

15-319 / 15-619

Cloud Computing

Recitation 5

February 10th & 12th, 2015

Overview

- **Administrative issues**
 - Piazza guidelines, 15619 Project teams, **credentials**, Quiz 2
- **Last week's reflection**
 - Project 2.1, OLI unit 2 module 5, Quiz
- **This week's schedule**
 - Project 2.2 - Feb 15th
 - Unit 3 - Module 6 and 7
- **Demo**

Announcements

- Monitor AWS expenses regularly
 - Check your bill frequently
- Protect your credentials
 - A Student left credentials in a public place and got hacked. \$5k spent in 20 hours.
 - Crawlers are looking for AWS credentials on public repos!

Announcements

- Manual grading of Question 26 on Quiz 2
 - Question on PUE was manually graded to take into account different interpretations of graph
- 15-619 Students:
 - Form your 15619 Project teams
 - Up to 3 students per team
 - Frontend, databases, scripting, ...
 - Piazza post [@5](#)

Code Submission Announcements

In your submissions:

Do include:

- readme.txt
- References in a file called 'references'

Do **not include:**

- Packages (jar files)
- Folder structures
- Input or output data
- **Pem files**

Last Week's Reflection

- Content
 - Unit 2: Data Centers, Quiz 2 completed
- EC2 APIs
 - Amazon CLI, Java, Python
- Vertical and Horizontal Scaling
 - Instance Capacity
- First experience with Elastic Load Balancing

Project 2.1 Questions

- Cannot connect to instance
 - Check your SecurityGroup
- How to make sure the instance is running?
 - DescribeInstanceRequest $\Rightarrow$ correct
 - instance.getState() $\Rightarrow$ wrong
- And...you still need time to initialize?
 - Call /track/person to ensure DC server is up
- ELB will be warmed up after the first run, and will have better performance than before
- Getting aggregate RPS for the last minute
 - Parse the logs and sum the RPS up

This Week: Content

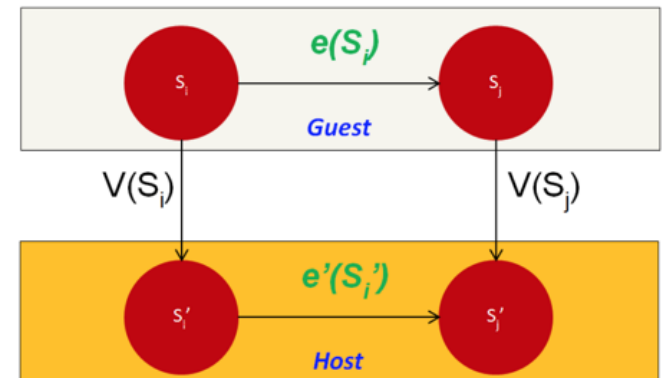
- UNIT 3: Virtualizing Resources for the Cloud
 - **Module 6: Introduction and Motivation**
 - **Module 7: Virtualization**
 - Module 8: Resource Virtualization - CPU
 - Module 9: Resource Virtualization - Memory
 - Module 10: Resource Virtualization – I/O
 - Module 11: Case Study

Module 6: Introduction and Motivation

- Why Virtualization
 - Enabling the cloud computing system model
 - Elasticity
 - Resource sandboxing
 - Improved system utilization and reduce costs

Module 7: Virtualization

- What is Virtualization
 - Involves the construction of an isomorphism that maps a virtual guest system to a real (or physical) host system
- Virtual Machine Types
 - Process Virtual Machines
 - System Virtual Machines



Virtualization

This Week: Project

- P2.1: Introduction to APIs
 - MSB Recruitment Knowledge Test
- P2.2: Autoscaling and Elastic Load Balancing
 - Junior System Architect at the MSB

Project 2.1 revisited

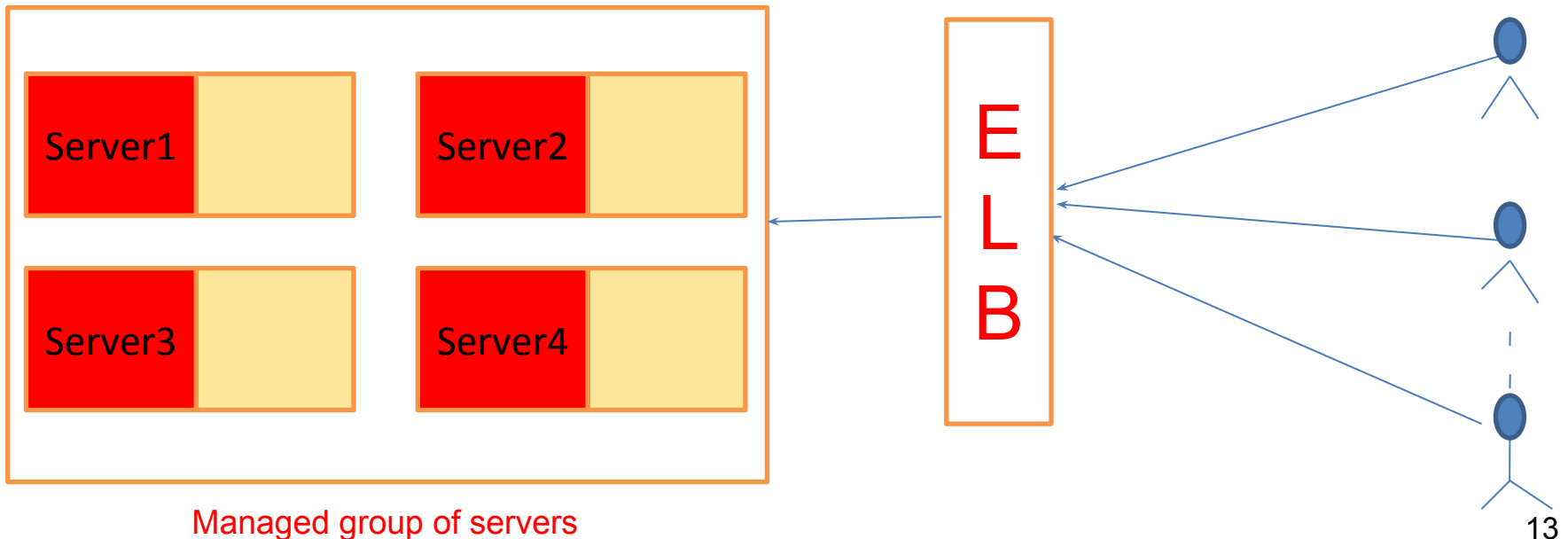
- Vertical Scaling Limitations
 - Can only increase the capacity to a limit
 - When scaling, need to transfer data, have to reboot
- Solution: Horizontal Scaling (add more resources)
 - Load balancer to distribute the load



Elastic Load Balancer

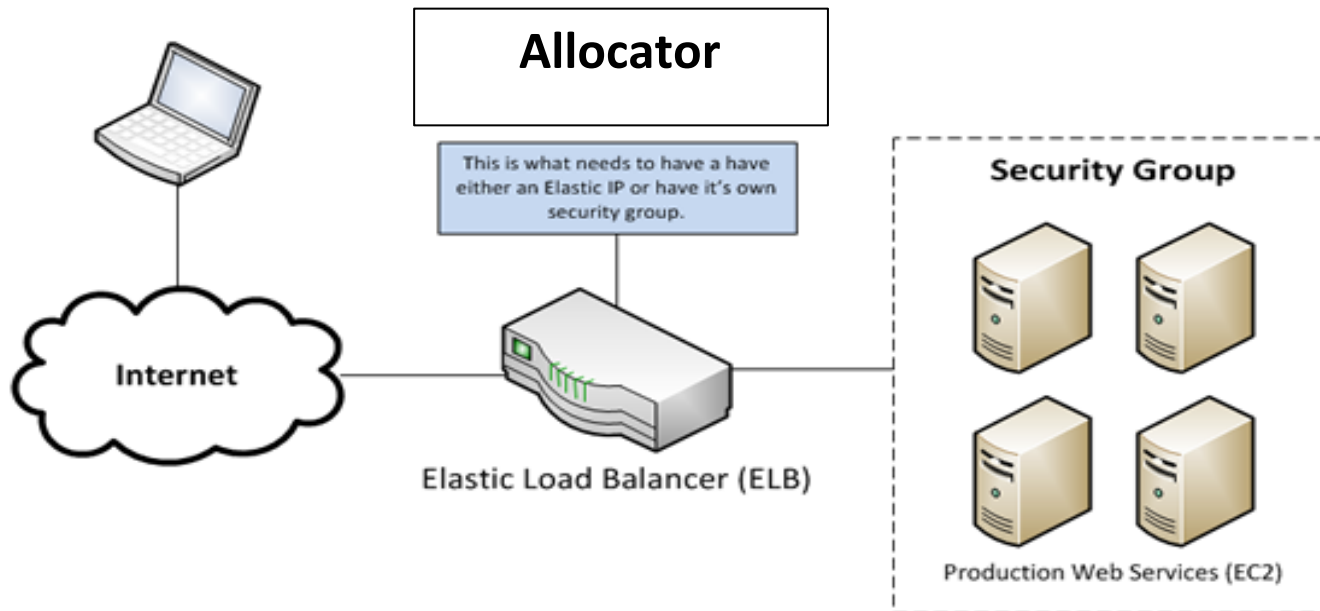
- Make sure that workload is even on each server
- Do not assign load to servers that are down

How does AWS help solve these problems?



Elastic Load Balancer

- ELB is a gateway that acts as a router interface and sends incoming requests to multiple EC2 Instances sitting behind it
- Distribute requests from clients to all servers equally



ELB Features

- Using ELB, you can distribute incoming traffic across your Amazon EC2 instances in multiple Availability Zones (redundancy within the same region)
- ELB can detect the health of Amazon EC2 instances. When it detects unhealthy instances, it spreads the load to other healthy instances
- ELB can offer integration with Auto Scaling to ensure that you can meet varying levels of traffic without requiring manual intervention
- Can (theoretically) handle any amount of traffic!

ELB Warmup

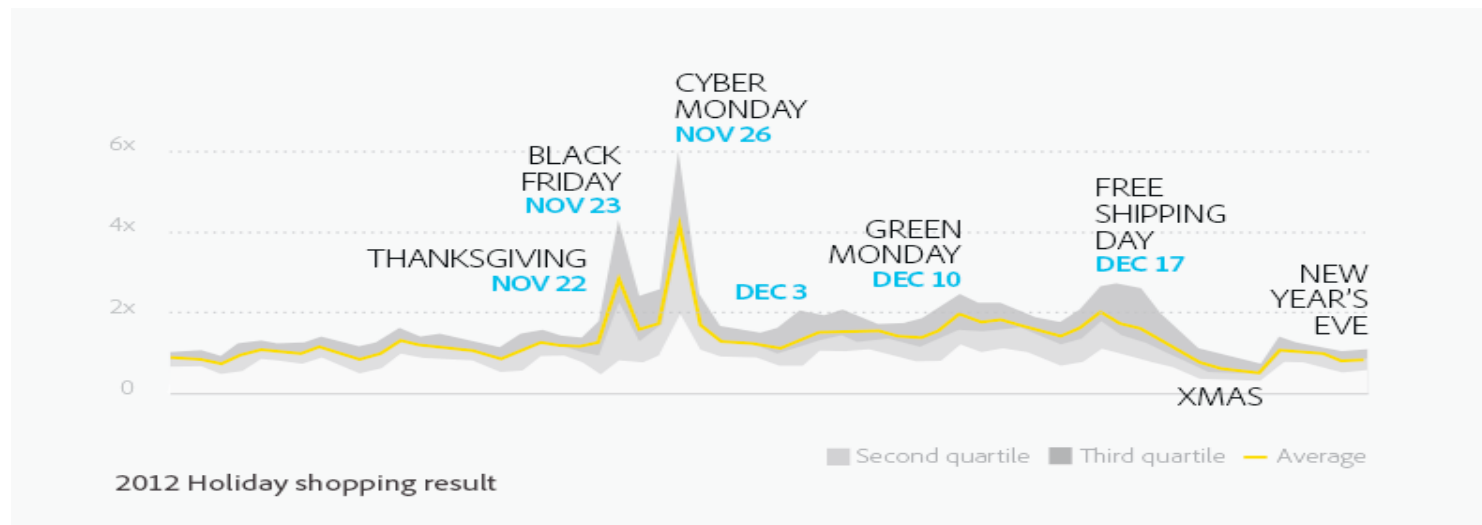
- ELB has a starting point for its initial capacity, and it will scale up or down based on traffic
- It struggles with high traffic spikes in shorter periods
- It is recommended that the load is increased at a rate of no more than 50 percent every five minutes

ELB Best Practices

<https://aws.amazon.com/articles/1636185810492479>

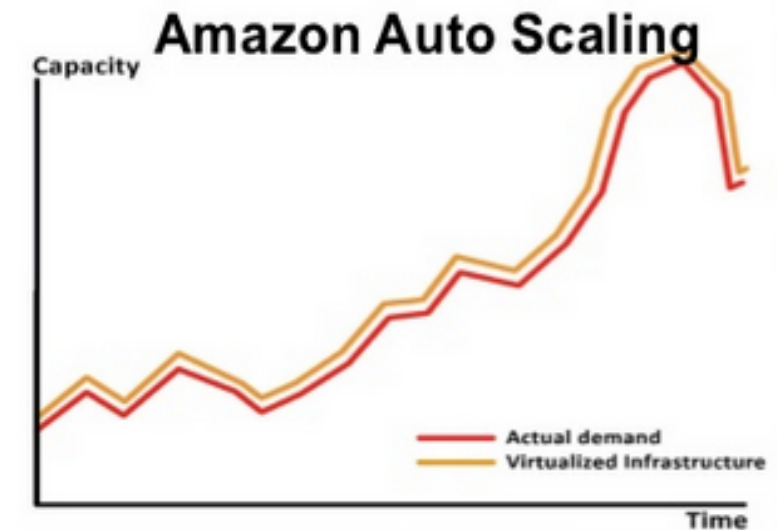
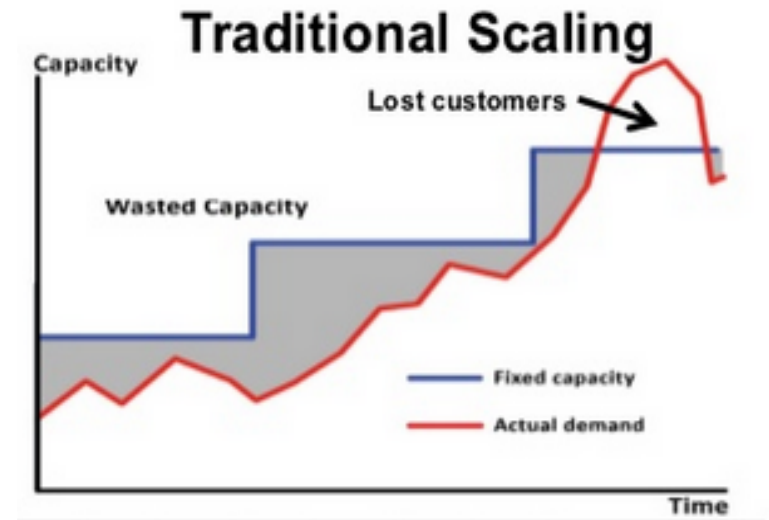
Why Auto Scaling?

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



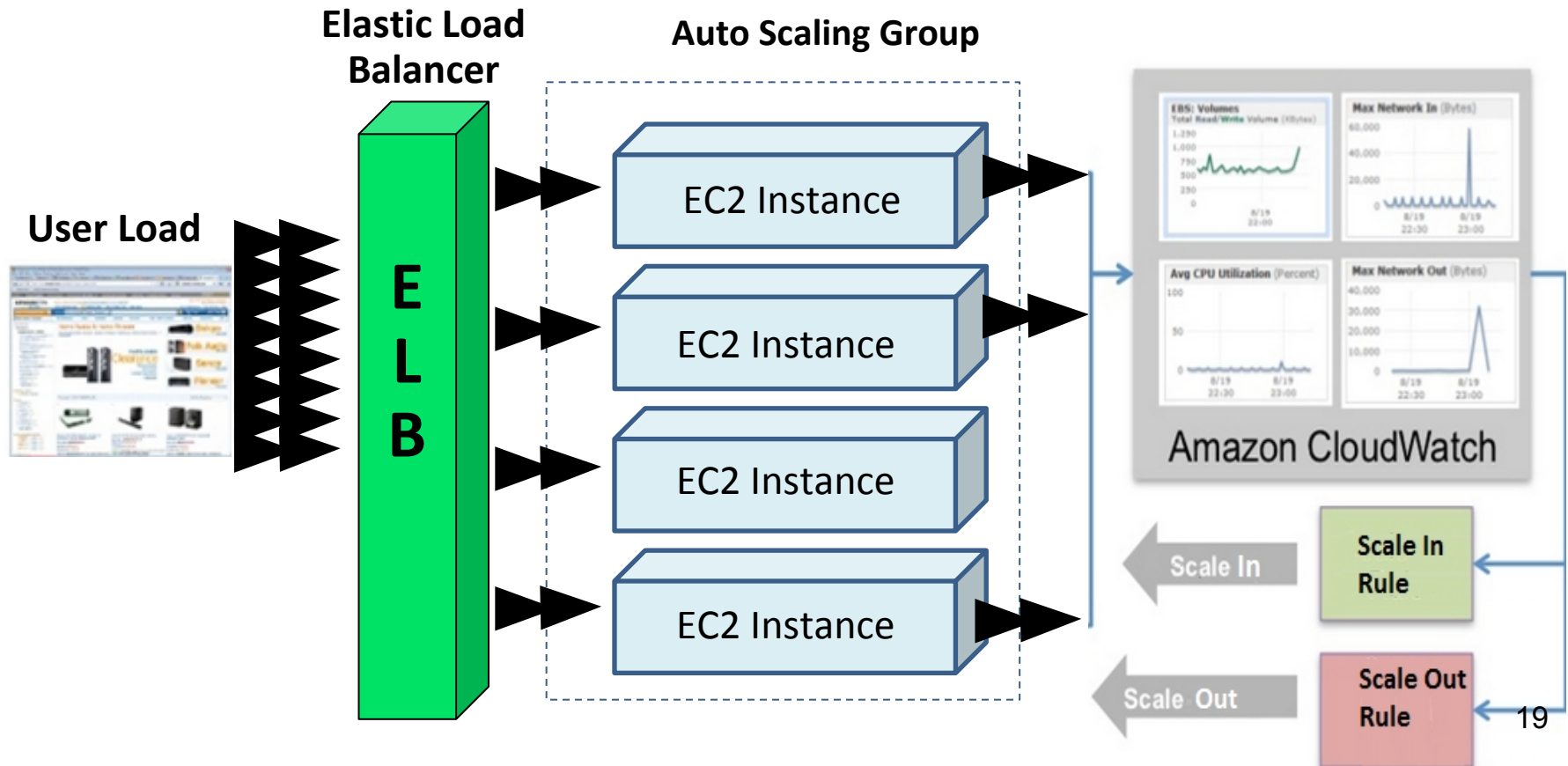
Why Auto Scaling?

- Traditional Scaling:
 - Manually control the size
 - Under utilization of resources
 - Lose customers
- Auto Scaling:
 - Automatically adjust the size based on demand
 - Flexible capacities and scaling sizes
 - Save cost



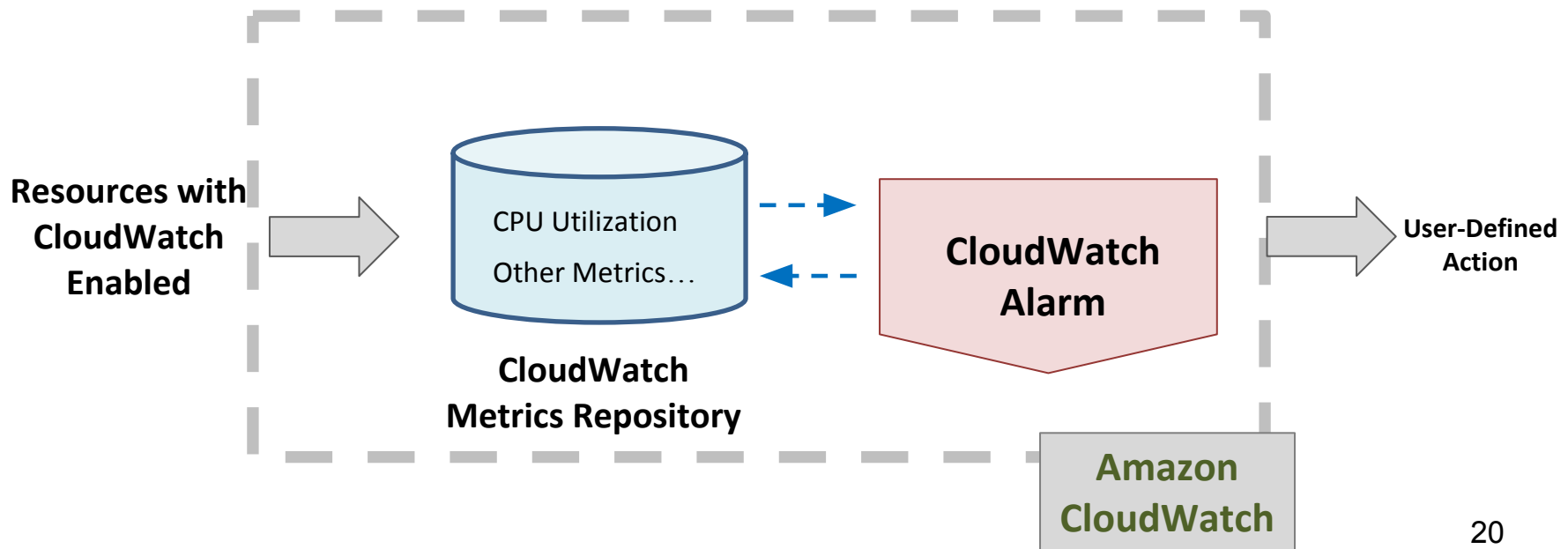
Amazon Auto Scaling Group

- Scale Amazon EC2 capacity automatically according to policies you define



Amazon's CloudWatch Alarm

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



Case Study

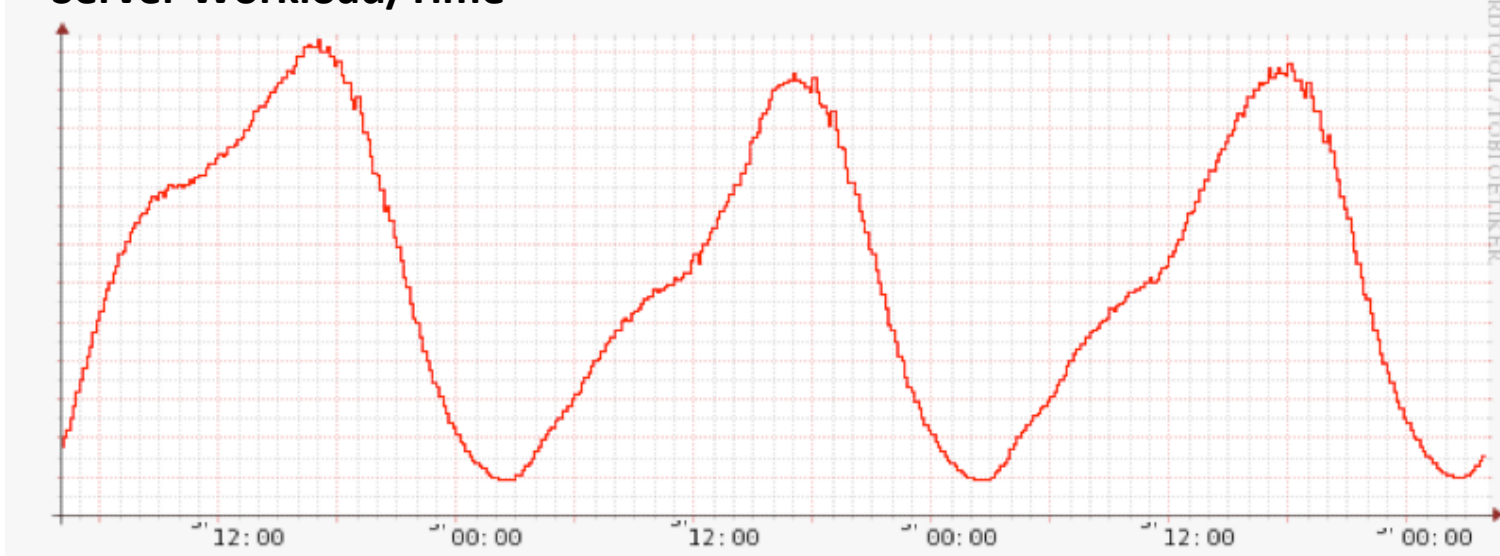


- Netflix is one of the most popular provider of on-demand Internet streaming media
- Netflix has been using Amazon Auto Scaling Group for about 5 years.
- Data shows that use of ASG greatly improves the availability of Netflix services and provides an excellent means of optimizing cloud costs. <http://techblog.netflix.com/2012/01/auto-scaling-in-amazon-cloud.html>

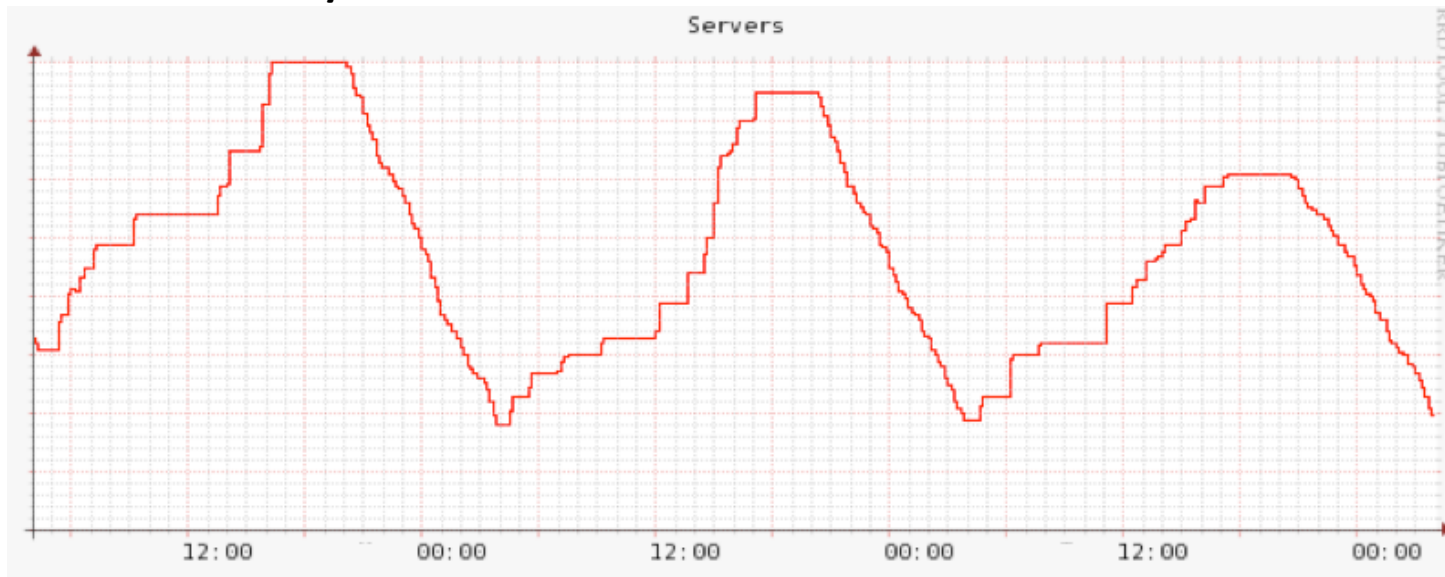
Case Study

NETFLIX

Server Workload/Time



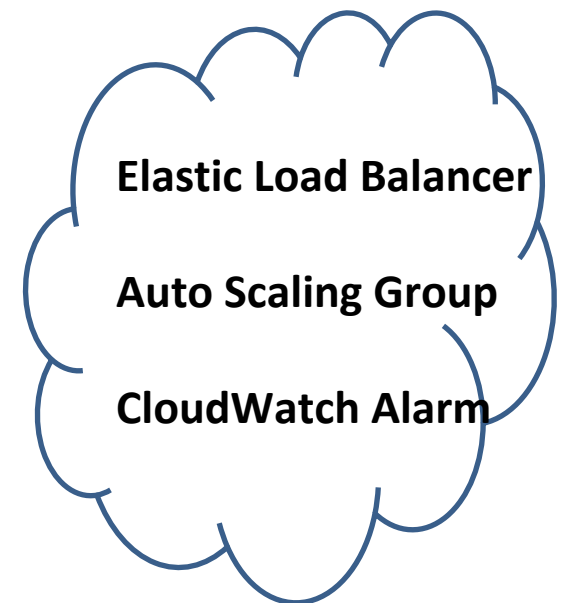
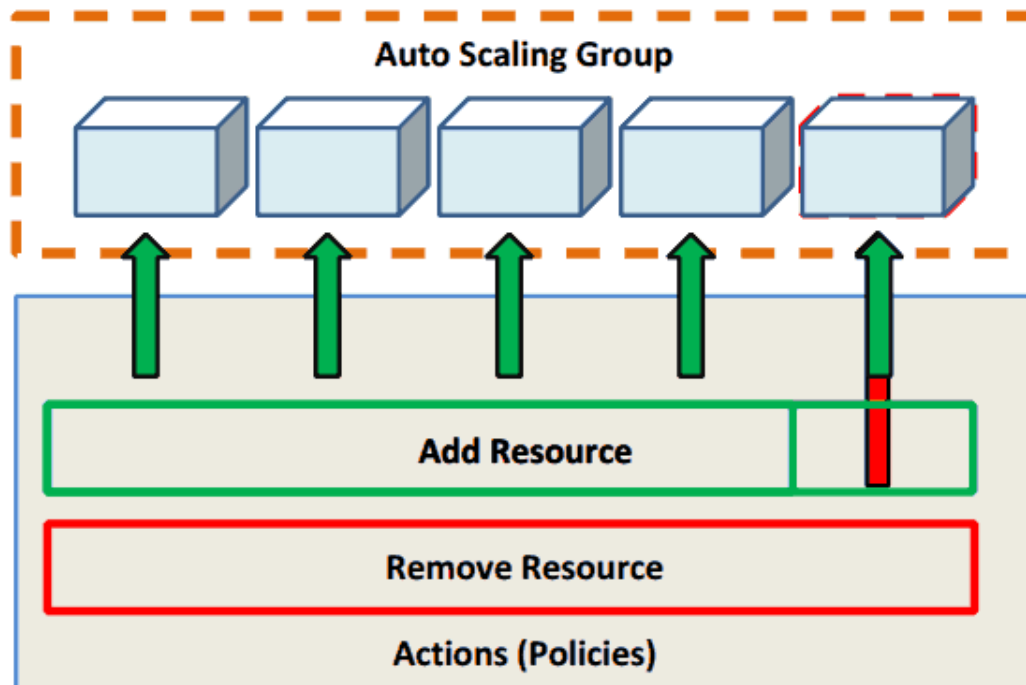
Server Number/Time



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

P2.2 - Your Task

- Programmatically create an Elastic Load Balancer (ELB) and an Auto-Scaling Group (ASG) linked to ELB
- Test by submitting a URL request and observe changes
- Decide on Scale-Out and Scale-In policy



Project 2.2 Suggestions

- Read project description more than once
- Think about workflow before starting
- Look up API references
 - Read overview first, then details
 - Use samples to go over simple APIs
 - Search the Internet to debug
- Check every step carefully, Debug with AWS console
- **Don't forget to terminate your Auto Scaling Group and your Elastic Load Balancer after everything is done.**

Resources

- Amazon's Auto Scaling Service
 - <http://aws.amazon.com/autoscaling/>
- Amazon's CloudWatch Alarm
 - <http://aws.amazon.com/cloudwatch/>
- Amazon's Scaling Developer
 - <http://aws.amazon.com/autoscaling/developer-resources/>

Upcoming Deadlines

- Project 2.2: Load Balancing and AutoScaling
 - Due: 02/15/15 11:59PM Pittsburgh
- Unit 3: Virtualization Resource for the Cloud
 - Module 6: Introduction and Motivation
 - Module 7: Virtualization



Project 2.2 Demo

Project 2.2 Demo

- Create Auto Scaling Group Manually
 - [AWS Console](#)
- Load Generator
 - Start ELB Warmup
 - Start Junior System Architect Test
- Issues
 - Scaling History
 - Low RPS
 - No Data Points
 - Traffic Plateau

Scaling History

The screenshot displays the AWS Management Console interface for an Auto Scaling Group. On the left, a navigation pane lists various AWS services, with 'Auto Scaling Groups' selected. The main content area shows the 'Create Auto Scaling group' button and an 'Actions' dropdown. Below this, a table lists the Auto Scaling groups. The 'P22AutoScalin...' group is selected, and its details are shown. The 'Scaling History' tab is highlighted with a red box. The table below shows the scaling history for this group.

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Cooldown	Health Check Grace
P22AutoScalin...	P22LaunchConfiguratio...	1	1	1	1	us-east-1b	300	300

Auto Scaling Group

Details Scaling History Scaling Policies Instances Notifications Tags

Filter: Any Status Filter scaling history...

Status	Description	Start Time	End Time
Successful	Launching a new EC2 instance: i-1f3be3f0	2015 February 9 20:58:01 UTC-5	2015 February 9 20:58:34 UTC-5

[Minute 33]
rps=851.88

[Minute 34]
rps=0

[Minute 35]
rps=0

[Minute 36]
rps=0

[Minute 37]
rps=0

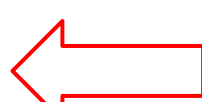
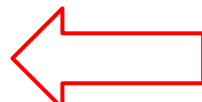
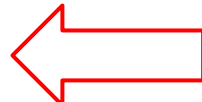
[Minute 38]
rps=0

[Minute 39]
rps=0

[Minute 40]
rps=-.01

[Minute 41]
rps=0

[Minute 42]
rps=66.66



RPS = 0, why?

CloudWatch Monitoring Details



Sum HTTP 2XXs (Count)

Statistic: Sum

Time Range: Last 3 Days

Period: 1 Minute



Close

CloudWatch Monitoring Details

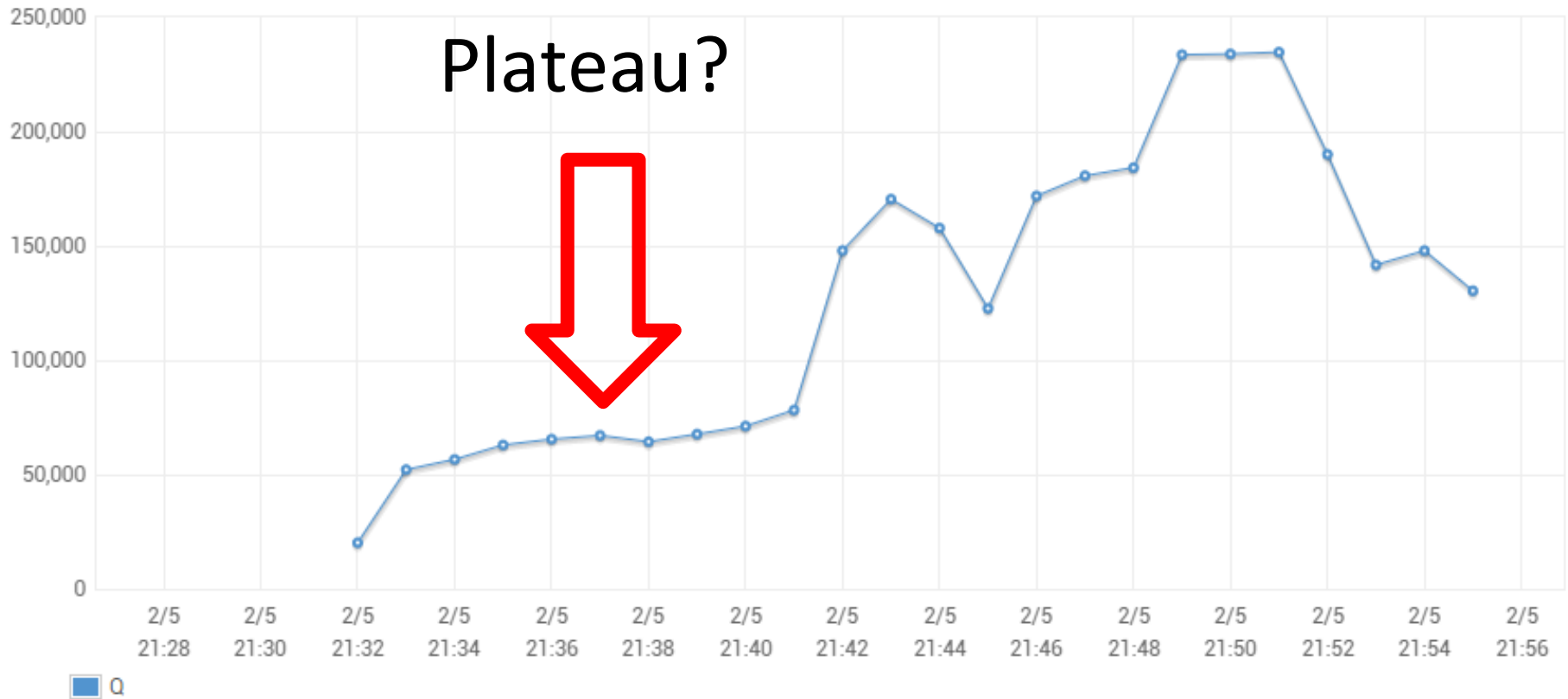
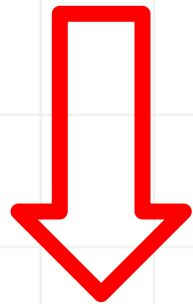


Sum HTTP 2XXs (Count)

Statistic: Sum ▾

Time Range: Last 12 Hours ▾ Period: 1 Minute ▾ 

Plateau?



Close