

# AWS: IAM, S3, Spot Instances and more

Xin Yin

10/01/2015

Cloud Computing Working Group

# IAM - Identity and Access Management

# Data Storage - Your options

- ▶ EBS - Elastic Block Store
  - ▶ **Good:** can be mounted as a block device to EC2 instances.
  - ▶ **Good:** provides file system.
  - ▶ **Bad:** can only be mounted to one instance.
  - ▶ **Bad:** charged at provisioned volume.
- ▶ S3 - Simple Storage Service
  - ▶ **Good:** *Simple* and versatile
  - ▶ **Good:** more reliable (than EBS)
  - ▶ **Good:** cheaper (than EBS)  
\$0.03/GiB *vs.* \$0.10/GiB (*provisioned*)
  - ▶ **Good:** supports versioning
  - ▶ **Good:** files can be made publicly available
  - ▶ **Bad:** not a file system

# S3 - Simple Storage Service

- ▶ live demo

# Calculate the costs of your spot instances

- ▶ EC2 Instances: price varies by instance type
- ▶ S3 Storage:
  - ▶ Volume: \$ 0.03 / GiB × month
  - ▶ Requests: ~ \$ 0.005 / 1000 Reqs
- ▶ Data transfer:
  - ▶ Inbound: *free*
  - ▶ Outbound (EC2/S3 to the Internet): \$ 0.09 / GiB
  - ▶ Between AWS Regions: \$ 0.02 / GiB

# How to reduce costs?

1. Determine the most suitable instance types
2. Minimize/eliminate cross-regional data transmission
3. Maximize the utilization of computational resources
4. Protect yourself from data loss due to instance termination

# What is the best instance type for me?

- ▶ Determining the limiting factor of your program:

- ▶ CPU: *simulations*
- ▶ Memory: *bioinformatics*
- ▶ Storage: *big datasets*

- ▶ time

```
alias timem="/usr/bin/time -f 'real %e\nuser %U\nsys %S\nmem %M Kb'"
```

```
timem <some_program>
```

```
real 18390.10
```

```
user 128669.82
```

```
sys 2318.05
```

```
mem 45031068 Kbytes
```

- ▶  $(128669.82 + 2318.05)/18390.10 = 7.13 \approx (8 \text{ cores})$
- ▶  $45031068/1024/1024 = 42.94(\text{GiB})$

# What is the best instance type for me?

- ▶ Determining the limiting factor of your program:

- ▶ CPU: *simulations*
- ▶ Memory: *bioinformatics*
- ▶ Storage: *big datasets*

- ▶ time

```
alias timem="/usr/bin/time -f 'real %e\nuser %U\nsys %S\nmem %M Kb'"
```

```
timem <some_program>
```

```
real 18390.10
```

```
user 128669.82
```

```
sys 2318.05
```

```
mem 45031068 Kbytes
```

- ▶  $(128669.82 + 2318.05)/18390.10 = 7.13 \approx (8 \text{ cores})$
- ▶  $45031068/1024/1024 = 42.94(\text{GiB})$
- ▶ How many GBs of memory per core?
- ▶ <https://aws.amazon.com/ec2/instance-types/>



# Minimize cross-regional data transmission

- ▶ Launch EC2 instances in the same region of your S3 buckets
  - ▶ No £ 0.02 / GiB cost for data transfer
  - ▶ Low-latency, high-throughput

# Maximize the utilization of computational resources

- ▶ CPUs, are they 100% utilized?
- ▶ Reduce the amount of time on setups:
  - ▶ Use *user data*
  - ▶ Launch instance from your AMI
  - ▶ Use **CodeDeploy**
- ▶ Load balancing:
- ▶ Automation and monitoring:

# Inevitable price fluctuation

- ▶ Study the pricing history (and predict the variation)
- ▶ Don't be intimidated to bid **higher**
  - ▶ You are only charged for the *current price*, not your *maximum bid price*.
  - ▶ Protect you from price surges and spikes.
- ▶ If possible, divide your program into small segments:
  - ▶ Example: simulations, MCMC
  - ▶ After each segment, backup the results to S3
- ▶ Termination Notices:  
`http://169.254.169.254/latest/meta-data/spot/termination-time`