

# CS15-319 / 15-619

## Cloud Computing

Recitation 4

February 4<sup>th</sup> & 6<sup>th</sup>, 2014

# Announcements

- Encounter a general bug:
  - Post on Piazza publicly
- Encounter a grading bug:
  - Post Privately on Piazza
- Do not post your code on Piazza
- Post feedback on OLI

MY RESPONSE...



Data Center Design Considerations: Feedback, comments, and bugs

---

- Do not forget to Tag your instances
  - Key: Project      Value: 2.1

# Last Week

- You have completed
  - Sequential Analysis
  - Elastic MapReduce
- You should have learned
  - Why MapReduce for big data
  - How MapReduce works
  - How to write your own Mapper/Reducer

# Project 1.2 Checkpoint

- We will manually grade Question 1
  - Be sure to make your code readable
  - We will not deduct points for style
    - ONLY this time!

# Piazza Questions

- Elastic MapReduce Billing Question
  - [Normalized Hours \(Elastic MapReduce\)](#)

Date	Elapsed Time	Normalized Instance Hours
1 11:59 EDT	1 hour 46 minutes	40

- 1 hour of m1.small = 1 hour normalized compute time
- 1 hour of m1.medium = 2 hours normalized compute time
- 1 hour of m1.large = 4 hours normalized compute time
- 1 hour of m1.xlarge = 8 hours normalized compute time
- 1 hour of c1.medium = 2 hours normalized compute time

# Piazza Questions

- Elastic MapReduce Billing Question
  - [Elastic MapReduce Pricing](#) (On demand)
  - for US East (N. Virginia)

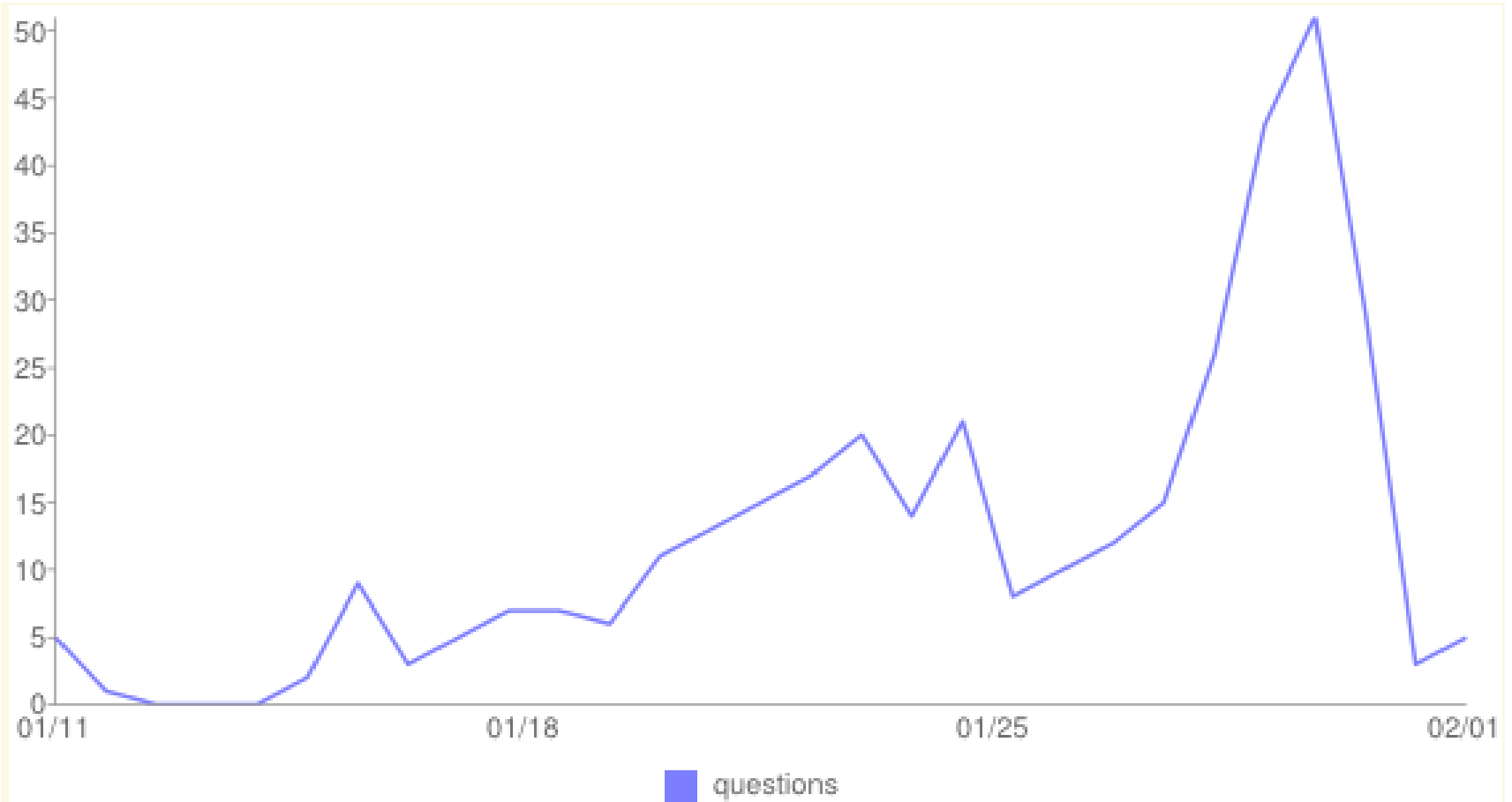
Standard On-Demand Instance	Amazon EC2 Price	Amazon Elastic MapReduce Price
Small	\$0.06 per Hour	\$0.015 per Hour
Medium	\$0.12 per Hour	\$0.03 per Hour
Large	\$0.24 per Hour	\$0.06 per Hour
Extra Large	\$0.48 per Hour	\$0.12 per Hour

Billing price = ( $\$0.12 * 4 \text{ instances} + \$0.03$ ) \* 2 hours

# Piazza Questions

- Late policy
  - We do not have a late policy!
    - All deadlines are hard.
  - Please start early!

# Piazza Posts





# Piazza Traffic

## Top Student Answerers

Name, Email	number of answers
<a href="#">Haibo Wang</a> <a href="mailto:haibow@andrew.cmu.edu">haibow@andrew.cmu.edu</a>	7
<a href="#">Benjamin Gummoe</a> <a href="mailto:bgummoe@andrew.cmu.edu">bgummoe@andrew.cmu.edu</a> , <a href="mailto:b.gummoe@gmail.com">b.gummoe@gmail.com</a>	7
<a href="#">Pavan Kumar Sunder</a> <a href="mailto:psunder@andrew.cmu.edu">psunder@andrew.cmu.edu</a>	7
<a href="#">Patrick Woody</a> <a href="mailto:pwoody@andrew.cmu.edu">pwoody@andrew.cmu.edu</a>	6
<a href="#">Karan Sikka</a> <a href="mailto:ksikka@cmu.edu">ksikka@cmu.edu</a>	6
<a href="#">Sairam Krishnan</a> <a href="mailto:sbkrishn@andrew.cmu.edu">sbkrishn@andrew.cmu.edu</a>	5
<a href="#">Jack Wang</a> <a href="mailto:hojehw@andrew.cmu.edu">hojehw@andrew.cmu.edu</a> , <a href="mailto:dilta.ppc@gmail.com">dilta.ppc@gmail.com</a>	5
<a href="#">Dylan Fitzpatrick</a> <a href="mailto:djfitzpa@andrew.cmu.edu">djfitzpa@andrew.cmu.edu</a>	4
<a href="#">David Curd</a> <a href="mailto:dcurd@andrew.cmu.edu">dcurd@andrew.cmu.edu</a>	4
<a href="#">Taylor Makela</a> <a href="mailto:tmakela@andrew.cmu.edu">tmakela@andrew.cmu.edu</a>	4

# This Week

- Unit 2: Data Centers
  - Module 3: Data Center Trends
  - Module 4: Data Center Components
- Read and complete:
  - Module 5: Design Considerations
  - Unit 2: Checkpoint Quiz
    - 150 minutes, due 6<sup>th</sup> Feb., 2014 (Pittsburgh Time)

## [UNIT 2: Data Centers](#)

[Module 3: Data Center Trends](#)

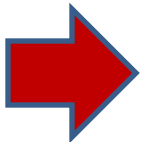
[Module 4: Data Center Components](#)

[Module 5: Design Considerations](#)

Quiz 2: Data Centers

[Checkpoint](#)

[Available 2/03/14 12:01 AM](#)  
[Due 2/06/14 11:59 PM](#)



# This Week

- Project 2.1: Introduction to APIs
  - Single Instance Benchmarking
- Start early!
  - Project 2.1, **due 9<sup>th</sup> February, (Pittsburgh time)**
  - You have three attempts, but no late submission

[Project 2](#)

[Introduction and APIs](#)

Single Instance Benchmarks

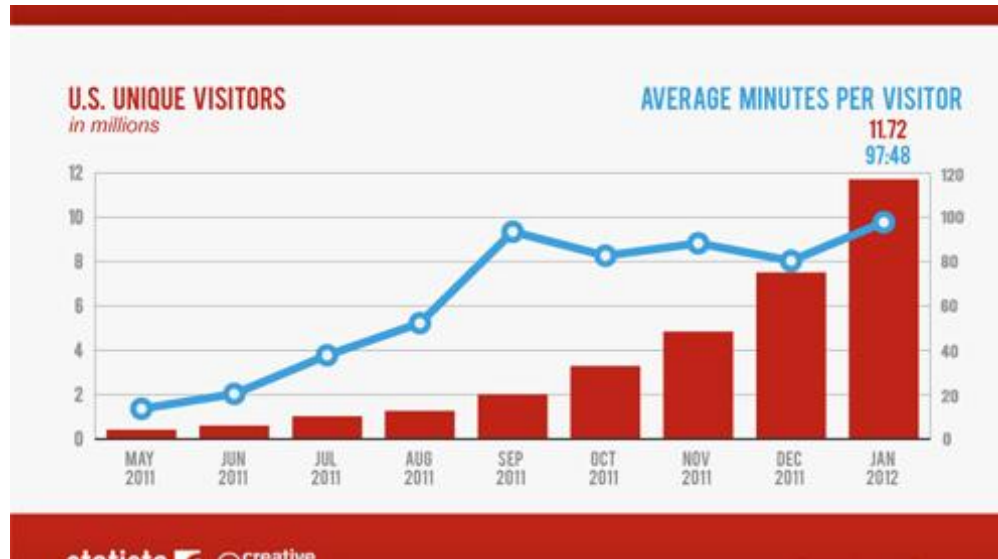
[Checkpoint](#)

[Available Now](#)

[Due 2/09/14 11:59 PM](#)



# Pinterest



- Started with 12 engineers, now 140+ employees
- Total number of users: 70 million
- Growth in web traffic from 9/12-9/13: 66.52%



- Utilizes AWS
  - Auto Scaling
    - Scale up and down
  - Elastic Load Balancer
    - Distribute traffic
  - S3
    - File storage
  - Elastic MapReduce
    - Data analysis

# Project 2 - Scenario

- Online Photo Verification, *PixVerify*

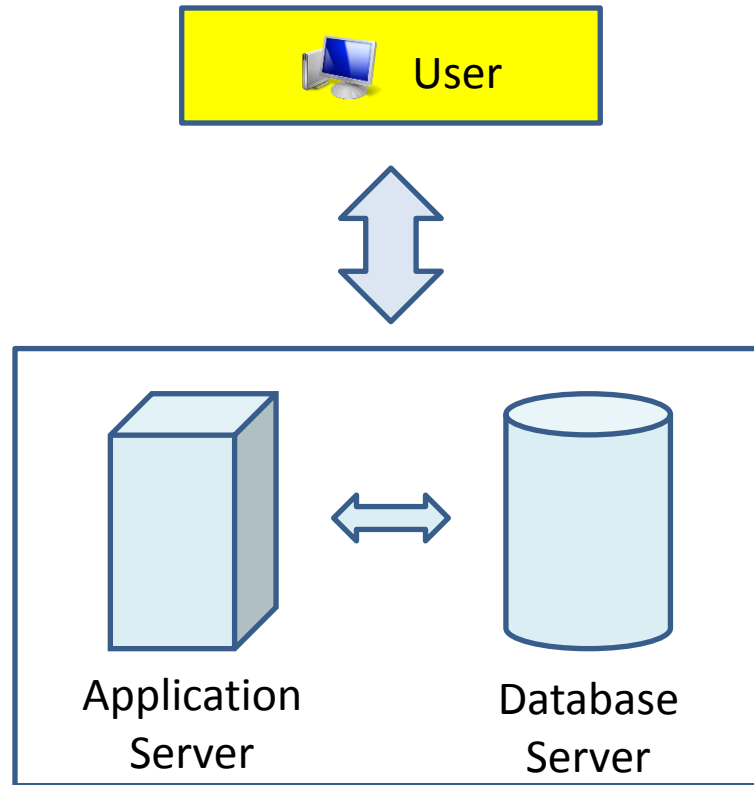


Sales

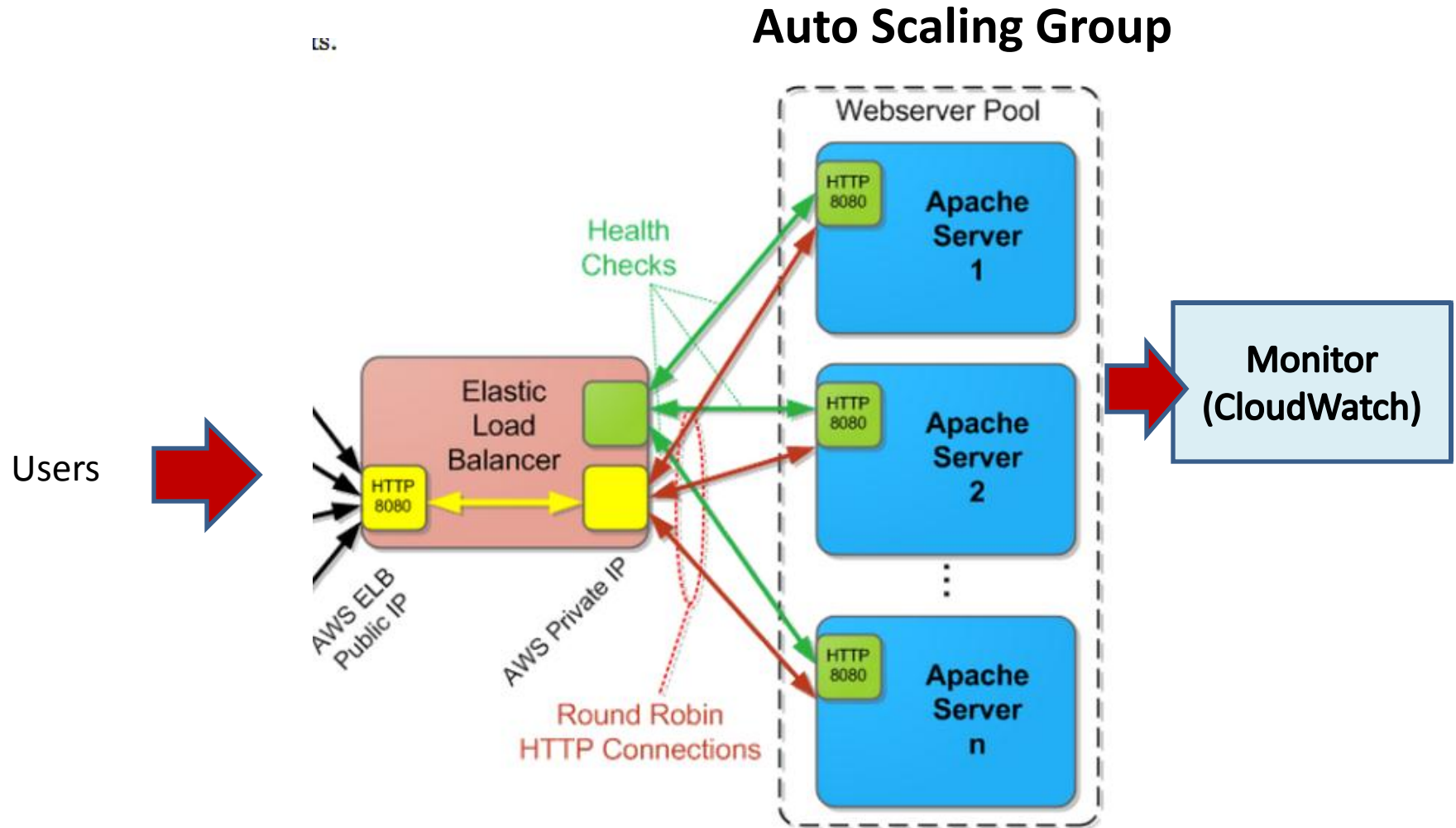


Reliability

# Current Infrastructure

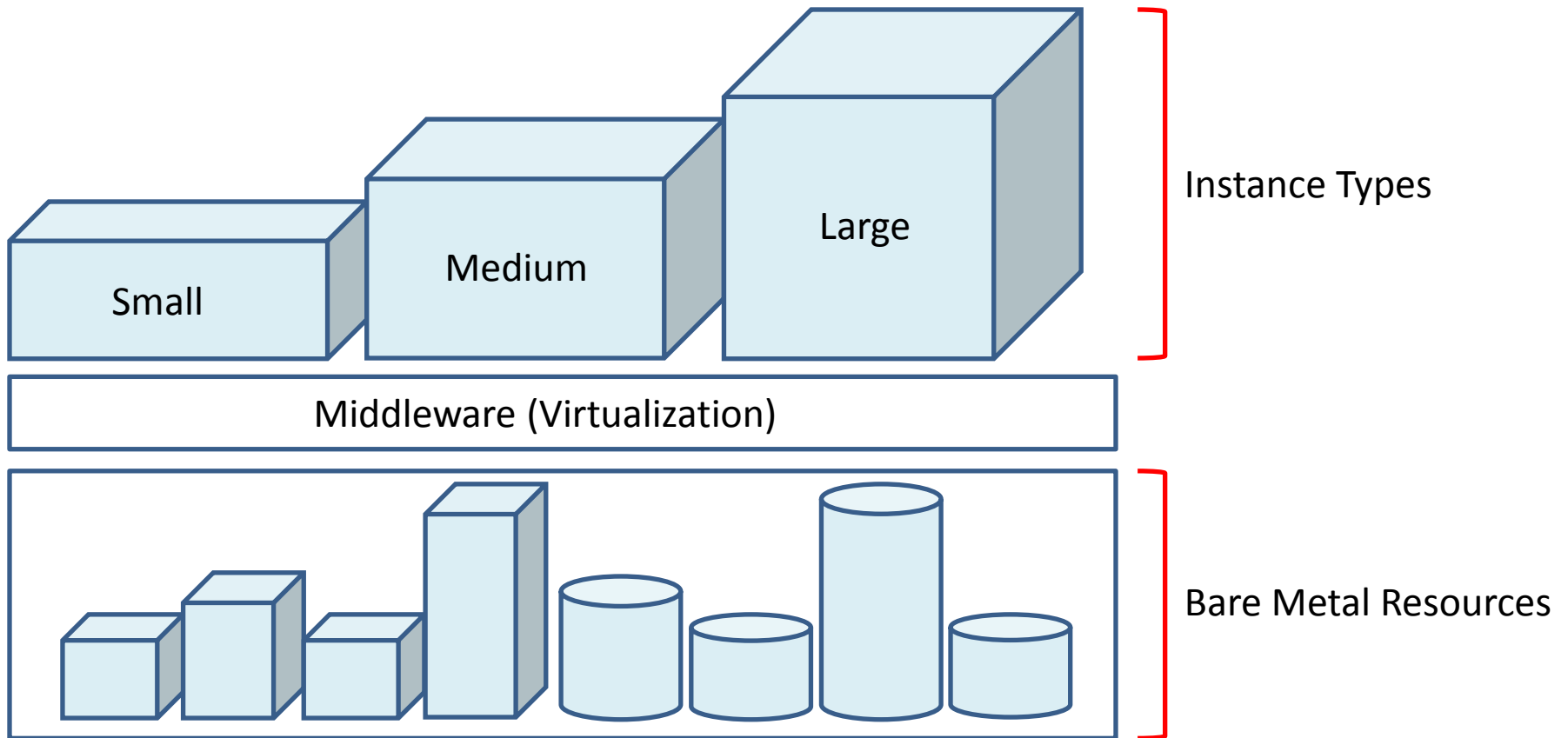


# Overview

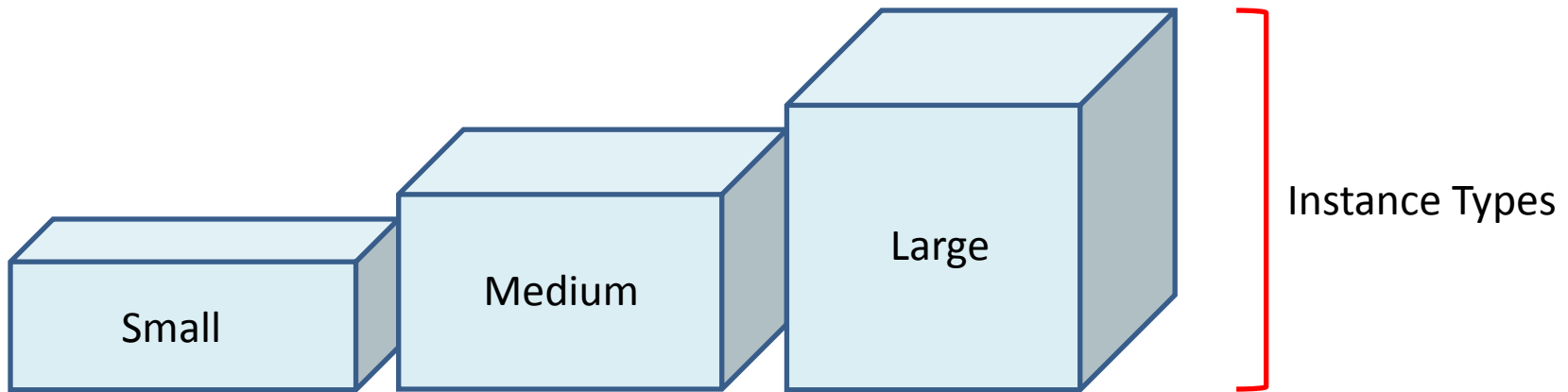




# Cloud Infrastructure



# Maximize Requests per Dollar



**X** REQUESTS

**Y** REQUESTS

**Z** REQUESTS

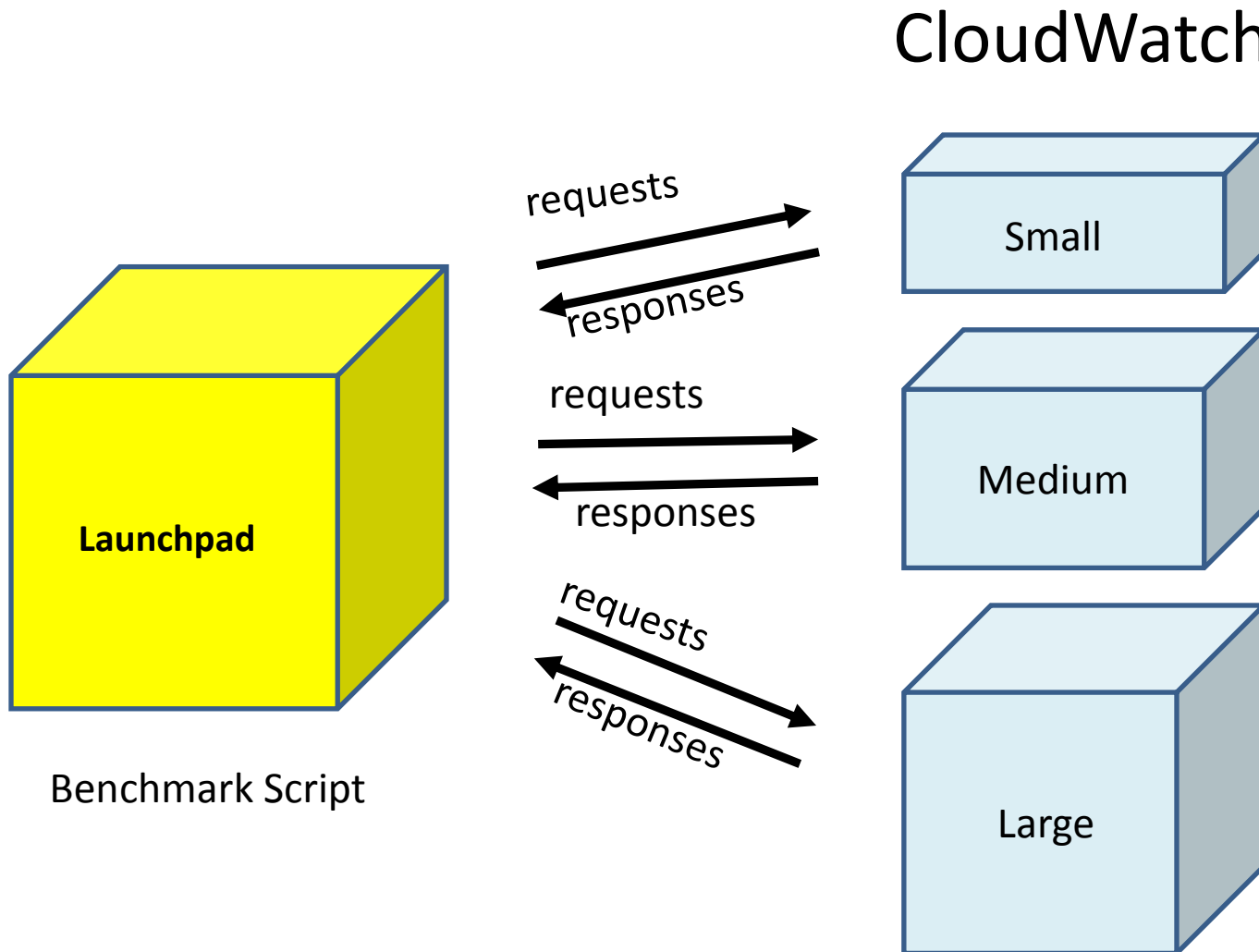
**COST A** <

**COST B** <

**COST C**

**GOAL:** MAXIMIZE (REQUESTS/DOLLAR)

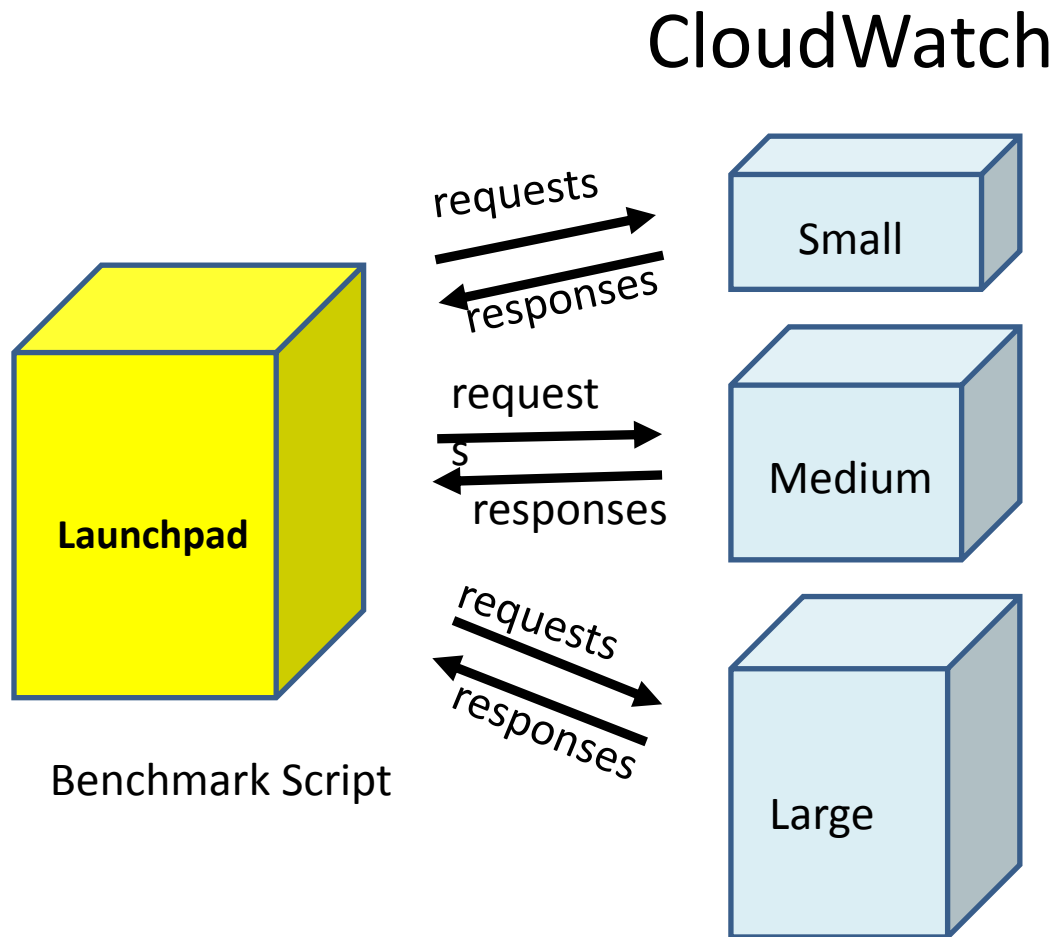
# Load Testing Request & Response Flow



# Amazon APIs

- Provisioning & Monitoring:
  - EC2 API
  - CloudWatch API
- Supported APIs
  - Command Line Interface API Tools
  - AWS SDK for Java
  - AWS SDK for Python

# Load Testing Request & Response Flow



## Steps:

Launch the Launchpad

Upload script/program

- Provision test instance
- Run benchmark script
- CloudWatch to retrieve CPUUtilization
- Store benchmark output into a file
- Terminate test instance
- Repeat for other test instances

Requests/second & CPUUtilization

# Tips

- Make sure the Launchpad and other instances are in the same subnet (availability zone)

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

## Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances	<input type="text" value="1"/>
Purchasing option	<input type="checkbox"/> Request Spot Instances
Network	vpc-4e92742b (172.31.0.0/16) (default) <a href="#">Create new VPC</a>
<b>Subnet</b>	<input checked="" type="checkbox"/> No preference (default subnet in any Availability Zone) <a href="#">Create new subnet</a>
Public IP	subnet-f5b98281(172.31.16.0/20)   Default in us-east-1d subnet-cc082c8a(172.31.0.0/20)   Default in us-east-1a subnet-41e4a069(172.31.32.0/20)   Default in us-east-1c
IAM role	None
Shutdown behavior	Stop
Enable termination protection	<input type="checkbox"/> Protect against accidental termination
Monitoring	<input type="checkbox"/> Enable CloudWatch detailed monitoring Additional charges apply.
Tenancy	Shared tenancy (multi-tenant hardware) Additional charges will apply for dedicated tenancy.

Cancel Previous **Review and Launch** Next: Add Storage

# Tips

- Terminate instances vs. Stop instances
  - Stop will charge for storage (eg. EBS volumes)
  - Record all the data and results you need before you terminate an instance.
- S3 URL submission
  - Verification link
    - <http://ec2-54-225-106-182.compute-1.amazonaws.com/>
  - Do not include a space in the S3 link
- Start early!

# Upcoming Deadlines

- Unit 2: **Timed Quiz (150 minutes)**

## UNIT 2: Data Centers

Module 3: Data Center Trends

Module 4: Data Center Components

Module 5: Design Considerations

Quiz 2: Data Centers

Checkpoint

Available 2/03/14 12:01 AM

Due 2/06/14 11:59 PM



- Project 2: Tag - Key: Project      Value: 2.1

## Project 2

Introduction and APIs

Single Instance Benchmarks

Checkpoint

Available Now

Due 2/09/14 11:59 PM





# Demo Outline

- 1. Launch an instance using command line
  - Setup command line tool environment
  - Launch an instance
  - Demonstrate CloudWatch command line
- 2. Launch an instance using Java API
  - Install AWS SDK in Eclipse
  - Configure AWS account in Eclipse
  - Demonstrate Amazon EC2 API

# Setup ec2 command line tool

- Download and Copy ec2 command line tool into the ec2 instance
- Configuration:
  - Make sure that JAVA\_HOME environment variable is set
  - Set EC2\_HOME & system PATH variable
  - Configure your access key and secret key
  - Set your private key & certificate
- [http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/SettingUp\\_CommandLine.html](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/SettingUp_CommandLine.html)

# Launch an instance

- `ec2-run-instances ami_id [-n instance count] [-k keypair] [--instance-type instance_type] [--availability-zone zone]`
- <http://docs.aws.amazon.com/AWSEC2/latest/CommandLineReference/ApiReference-cmd-RunInstances.html>

# CloudWatch Command Line

- Download CloudWatch
  - <http://aws.amazon.com/developertools/2534>
- Fill your access key and secret key in  
\$AWS\_CLOUDWATCH\_HOME/credential-file-  
path.template
- You will have to figure out how to use the  
command line
  - Hint: mon-get-stats

# Install AWS Toolkit for Eclipse

- Plug-in for the Eclipse Java IDE
- Help -> Install New Software
- Enter '<http://aws.amazon.com/eclipse>' in 'Work with'
- Select 'AWS Toolkit for Eclipse'
- <http://aws.amazon.com/eclipse/>

# Amazon EC2 API

- Load AWS Credentials
  - BasicAWSCredentials
- Create Amazon EC2 Client
  - AmazonEC2Client
- Create and Configure Instance Request
  - RunInstancesRequest
- Launch Instance
  - RunInstanceResult

# Amazon CloudWatch API

- Load AWS Credentials
  - BasicAWSCredentials
- Create Amazon CloudWatch Client
  - AmazonCloudWatchClient
- Create and Configure Metric Request
  - GetMetricStatisticsRequest
- Get Metric Result
  - GetMetricStatisticsResult
- <http://aws.amazon.com/sdkforjava/>

# Important Notes

1. Run your program of project 2.1 on the same directory of benchmark;
2. Figure out how to
  - a. Set security group for your instance;
  - b. Set availability zone for your instance;
  - c. Tag your instance;
  - d. Use CloudWatch;
3. Start Early;
4. Q & A.