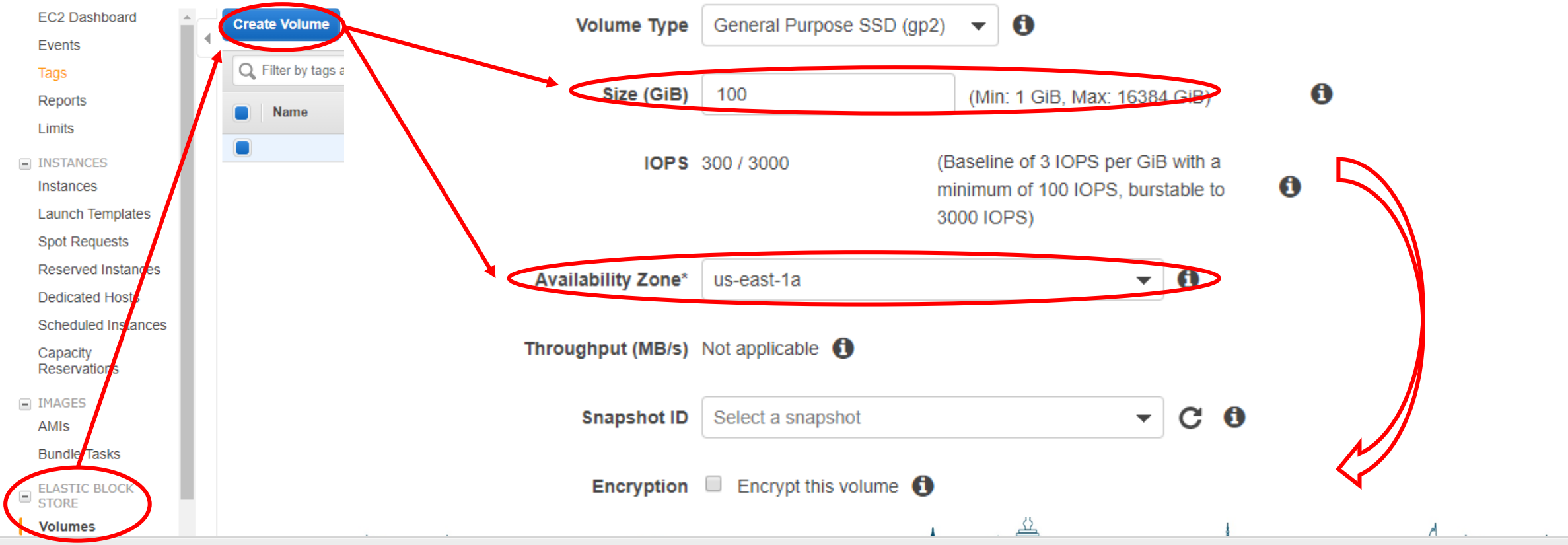
- On-demand instances support this kind of demand.
- Stop the instance -> Change its instance type
  -> start it again



# Access & Use AWS Cloud with a joint account – Advanced Topics

## What if I need more volume size to a launched EC2 instance?

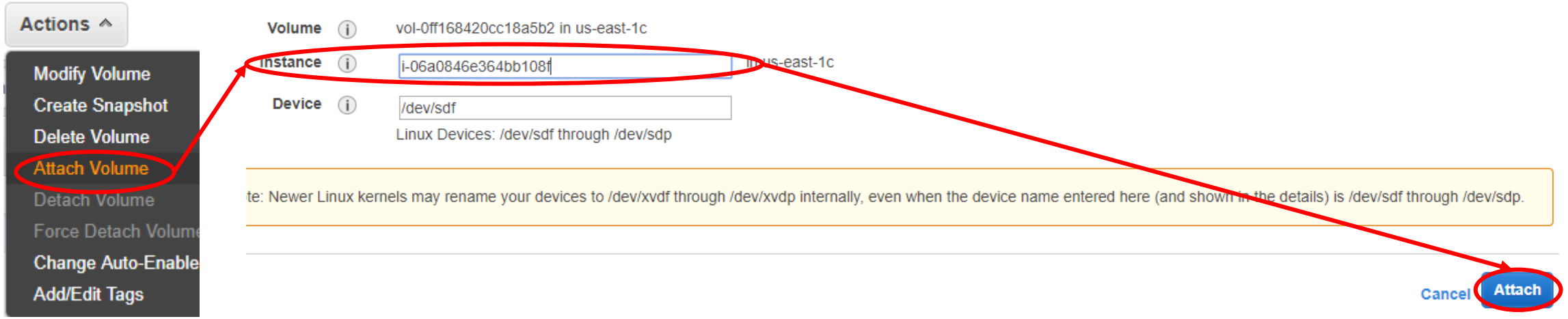
- Create a volume



# Access & Use AWS Cloud with a joint account – Advanced Topics

## What if I need more volume size to a launched EC2 instance?

- Attach that volume to an EC2 instance



# Access & Use AWS Cloud with a joint account – Advanced Topics

## What if I need more volume size to a launched EC2 instance?

- Make that volume usable

```
#  Show that volume to be mounted
$  lsblk
#  Create a file system for that volume
$  sudo mkfs -t ext4 /dev/nvme1n1
#  Mount that volume to a directory
$  mkdir new_disk
$  sudo mount /dev/nvme1n1 new_disk
```

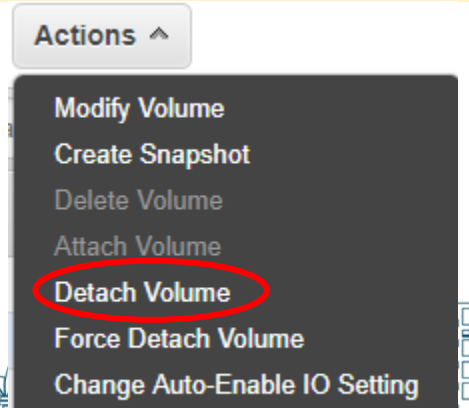
```
#  Show the mounted volume
$  df –h
#  Change permission
$  sudo chown ubuntu new_disk
#  Touch a new file
$  touch ~/new_disk/new_text
```

# Access & Use AWS Cloud with a joint account – Advanced Topics

## What if I need more volume size to a launched EC2 instance?

- Share files between EC2 instances via EBS

```
#  Umount the mounted volume
$  sudo umount /dev/nvme1n1
#  Rmove new_disk (Optional)
$  rmdir new_disk
#  Detach that volume
```



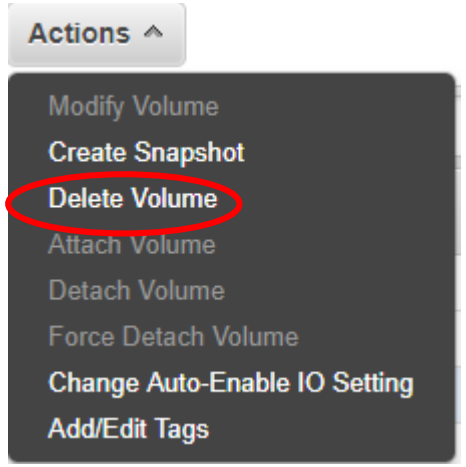

```
#  Attach to another instance
#  Mount that volume to a directory
$  mkdir new_disk
$  sudo mount /dev/nvme1n1 new_disk
#  Show contents in new_disk
$  ls new_disk/
```

# Access & Use AWS Cloud with a joint account – Advanced Topics

## What if I need more volume size to a launched EC2 instance?

- Sweep the battlefield

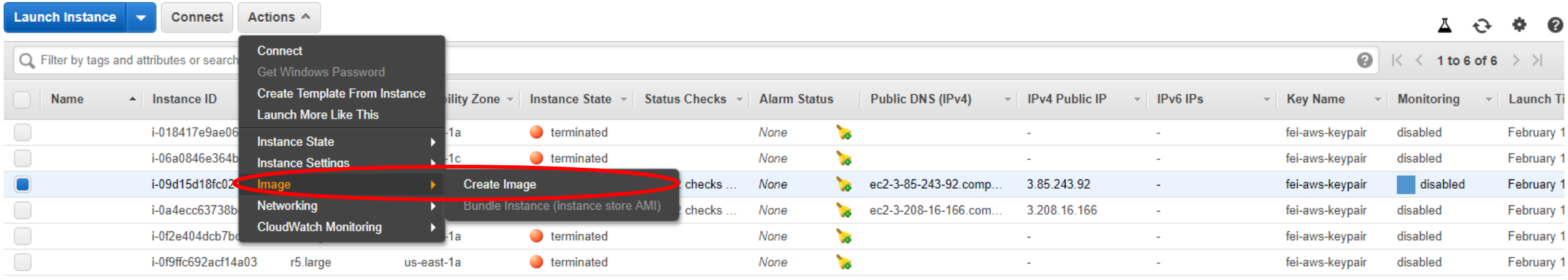
```
#  Save files to S3 if applicable
#  Umount the mounted volume
$  sudo umount /dev/nvme1n1
#  Rmove new_disk (Optional)
$  rmdir new_disk
#  Detach that volume
#  Delete that volume
```




# Access & Use AWS Cloud with a joint account – Advanced Topics

## What if I want to customize my own EC2 instance and save it in order to facilitate later use?

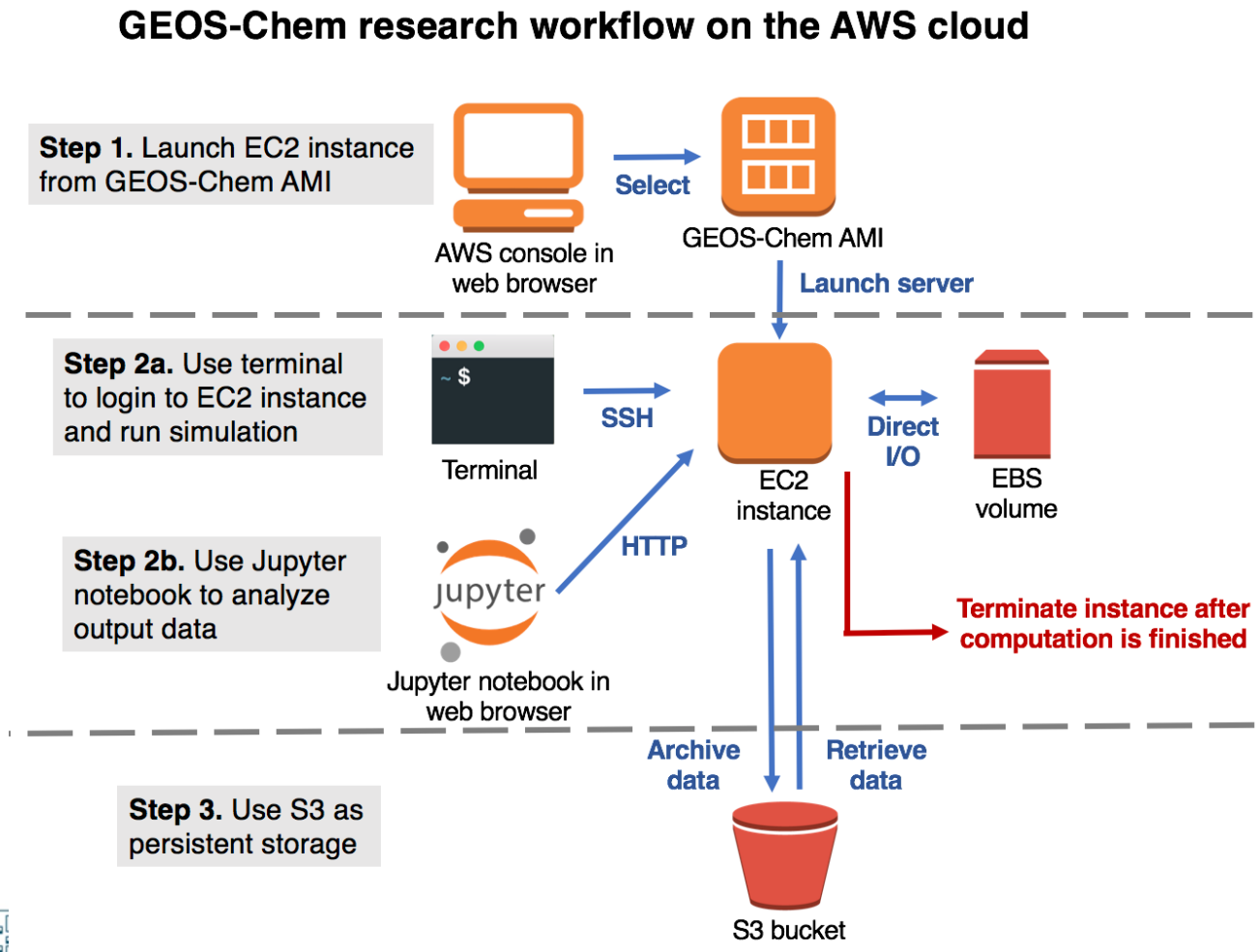
- Launch an instance from a clean AMI
- Install compliers, libraries, and etc. by apt or yum dependent on which system you launch
- Save that instance as your AMI



# Contents

- What is Cloud Computing?
- What is AWS Cloud and Why?
- Access & Use AWS Cloud with a Joint Account
  - Basic usage
  - Advanced topics
- Summary & Prospects

# Summary & Prospects




(Zhuang et al., 2019, submitted to BAMS)

# Summary & Prospects

## High Performance GEOS-Chem

- Already on AWS deserving exploration.

## More flexible permission controls within group

- Will surely come in the near future with your effort.

## Prepayment

- e.g. Reserved Instance

# Materials mainly from:

## GEOS-Chem on AWS cloud tutorial

- https://cloud-gc.readthedocs.io/en/latest/index.html

## AWS Document

- https://docs.aws.amazon.com/index.html#lang/en_us

THE UNIVERSITY of EDINBURGH

# Thanks! Your Response is appreciated!