CS15-319 / 15-619
Cloud Computing

Recitation 6

Feb 18th and 20th, 2014

Announcements

- Protect your AWS account!
  - Protect your credentials
  - Do not give anyone access to your account
- Budget Control!
  - Do remember to **TERMINATE** instances when you are done
  - You will be penalized if you spend much more than the class average

Announcements

- Do not cheat!
  - Some suspected cases were found
  - We are using tools to compare code/design
  - You learn nothing when you cheat
- Manual Grading
  - Will be done before Tuesday **one week** after the deadline
- Submission
  - You are allowed to upload a single **ZIP** file to S3

Announcements

- Tag your instances so we can track your expenses
- Ask proper questions on Piazza
  - Search Piazza or the internet (Google) before asking
  - Duplicate questions will be deleted
  - Posting solutions on Piazza will be deleted
    - Might lead to penalties
  - The TAs will not debug for you, even if a private post
  - AWS SDK doc will answer most of your programming questions

Last Week Review

- Horizontal Scaling vs Vertical Scaling
  - Understand the concept and differences
  - When is vertical scaling useful, what are its limitations?
  - How does horizontal scaling work, what is necessary for it to work well?
- Elastic Load Balancing (2 modules)
  - Elastic Load Balancer
  - Static Load Benchmarking

Piazza Questions

- Script/Program must be under the same folder as the benchmark folder
- When trying to run the apache benchmark, please make sure that you are not including the HTTP in front of the DNS

ELB Needs Warming Up

- ELB has a starting point for its initial capacity, and it will scale up or down based on traffic
- It struggles with high traffic spikes in shorter periods
- It is recommended that the load is increased at a rate of no more than 50 percent every five minutes

Unit 3: Virtualizing Resources for the Cloud

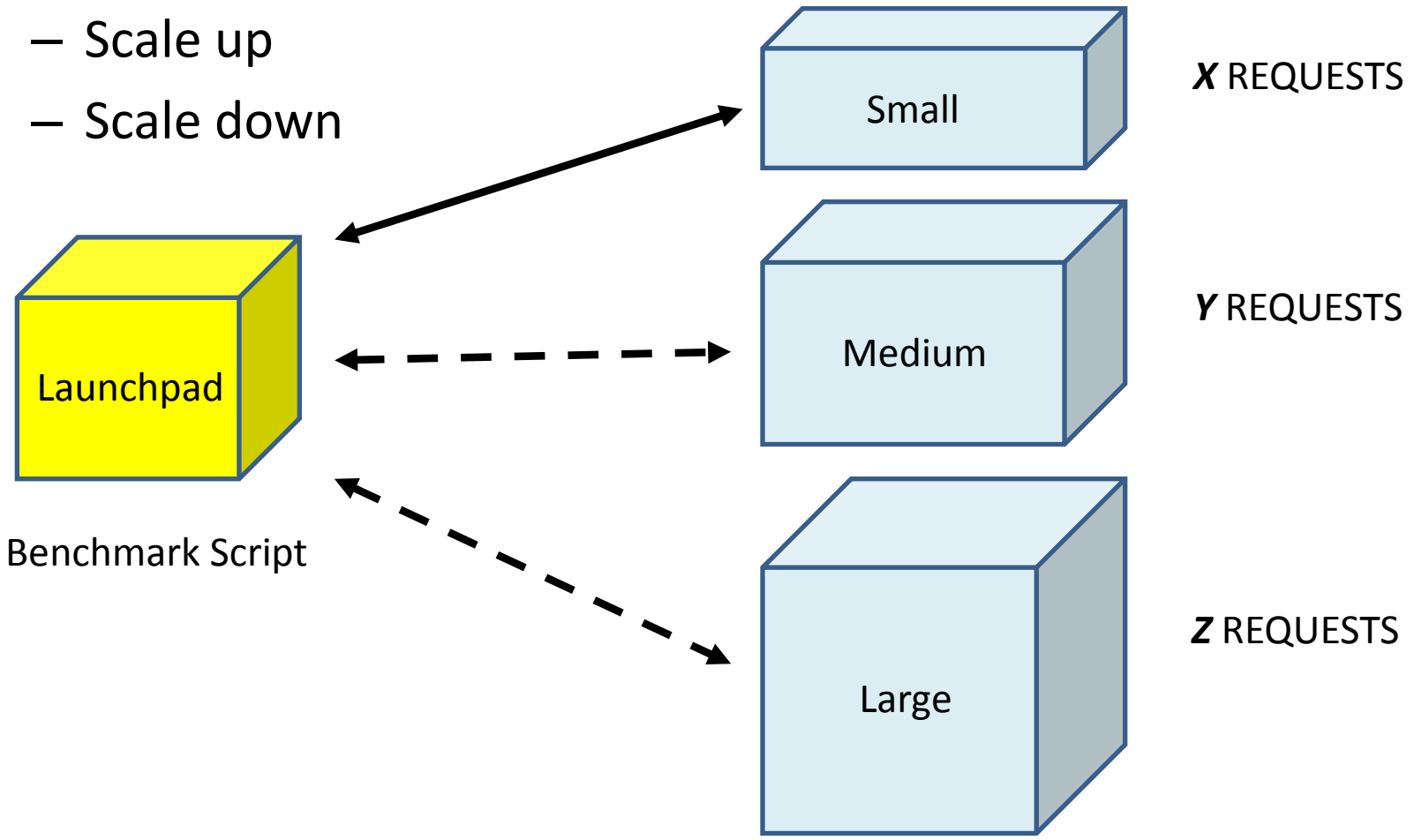
- UNIT 3: Virtualizing Resources for the Cloud
  - Module 6: Introduction and Motivation
  - Module 7: Virtualization
  - **Module 8: Resource Virtualization - CPU**
  - **Module 9: Resource Virtualization - Memory**
  - Module 10: Resource Virtualization – I/O
  - Module 11: Case Study
  - Quiz 3: Virtualizing Resources for the Cloud

This week-Project 2

- Introduction and APIs
  - Single Instance Benchmarks
- Updated: Elastic Load Balancing
  - Elastic Load Balancer
  - Static Load Benchmarking
- Auto Scaling on Amazon
  - Auto Scaling

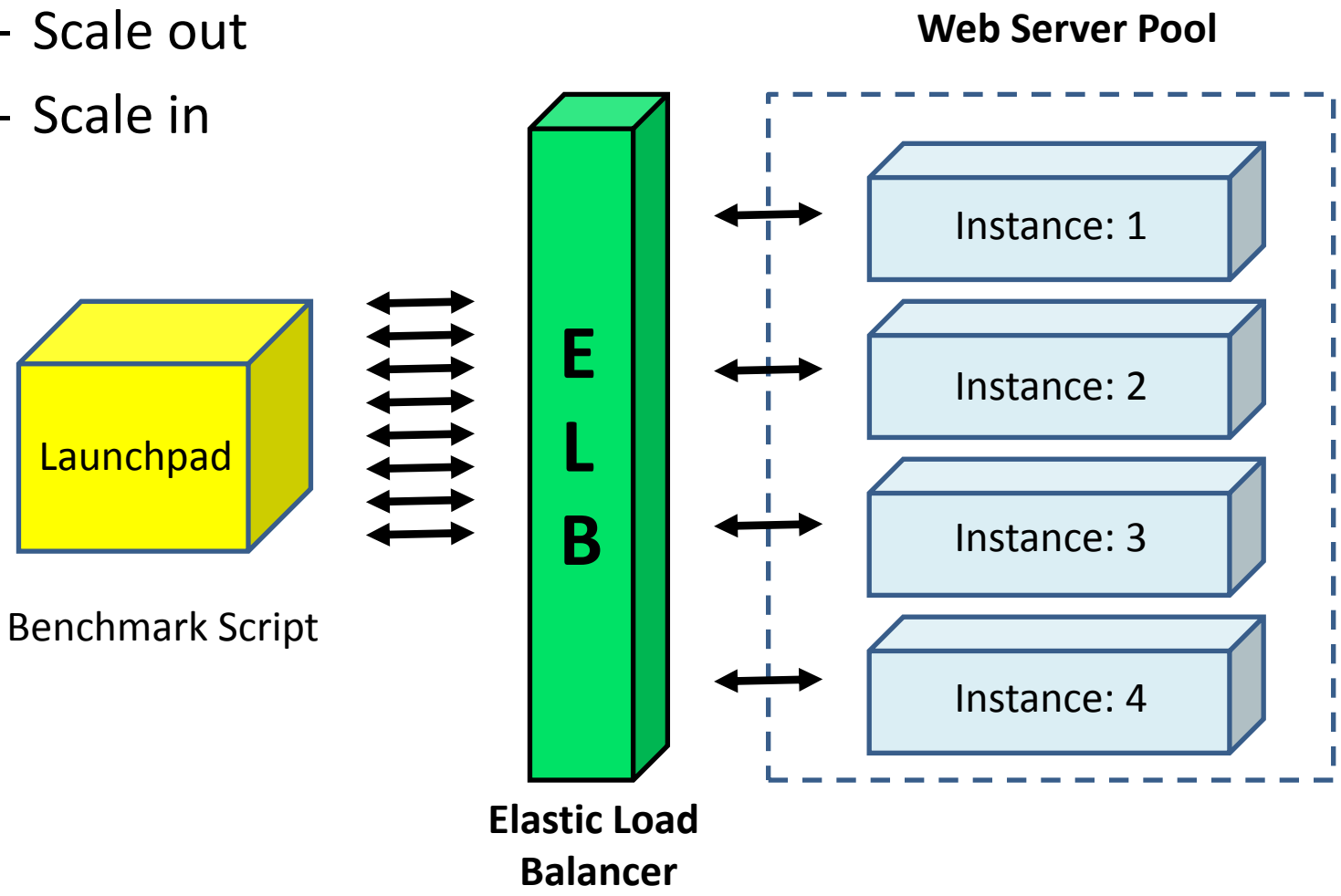
Elasticity

- Vertical Scaling
  - Scale up
  - Scale down



Elasticity

- Horizontal Scaling
  - Scale out
  - Scale in



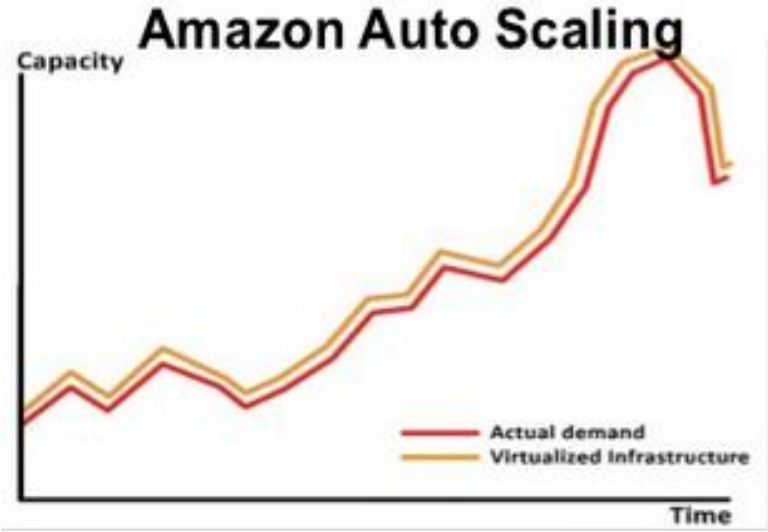
Why Auto Scaling

- Different network traffic throughout the year
  - There is a burst in holiday season
  - If performance suffers, you are losing customers
  - Should vary the system size for different seasons

Why Auto Scaling?

- Traditional Scaling:
  - Manually control the size
  - Under utilization of resources
  - Lose customers

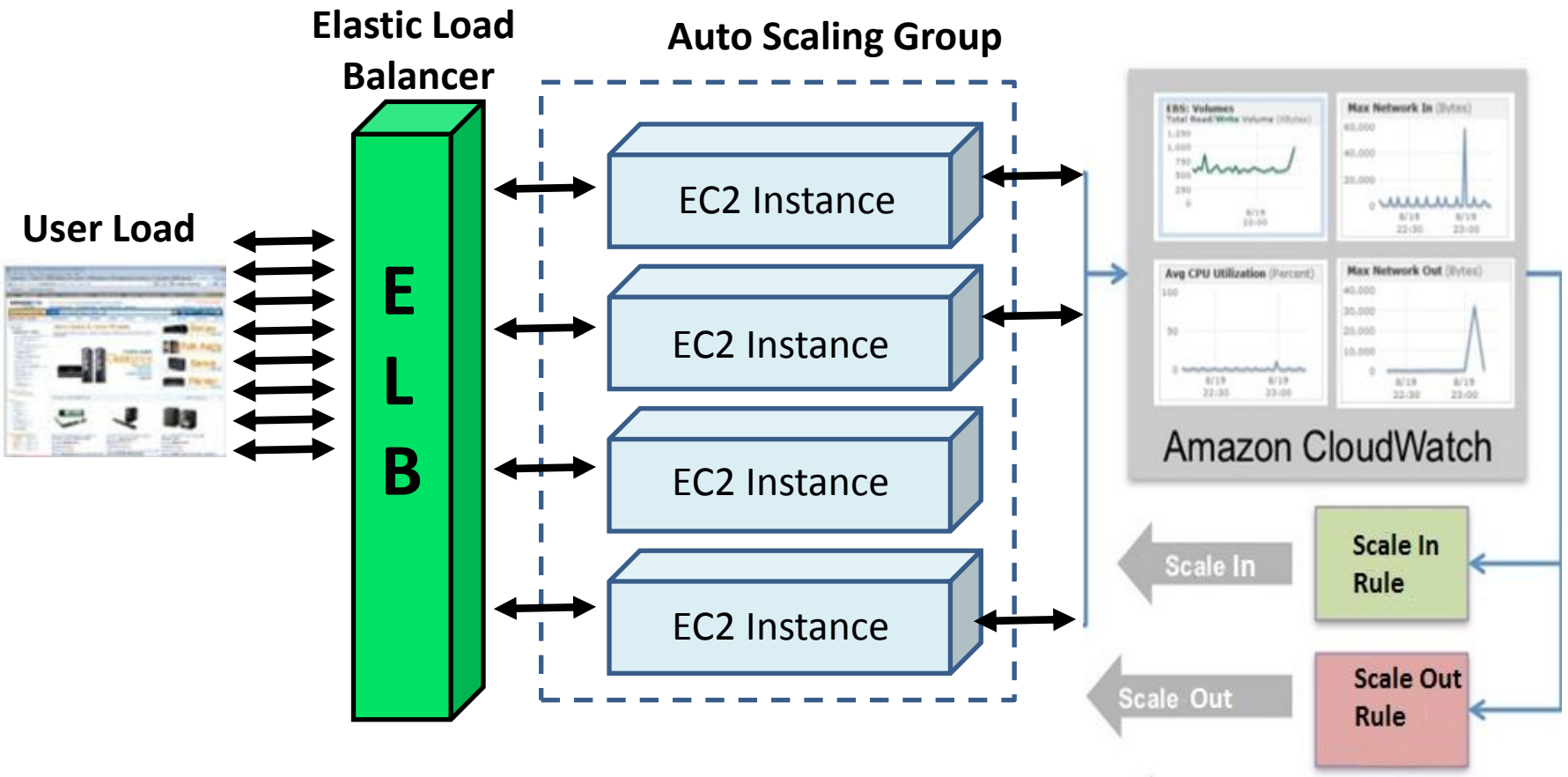
- Auto Scaling:
  - Adjust the size automatically based on demand
  - Flexible capacities and scaling sizes
  - Save cost



Picture From Amazon

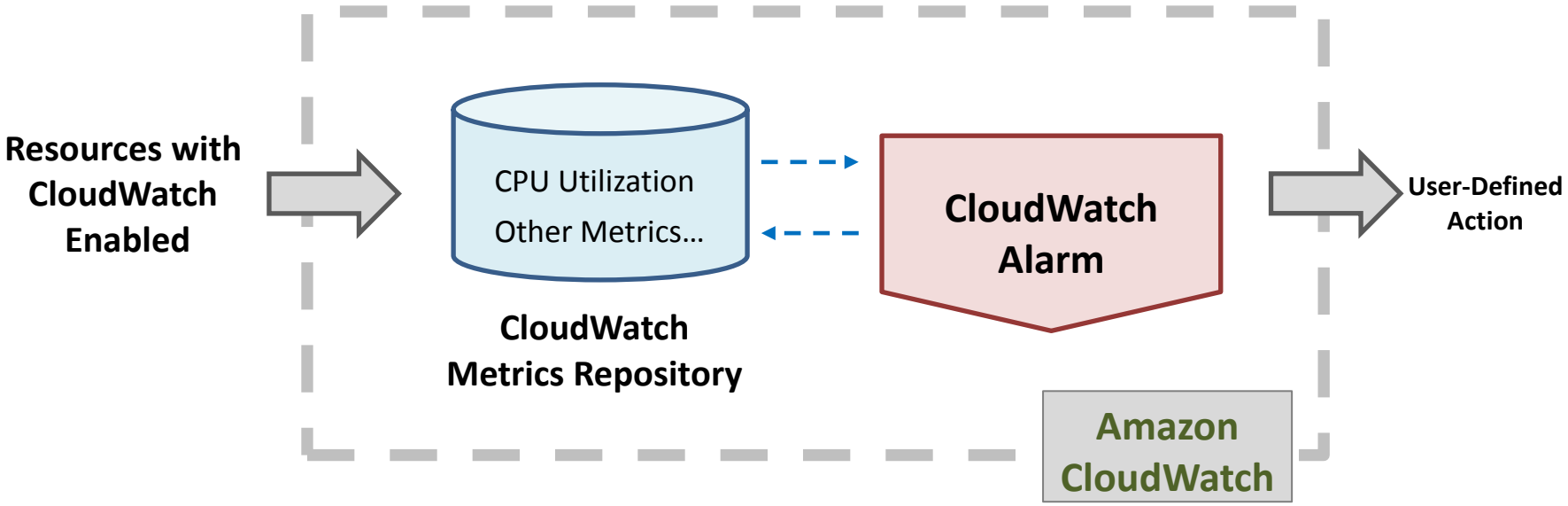
Amazon Auto Scaling Group

- Scale Amazon EC2 capacity automatically according to conditions you define



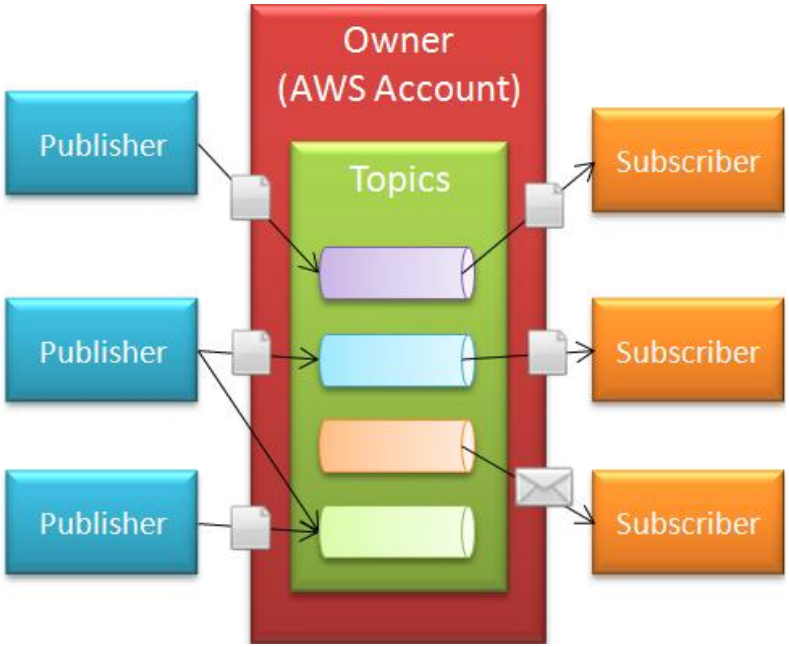
Amazon's CloudWatch Alarm

- Monitor CloudWatch metrics for some specified alarm conditions
- Take automated action when the condition is met



Amazon's SNS

- Simple Notification Service
- Fast and flexible messaging service
- Publishers push when certain events happen
- Messages belong to topics
- Subscriber will **instantly** receive messages from the topic they subscribed to when they are published

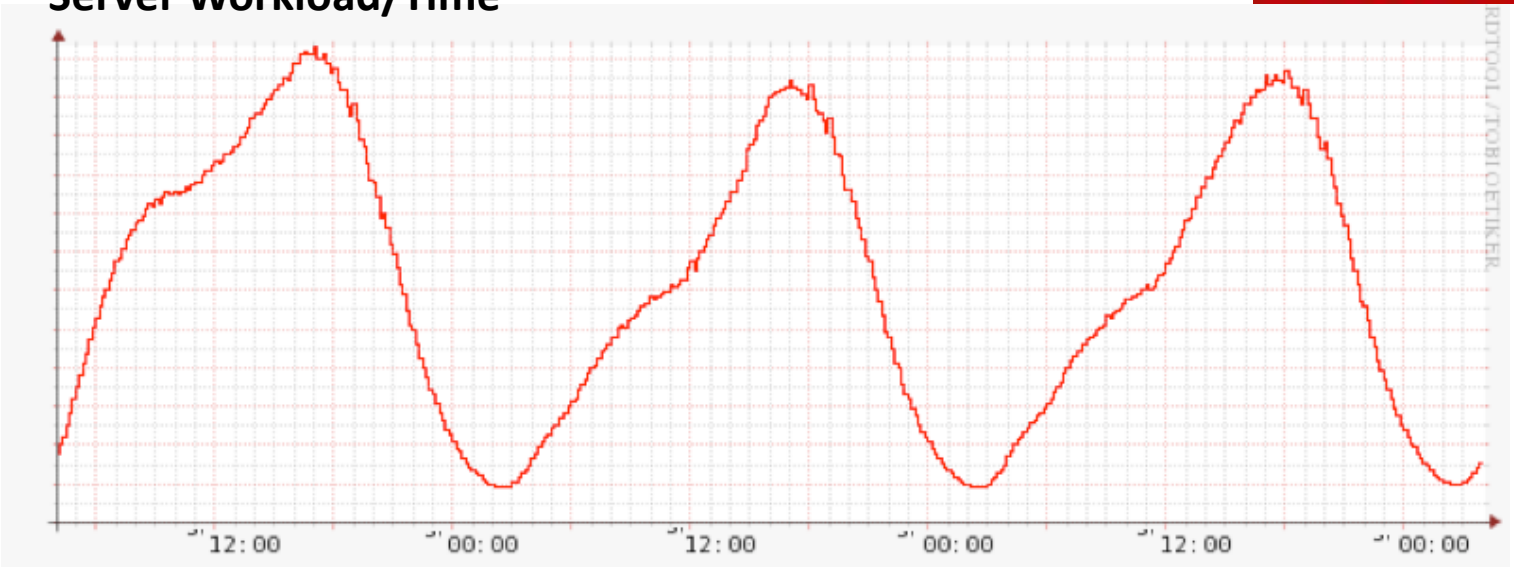


Case Study

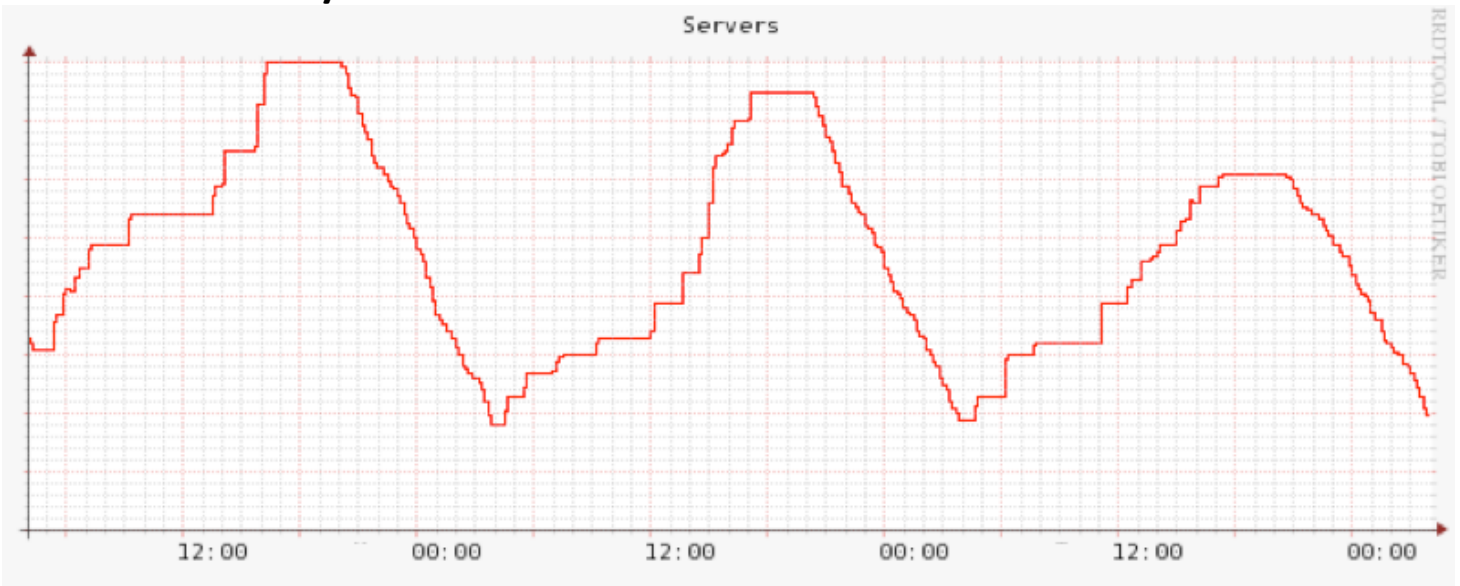
- Netflix is one of the most popular provider of on-demand Internet streaming media
  - FYI - House of Cards Season II is coming!
- Netflix has been using Amazon Auto Scaling Group for about 3 years.
- Netflix takes advantage of ASG features to manage running a pool of servers, including the capability to replace failed instances and automatically grow and shrink the size of the pool.
- Data shows that use of ASG greatly improves the availability of Netflix services and provides an excellent means of optimizing cloud costs. http://techblog.netflix.com/2012/01/auto-scaling-in-amazon-cloud.html

Case Study

Server Workload/Time



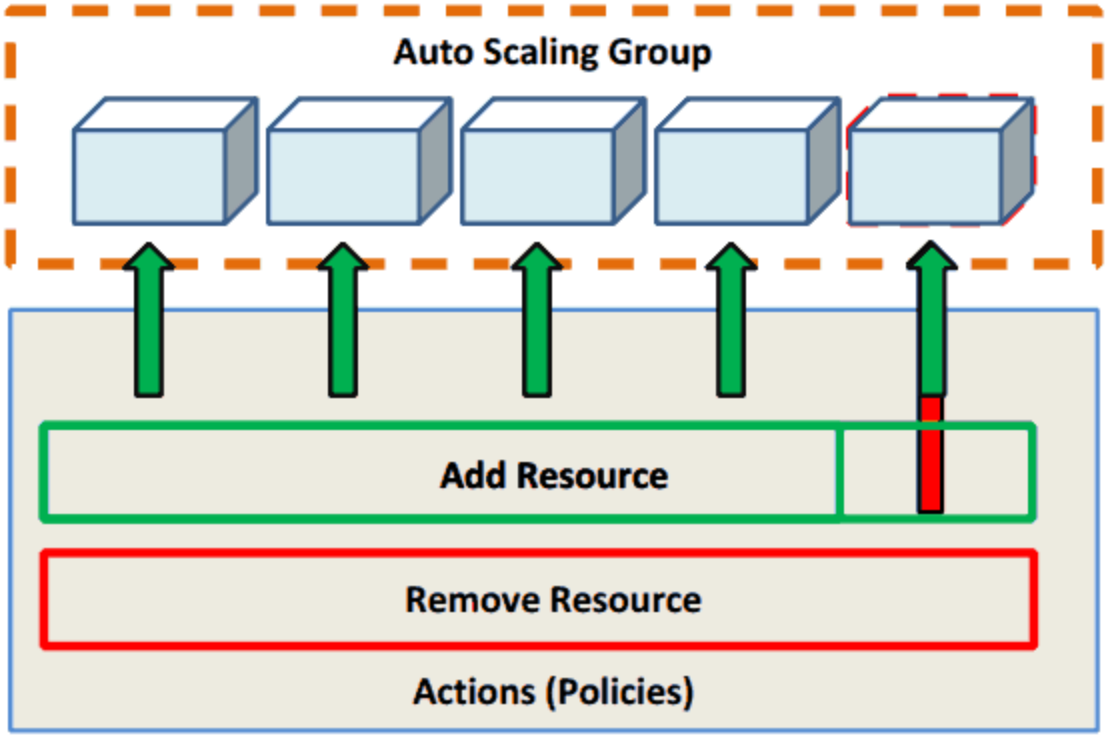
Server Number/Time



http://techblog.netflix.com/2012/01/auto-scaling-in-amazon-cloud.html

Your Task

- Use AWS Console to create an Auto-Scaling Group (ASG)
- Write a Program to build an ASG and publish its messages to SNS
- Run the benchmark program and observe changes



Resources

- Amazon's Auto Scaling Service
  - http://aws.amazon.com/autoscaling/
- Amazon's CloudWatch Alarm
  - http://aws.amazon.com/cloudwatch/
- Amazon's SNS (Simple Notification Service)
  - http://aws.amazon.com/sns/
- Amazon's Scaling Developer
  - http://aws.amazon.com/autoscaling/developer-resources/

Upcoming Deadlines

- Project 2.4 (Due Feb 23 11:59PM)

AutoScaling on Amazon (Gradebook) (Learning Dashboard)			
	Auto Scaling	Checkpoint	Not yet assigned Due date TBD by instructor

Demo Outline

- Create Auto Scaling Group using AWS console
  - Create Auto Scaling Launch Configuration
  - Create Auto Scaling Group
  - Create Auto Scaling Policy
  - Configure SNS
  - Delete Auto Scaling Group

Create Auto Scaling Launch Configuration

Create Auto Scaling Group

Create Auto Scaling Policy

Configure SNS

- Create a SNS topic
- Subscribe to topic with email

Delete Auto Scaling Group

- Terminate all instances
- Delete Auto Scaling Group
- Delete Auto Scaling Launch Configuration

Create Auto Scaling Group using Java API

- Useful packages
  - com.amazonaws.services.autoscaling
  - com.amazonaws.services.autoscaling.model
- http://aws.amazon.com/articles/3586?_encoding=UTF8&jiveRedirect=1

Create Auto Scaling Group using Python API

- Useful packages
  - http://boto.readthedocs.org/en/latest/autoscale_tut.html